REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE YAHIA FARES DE MEDEA

Faculté de **T**echnologie

Département du **G**énie **E**lectrique

Mémoire de fin d'études de Master

Filière : Electronique

Spécialité : Instrumentation

Conception et réalisation d'un compteur d'énergie Intelligent à base d'IoT

Présenté par :

BOUDRA Abdelouahab

BEZAZ Miloud Zin El Abidine

Proposé et Dirigé par :

Encadrant : Dr. BENYEZZA Hamza

Co-encadrant : Pr. BOUHEDDA Mounir

Promotion 2023

Travail préparé au sein du Laboratoire des Systèmes Électroniques Avancés (LSEA)

de l'Université de Médéa, ALGÉRIE

Dédicace

Je souhaite dédier ce mémoire à plusieurs personnes qui ont joué un rôle significatif dans mon parcours académique et personnel.

Tout d'abord, j'aimerais exprimer ma gratitude envers mes parents, qui m'ont soutenu de manière inconditionnelle tout au long de mes études. Leur amour, leur encouragement et leur confiance en moi ont été des sources constantes de motivation. Ce mémoire est le fruit de leur dévouement et de leur soutien indéfectible.

J'adresse également mes remerciements à mes professeurs et encadrants académiques. Leurs connaissances, leurs conseils éclairés et leur patience ont été essentiels pour mon développement intellectuel. Leur engagement envers l'éducation m'a inspiré à poursuivre mes aspirations académiques.

Je tiens à exprimer ma reconnaissance envers mes amis qui ont partagé cette aventure avec moi. Leurs encouragements, leur soutien mutuel et leurs discussions stimulantes ont rendu cette expérience encore plus enrichissante.

Enfin, je voudrais exprimer ma reconnaissance à tous ceux qui ont contribué indirectement à la réalisation de ce mémoire. Leurs précieuses ressources, leur aide technique et leurs précieux commentaires ont été d'une grande valeur pour moi.

Merci du fond du cœur à tous ceux qui ont fait partie de ce voyage. Votre soutien a été inestimable et je vous en suis éternellement reconnaissant.

[BEZAZ Miloud Zin El Abidine]

Dédicace :

C'est avec profond et sincères mots

Que je dédier ce modeste travail de la fin d'étude a

Mes chers parent ; qui ont sacrifice leur vie pour mon réussite

Et m'on éclaire le chemin par

Leur conseil judicieux.

J'espère qu'un jour,

Je pourrai leurs rendre un peu de ce qu'ils ont fait

Pour moi, que dieu leur prête bonheur et long vie.

Je dédier aussi ce travail à nos amis qui ont soutenu en cours de tout ce parcours et spécialement a monsieur TAKDJOUT AMINE qui ma donner un grand pousse pour finir ce travail

A mes camarades d'université

A tous nos professeurs qui nous enseigner

Et à tous ceux qui nous sont chers.

Boudra Abdelouahab

Remerciements

Nous souhaitons exprimer notre profonde gratitude envers toutes les personnes qui ont contribué de près ou de loin à la réalisation de ce mémoire.

Tout d'abord, nous sommes reconnaissants envers nos parents pour leur soutien inconditionnel, leur amour et leur encouragement tout au long de nos études. Leurs sacrifices et leur confiance en nous ont été des sources d'inspiration et de motivation.

Nous tenons également à exprimer notre reconnaissance à nos professeurs et encadrants académiques. Leurs connaissances, leur expertise et leur dévouement ont grandement enrichi notre expérience d'apprentissage. Leurs conseils éclairés et leurs commentaires constructifs ont joué un rôle essentiel dans l'amélioration de ce travail.

Nous sommes également reconnaissants envers nos amis et camarades de classe qui ont partagé ce voyage avec nous. Leurs encouragements, leur soutien mutuel et leurs discussions animées ont créé un environnement propice à l'apprentissage et à l'épanouissement.

Nous tenons à remercier toutes les personnes qui ont contribué indirectement à ce mémoire, que ce soit en fournissant des ressources, des données ou des conseils précieux. Votre contribution a été inestimable et a grandement enrichi ce travail.

Enfin, nous voudrions adresser nos remerciements à nos familles et à nos proches pour leur soutien constant, leurs encouragements et leur compréhension pendant cette période intense d'études.

Nos sincères remerciements à tous ceux qui ont joué un rôle dans la réalisation de ce mémoire. Votre contribution a été inestimable et nous apprécions sincèrement tous les efforts déployés.

الملخص

يهدف المشروع إلى تحسين عدادات الطاقة الحالية باستخدام الإنترنت للتحكم في استهلاك العملاء للطاقة وتسهيل الفوترة. يتم تثبيت بطاقة إلكترونية مدمجة بجوار عداد الطاقة لإرسال معلومات الاستهلاك إلى خادم حيث يمكن تخزينها في قاعدة بيانات للحفاظ على سجل. تتصل تطبيق ويب بمزود الطاقة والعميل والعداد لتسهيل إصدار واستلام الفواتير ورصد الطاقة وزيادة الأمان من خلال التدخل عن بعد.

الكلمات المفتاحية : عداد طاقة ذكي - انترنت الأشياء - نظام ذكي - قاعدة بيانات

Résumé

Le projet vise à améliorer les compteurs d'énergie actuels en utilisant l'IoT pour contrôler la consommation d'énergie du client et faciliter la facturation. Une carte électronique embarquée est installée à côté du compteur d'énergie pour envoyer les informations de consommation vers un serveur où elles peuvent être stockées dans une base de données pour garder un historique. Une application web connecte le fournisseur d'énergie, le client et le compteur pour faciliter l'émission et la réception des factures, le monitoring énergétique et l'augmentation de la sécurité grâce à une intervention à distance.

Mots clés : Compteur d'énergie intelligent – IdO – Système intelligent – Base de données

Abstract

The project aims to improve current energy meters using IoT to control customer energy consumption and facilitate billing. An embedded electronic card is installed next to the energy meter to send consumption information to a server where it can be stored in a database to keep a record. A web application connects the energy supplier, the customer, and the meter to facilitate the issuance and receipt of bills, energy monitoring, and increased security through remote intervention.

Keywords : Smart energy meter – IoT – Smart system - Database

Sommaire

Liste des abréviations

Liste des figures

Liste des tableaux

Chapitre 01 : Généralités sur les compteurs d'énergie

I.1 Introduction 5

I.2 Historique des compteurs d'énergie électrique 5

I.3 Compteur électromécanique 6

I.3.1 Définition 6

I.3.2 Principe de fonctionnement 7

I.3.3 Avantages et Inconvénients d'un compteur électromécanique 8

I.4 Compteur électronique 9

I.4.1 Définition 9

I.4.2 Principe de fonctionnement d'un compteur électronique 10

I.4.3 Avantages et inconvénients d'un compteur électronique 10

I.5 Compteur intelligent 11

I.5.1 Définition 11

I.5.2 Histoires des compteurs intelligents 12

I.5.3 Principe du fonctionnement des compteurs intelligents 13

I.5.4 Compteur intelligent à base d'IoT 13

I.5.5 Avantages et inconvénients d'un compteur intelligent 14

I.6 Conclusion 15

II. Chapitre 02 18

II.1 Introduction 18

II.2 Internet des objets (IoT) 18

II.3 Partie Hardware 22

II.3.1 Présentation du module nRF24L01 22

II.3.2 Caractéristiques techniques du module nRF24L01 22

II.3.3 Description des pins du nRF24L01 23

II.3.4 Utilisations courantes du module nRF24L01 24

II.3.5 Module Arduino nano 25

II.3.6 Communication Arduino Nano et nRF24L01 28

II.4 Partie Software 29

II.4.1 Arduino IDE 29

II.4.2 VSCode 31

II.4.3 Appwrite 34

II.5 Conclusion 35

III. Chapitre 03 38

III.1 Introduction 38

III.2 Description du système 38

III.3 Principe de fonctionnement de système proposé 39

III.4 Solutions hardware et software 42

III.4.1 Hardware 42

III.4.2 Software 42

III.5 Algorithmes 43

III.6 Implémentation du système 44

III.6.1 Captage et transfert des données 44

III.6.2 Création de la base de données 48

III.6.3 Interface Homme Machine IHM 52

III.7 Résultats des tests 63

III.8 Conclusion 69

Liste des abréviations

CAN : Convertisseur Analogique Numérique

CPL : Courants Porteurs en Ligne

CSS : Cascading Style Sheets

GPRS General Packet Radio Service

HTML : HyperText Markup Language

I2C: Inter-Integrated Circuit

IDE : Integrated Development Environment

IoT : Internet of Things

Mbps : Mégabit par seconde

SPI : Serial Peripheral Interface

TIC : Technologies de l'information et de la communication

UART : Universal Asynchronous Receiver Transmitter

VSCode : Visual Studio Code

Wi-Fi : Wireless Fidelity

Liste des figures

Figure I.1 : Compteur électromécanique 6

Figure I.2 : Architecture interne d'un compteur électromécanique 7

Figure I.3 : Compteur électromécanique 8

Figure I.4 : Compteur électronique (numérique) 9

Figure I.5 : principe de fonctionnement d'un compteur électronique [1] 10

Figure I.6 : Compteur intelligent 11

Figure I.7 : illustration du principe général de l'IoT 11

Figure II.1 : Domaines d'application de l'IoT 20

Figure II.2 : Module nRF24L01 22

Figure II.3 : Pins du nRF24L01 24

Figure II.4 : Module Arduino Nano 26

Figure II.5 : Pins du Arduino Nano 27

Figure II.6 : Cablage Arduino Nano avec nRF24L01 28

Figure II.7 : Arduino IDE 29

Figure II.8 : Interface Arduino IDE 31

Figure II.9 : VSCode 32

Figure II.10 : Appwrite 34

Figure III.1: Architecture du système proposé 41

Figure III.2 : Organigramme de fonctionnement de notre système 43

Figure III.3 : Type de carte Arduino choisie 45

Figure III.4 : Ajouter Bibliothèque RF24 45

Figure III.5 : Installer bibliothèque RF24 sur Arduino IDE 46

Figure III.6 : Transmission des données sur les serials monitors 46

Figure III.7 : Communication Nœud-Sink [12] 47

Figure III.8 : Création de la base de données 49

Figure III.9 : Overview sur Appwrite 49

Figure III.10 : Importation les bibliothèques Node.js 50

Figure III.11 : Connexion de Node.js avec la base données Appwrite 50

Figure III.12 : Connexion d'unité de traitement avec la base de données 51

Figure III.13 : Création de la base de données sur Appwrite 51

Figure III.14 : Création de document Appwrite 52

Figure III.15 : Interface d'application web 53

Figure III.16 : Organisation de notre projet Reac 54

Figure III.17 : Bibliothèques utilisés sur le programme React 54

Figure III.18 : Chemin de navigation vers la page consommation 55

Figure III.19 : Bouton consommation 55

Figure III.20 : Bibliothèques de login 55

Figure III.21 : le choix de type d'écriture 56

Figure III.22 : Programmation de la page Login 56

Figure III.23 : Interface de la page Login 57

Figure III.24 : authentification d'utilisateur 57

Figure III.25 : Programmation de la page Home 58

Figure III.26 : Interface de l'accueil 59

Figure III.27 : Bibliographie pour la page consommation 59

Figure III.28 : Programmation de l'affichage de la valeur de consommation 60

Figure III.29 : Affichage de la valeur de consommation 60

Figure III.30 : Inscription au niveau de Cloud pour récupère la données 61

Figure III.31 : Importation des bibliothèques de la page historique 61

Figure III.32 : Récuperation database realtime 62

Figure III.33 : Affichage de l'historique 62

Figure III.34 : Plateforme web de déploiement l'application web 63

Figure III.35 : Réalisation du montage 64

Figure III.36 : Donnée envoyé par NRF slave 1 64

Figure III.37 : Donnée envoyé par NRF slave 2 64

Figure III.38 : Réception des données par Arduino Sink 65

Figure III.39 ; Connexion à la base de données 65

Figure III.40 : Login d'utilisateur 1 66

Figure III.41 : Consommation de user 1 66

Figure III.42 : Affichage de l'historique de user1 67

Figure III.43 : Login pour user2 67

Figure III.44 : valeur de consommation pour user2 68

Figure III.45 : Historique de consommation de user 2 68

Liste des tableaux

Tableau II-1 : Caractéristiques techniques du module nRF24L01[5] 23

Tableau II-2 : Description des pins du nRF24L01 [5]..... 24

Tableau II-3 : caractéristiques technique du Ardiono Nano 26

Introduction générale

Le monitoring énergétique est un processus qui permet de récolter les données énergétiques, de suivre les consommations, analyser pour permettre des optimisations et piloter l'installation pour maîtriser les consommations énergétiques.

La facturation et la collection des informations et des données de consommation d'énergie sont devenue un souci pour l'autorité sen raison de l'augmentation des compteurs d'énergie dans chaque ville. D'autre part la consommation quotidienne du client augmente d'une façon rapide.

En effet, un compteur intelligent (ou communicant) comptabilise une consommation d'énergie de manière précise et en temps réel, capable d'envoyer et de recevoir des données entre le consommateur et le distributeur. Aussi, il permettra de rationaliser la consommation énergétique en responsabilisant et en sensibilisant les gens sur l'impact de leur empreinte énergétique, du moment que l'utilisateur de compteur intelligent réalisé va voir en temps réel sa facture énergétique augmentée au fur et à mesure de l'augmentation de sa consommation.

L'utilisation des nouvelles technologies de l'information et de la communication sont essentielles pour améliorer les compteurs d'énergie actuels, faciliter la facturation et contrôler la consommation du client.

L'Internet des objets, ou l'Internet of Things (IoT) est un système de dispositifs informatiques interconnectés, de machines mécaniques et numériques, d'objets, qui sont pourvus d'identificateurs uniques et de la capacité de transférer des données sur un réseau sans nécessiter des interconnexions homme-machine. Aujourd'hui l'IoT est anticipé avec de nouveaux domaines d'applications comprenant la surveillance, la sécurité, la santé, les maisons et villes intelligentes ainsi que les systèmes de logistique et de transportation intelligents.

Dans ce contexte, le projet vise à développer une solution intelligente à base d'IoT pour améliorer les compteurs d'énergie actuels, faciliter la facturation et contrôler la consommation du client. La solution basée sur une carte électronique embarquée « module NRF et Arduino Nano » installée à côté de compteur d'énergie actuel permet d'envoyer des informations sur la consommation d'énergie du client vers un serveur où ces informations peuvent être

stockées dans une base de données « Appwrite » pour garder un historique qui pourrait être utilisé dans d'autres projets pour lequel les données sont considérées comme élément vital et indispensable, et une application web pour connecter le fournisseur (SONELGAZ), le client et le compteur. L'utilisation de l'application web facilite l'émission / la réception des factures, le monitoring énergétique et l'augmentation de la sécurité (intervention à distance).

Le présent manuscrit est divisé en trois chapitres :

- **Le premier chapitre** présente quelques généralités sur les compteurs d'énergie et leurs principes de fonctionnement.
- **Le deuxième chapitre** présente le concept de l'Internet des Objets ainsi que les outils hardwares et softwares utilisés.
- **Le troisième chapitre** présente la conception et l'implémentation de notre Project.

Chapitre 1

Généralités sur les compteurs d’énergie

I.1 Introduction

Le compteur électrique est un dispositif indispensable pour mesurer la consommation d'électricité d'un logement, parc ou entreprise. Au fil du temps, les compteurs d'énergie électrique ont connu une évolution en passant par trois générations : le compteur électromécanique, le compteur électronique et le compteur intelligent.

Ce chapitre a pour objectif de présenter en détail cette évolution des compteurs d'énergie électrique en expliquant leur conception, leur principe de fonctionnement ainsi que les avantages et les inconvénients de chaque type de compteurs. Nous verrons ainsi que le compteur électromécanique, bien que simple et peu coûteux, est moins précis que le compteur électronique. Ce dernier, plus avancé, peut mesurer la consommation en temps réel et peut être utilisé avec les réseaux intelligents. Quant au compteur intelligent, il est équipé de technologies de communication avancées qui permettent une mesure de la consommation en temps réel et une communication bidirectionnelle entre l'utilisateur et le fournisseur d'énergie.

Ainsi, ce chapitre mettra en lumière les caractéristiques propres à chaque type de compteurs d'énergie électrique, tout en soulignant leur importance dans la gestion de la consommation d'électricité et la facturation des services énergétiques. En effet, il est essentiel de comprendre l'évolution des compteurs d'énergie électrique pour mieux appréhender les enjeux énergétiques actuels et futurs.

I.2 Historique des compteurs d'énergie électrique

Les compteurs d'énergie électrique ont été inventés à la fin du 19ème siècle pour répondre à la demande croissante d'électricité. Les premiers compteurs étaient électromécaniques et fonctionnaient à l'aide d'un disque rotatif, qui tournait en fonction de la quantité d'électricité consommée. Au fil du temps, les compteurs ont évolué, passant des compteurs électromécaniques aux compteurs électroniques, plus précis et fiables. Puis, avec l'avènement des réseaux intelligents, sont apparus les compteurs intelligents, capables de mesurer la consommation en temps réel et de communiquer avec le fournisseur d'énergie.

Aujourd'hui, les compteurs d'énergie électrique sont indispensables pour mesurer la consommation d'électricité des logements, des parcs ou des entreprises, et pour permettre une facturation précise des services énergétiques.

I.3 Compteur électromécanique

I.3.1 Définition

Les compteurs électromécaniques (Figure I.1) sont des dispositifs de mesure de la consommation d'énergie électrique qui fonctionnent à l'aide de mécanismes et d'instruments électromagnétiques. Ils utilisent généralement un disque rotatif qui tourne à une vitesse proportionnelle à la quantité d'énergie consommée. Le disque est connecté à un train d'engrenages qui transfère le mouvement à un compteur à rouleaux. Les chiffres sur les rouleaux représentent la consommation d'énergie en kilowattheures (kWh). Les compteurs électromécaniques sont les premiers compteurs d'énergie électrique inventés et ont été largement utilisés pendant de nombreuses années en raison de leur simplicité et de leur faible coût.[1]



Figure I.1 : Compteur électromécanique

I.3.2 Principe de fonctionnement

Le principe de fonctionnement des compteurs électromécaniques repose sur les lois de l'électromagnétisme, développées par le scientifique anglais Ferraris en 1885. Selon cette théorie, il est possible de créer un couple dans un élément rotatif en mouvement libre en présence de deux flux magnétiques et de deux écoulements qui se coupent à un angle déterminé. Dans le cas des compteurs électromécaniques, l'élément rotatif est le disque mobile qui est monté sur un axe tournant, et le flux résultant est généré par les enroulements de courant et de tension. Figure I.2 illustre l'architecture interne d'un compteur électromécanique

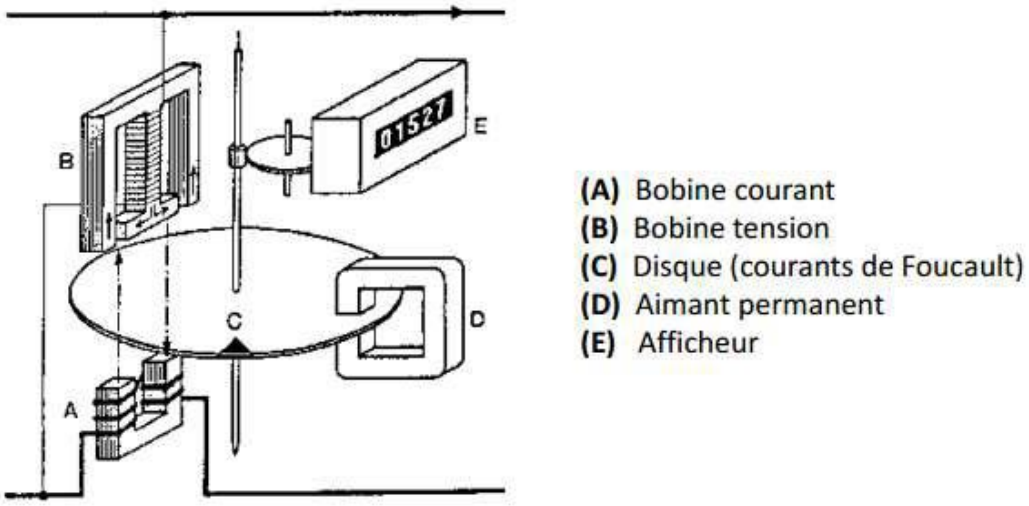


Figure I.2 : Architecture interne d'un compteur électromécanique

Le comptage de l'énergie électrique se fait en comptant le nombre de rotations du disque mobile. Ce dernier est soumis à des champs magnétiques alternatifs produits grâce à deux électroaimants disposés au-dessus et en dessous du disque. Chaque aimant est traversé par un courant électrique, l'un est parcouru par le courant circulant dans le fil de phase et l'autre par un courant proportionnel à la tension du réseau.

Le disque mobile opère un mouvement de rotation dont la vitesse est proportionnelle à la consommation d'électricité et indique la consommation d'électricité depuis sa mise en service. Pour les compteurs tripolaires, ils sont constitués des mêmes éléments que les

compteurs monophasés, mais avec deux ou trois bobines de tension et d'intensité. Leur équipement mobile est constitué soit d'un disque unique, soit de deux ou trois disques solidaires.[2]

Figure I.3 illustre une photo réelle de l'architecture interne d'un compteur électromécanique.



Figure I.3: Compteur électromécanique

I.3.3 Avantages et Inconvénients d'un compteur électromécanique

a- Avantages

- Haute fiabilité en cas de coupure de courant ou de court-circuit.
- Adapté aux réseaux de faible qualité.
- Prix d'achat moins élevé que celui des compteurs électroniques.
- Pas besoin de configuration supplémentaire, une installation correcte suffit.
- Dernière formation enregistrée reste affichée même en cas de détérioration ou de vandalisme.

- Longue durée de vie grâce à sa robustesse et sa qualité.
- Facilité d'installation et d'utilisation.

b- Inconvénients

- Petite classe de précision qui ne dépasse pas une valeur de 2 unités.
- Défaillances de suivi avec une forte variation des courants de charge.

I.4 Compteur électronique

I.4.1 Définition

Un compteur électronique est un type de compteur d'énergie qui utilise l'électronique pour mesurer et afficher la consommation d'électricité d'un bâtiment ou d'un équipement. Contrairement aux compteurs électromécaniques, les compteurs électroniques ne contiennent pas de pièces mobiles et fonctionnent sur la base de circuits électroniques intégrés. Ils mesurent la quantité d'énergie consommée à l'aide de capteurs électroniques et stockent ces informations numériquement pour un accès facile et précis. Les compteurs électroniques peuvent être monophasés ou triphasés et sont utilisés dans une variété de situations, allant des résidences individuelles aux grandes installations industrielles.[2]

Figure I.4 illustre une photo d'un compteur électronique.



Figure I.4 : Compteur électronique (numérique)

I.4.2 Principe de fonctionnement d'un compteur électronique

Le fonctionnement d'un compteur électronique consiste à mesurer la tension et le courant à l'aide de capteurs dédiés. Ces signaux sont ensuite multipliés pour obtenir la puissance instantanée, qui est intégrée et multipliée par le temps pour calculer l'énergie consommée en kWh. Les données mesurées sont stockées dans la mémoire du compteur électronique. Un voyant lumineux situé sur la façade du compteur permet de savoir s'il fonctionne correctement. Le clignotement du voyant varie en fonction de la quantité d'électricité consommée. [1]

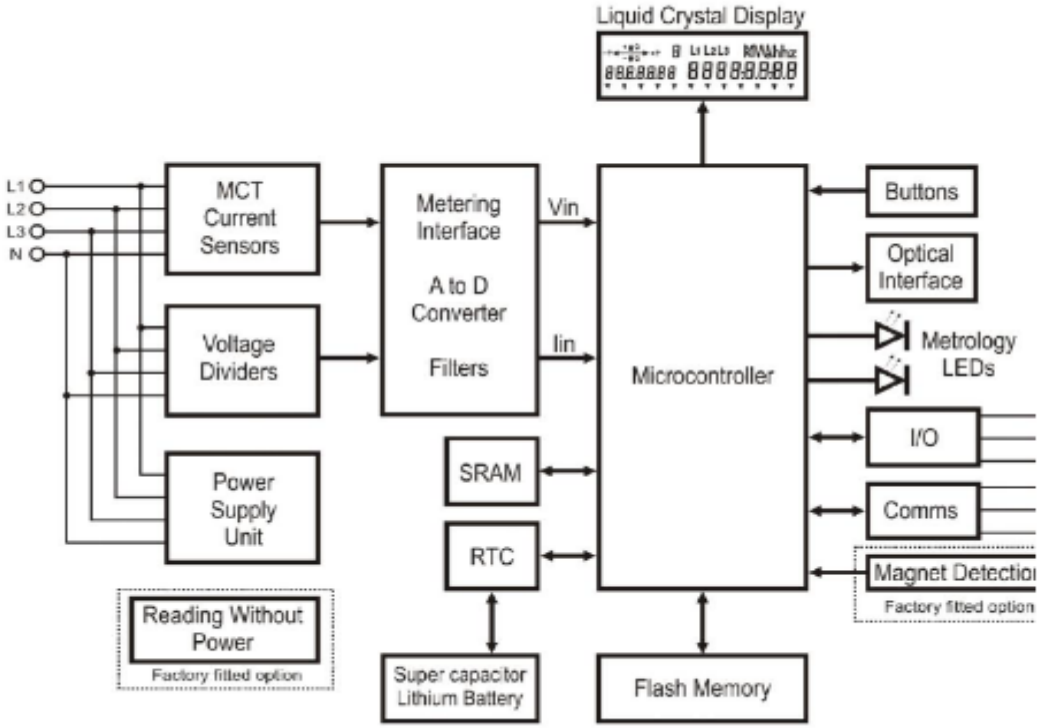


Figure I.5 : principe de fonctionnement d'un compteur électronique

I.4.3 Avantages et inconvénients d'un compteur électronique

a- Avantages

• Haute précision dans la mesure de l'énergie électrique consommée, même à faible consommation.

• Grande stabilité de mesure sur toute la plage de mesure.

• Grande capacité de stockage de données et d'enregistrement de la consommation d'énergie.

• Fonctionnalités avancées telles que la détection de pannes, la surveillance de la qualité de l'énergie et la mesure de la puissance réactive.

• Classe de précision élevée allant jusqu'à 0,2 unité.

b- Inconvénients

• Coût d'achat plus élevé par rapport aux compteurs électromécaniques.

• Vulnérabilité aux surtensions et aux pannes de courant, nécessitant souvent une maintenance et un remplacement réguliers.

• Dépendance à l'électricité pour fonctionner, ne fonctionne pas en cas de coupure de courant.

• Nécessite une configuration supplémentaire pour la connexion à un réseau de surveillance.

I.5 Compteur intelligent

I.5.1 Définition

Un compteur intelligent (Figure II.6), est un type de compteur d'énergie électrique qui est équipé de technologies de communication avancées. Contrairement aux compteurs électromécaniques et électroniques, un compteur intelligent mesure la consommation d'énergie en temps réel et de manière très détaillée. Il peut être utilisé pour enregistrer et transmettre des données de consommation, telles que les index, ainsi que pour recevoir des informations ou des ordres à distance. En outre, les compteurs intelligents peuvent être utilisés pour surveiller les habitudes de consommation d'énergie et aider les utilisateurs à économiser de l'énergie et de l'argent. [3]



Figure I.6 : Compteur intelligent

I.5.2 Histoires des compteurs intelligents

Les premiers projets de compteur intelligent datent des années 1970, lorsque les premiers ordinateurs personnels ont été mis sur le marché. Au début, ces compteurs étaient principalement utilisés dans le secteur industriel pour mesurer la consommation d'énergie et optimiser les processus de production.

Dans les années 1990, avec l'essor des technologies de l'information et de la communication, les compteurs intelligents ont commencé à se développer à grande échelle dans les réseaux électriques des villes. Les premiers modèles étaient encore limités en termes de fonctionnalités, mais ils permettaient déjà de mesurer la consommation d'énergie en temps réel et de transmettre les données aux fournisseurs d'énergie.

Au fil des ans, les compteurs intelligents sont devenus de plus en plus sophistiqués, avec l'ajout de fonctionnalités telles que la détection de pannes, la tarification dynamique et la gestion de la demande d'énergie. Ils sont désormais utilisés dans de nombreux pays du monde entier et sont considérés comme une technologie clé pour améliorer l'efficacité énergétique et réduire les émissions de gaz à effet de serre.

I.5.3 Principe du fonctionnement des compteurs intelligents

La conception et le principe de fonctionnement des compteurs intelligents peuvent varier en fonction des spécifications du protocole de communication et du nombre de fonctionnalités disponibles.

Cependant, en général, un compteur intelligent d'énergie fonctionne en mesurant la consommation d'électricité, et en envoyant les données collectées à un centre de traitement de données en temps réel. Le compteur est capable de communiquer avec le centre de traitement via différents types de protocoles de communication, tels que le GPRS, le CPL, IoT ou le Wi-Fi.

Le compteur intelligent peut également être équipé d'une gamme de fonctionnalités supplémentaires, telles que des alertes de consommation, des capacités de prépaiement, des outils de détection de fuites, des capteurs de température et des fonctions de contrôle à distance.

En résumé, la conception et le principe de fonctionnement d'un compteur intelligent dépendent du modèle spécifique du compteur, mais ils impliquent généralement la collecte et la transmission de données en temps réel à un centre de traitement de données, ainsi que la possibilité d'offrir des fonctionnalités supplémentaires pour une meilleure gestion de la consommation d'énergie.

I.5.4 Compteur intelligent à base d'IoT

Le compteur d'énergie intelligent IoT (Internet of Things) est une nouvelle génération de compteurs intelligents qui utilise la technologie de l'IoT pour fournir une surveillance et une analyse de la consommation d'énergie plus précises et plus avancées. Ces compteurs sont équipés de capteurs qui collectent des données en temps réel sur la consommation d'énergie et les transmettent via une connexion sans fil à un serveur distant pour une analyse plus poussée.

Le compteur d'énergie intelligent IoT utilise également des algorithmes d'apprentissage automatique pour apprendre les habitudes de consommation d'énergie des

utilisateurs et pour fournir des recommandations pour optimiser la consommation d'énergie et réduire les coûts.

La technologie IoT (Internet des objets) permet une communication en temps réel entre plusieurs appareils, permettant ainsi un grand nombre de fonctionnalités telles que le contrôle de l'ensemble des appareils de la maison avec une seule télécommande, la réduction de la consommation d'énergie et la prévention de problèmes électriques tels que les surcharges. [4]

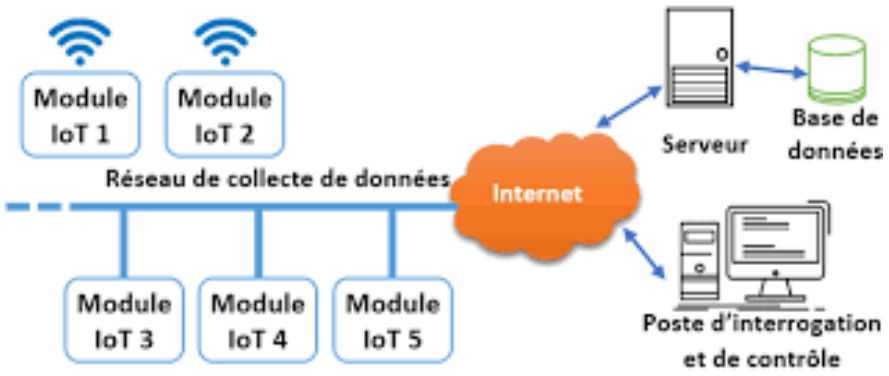


Figure I.7 : illustration du principe général de l'IoT

I.5.5 Avantages et inconvénients d'un compteur intelligent

Avantages :

- Il permet de relever et de contrôler la consommation électrique à distance, sans avoir besoin d'un agent sur place pour la lecture du compteur.
- enregistrer les consommations de manière détaillée et précise en temps réel, ce qui permet de mieux comprendre et maîtriser sa consommation d'énergie.
- Il permet de réaliser des économies d'énergie en adaptant ses habitudes, car le consommateur dispose d'informations plus détaillées et plus précises sur sa consommation.

- Il facilite la vie du consommateur en lui permettant d'être facturé en fonction de sa consommation réelle et non plus sur une estimation, et en évitant l'intervention d'un technicien pour la mise en service ou la modification de la puissance.
- Il permet au consommateur de suivre sa consommation exacte sur internet, pour mieux la maîtriser et réaliser des économies d'énergie.

Inconvénients :

• Certains utilisateurs peuvent se sentir envahis dans leur vie privée en raison de la collecte continue de données sur leur consommation électrique.

• Le coût d'installation et de maintenance des compteurs intelligents peut être élevé pour les gestionnaires et peut être transmis aux consommateurs via une augmentation des tarifs d'électricité.

• La connexion à distance des compteurs intelligents peut également poser des problèmes de sécurité et de protection des données, avec le risque que des personnes non autorisées puissent accéder aux données personnelles des consommateurs.

• Certains consommateurs peuvent ne pas être à l'aise avec l'idée de ne plus avoir de contrôle total sur leur consommation d'énergie, en particulier si leur fournisseur d'énergie a la capacité de réguler leur consommation à distance.

I.6 Conclusion

Dans ce chapitre, nous avons abordé l'évolution de la technologie utilisée dans la mesure de la consommation électrique, ainsi que les différents types de compteurs d'énergie électrique présents dans les installations électriques.

Nous avons également expliqué en détail le fonctionnement des compteurs, leurs avantages ainsi que les problèmes qui peuvent survenir.

Il est évident que le compteur d'énergie électrique est un élément crucial dans les réseaux et les installations électriques. Plus le compteur est précis et intelligent dans la

mesure, plus il contribuera à l'équité entre le consommateur et les sociétés de distribution électrique, contrairement aux anciennes technologies de comptage.

Dans le prochain chapitre, nous détaillerons les outils utilisés dans notre projet pour réaliser le compteur intelligent.

Chapitre 2

Outils matériels et logiciels utilisés

II. Chapitre 02

II.1 Introduction

La réalisation de notre compteur intelligent basé sur l'IoT nécessite l'utilisation de plusieurs outils matériels et logiciels. L'objectif de ce chapitre est de présenter en détail les différents outils matériels et logiciels utilisés pour la réalisation de notre projet. Nous explorerons les composants matériels clés ainsi que les logiciels nécessaires à la mise en œuvre du compteur intelligent basé sur l'IoT. En examinant ces outils, nous mettrons en évidence comment leur combinaison harmonieuse contribue à la création d'un système fonctionnel et efficace.

II.2 Internet des objets (IoT)

Définition de l'Internet des objets

Le concept Internet des Objets (en anglais : Internet of Things ou IoT), selon l'union international des télécommunications, l'internet des objets (IoT) est une « infrastructure mondiale pour la société de l'information, qui permet de disposer de services évolués en interconnectant des objets (physique ou virtuels) grâce aux technologies de l'information et de la communication interopérables existantes ou en évolution ». D'un point de vue technique, l'IoT consiste en l'identification numérique directe et normalisée (adresse IP, protocoles SMTP, http...) d'un objet physique grâce à un système de communication sans fil qui peut être une puce RFID, Bluetooth ou Wifi.[8]

Caractéristiques de l'IoT

Les caractéristiques fondamentales de l'IoT sont les suivantes [9] :

1. Interconnectivité : Dans l'IoT, tous les objets peuvent être connectés à l'infrastructure mondiale des technologies de l'information et de la communication.
2. Services liés aux objets : L'IoT est capable de fournir des services spécifiques aux objets, prenant en compte les exigences propres à ces derniers, telles que la protection de la vie privée et la cohérence sémantique entre les objets physiques et virtuels qui leur sont associés. Pour fournir de tels services tout en respectant ces exigences, les

technologies utilisées devront évoluer à la fois dans le monde physique et dans le monde de l'information.

3. Hétérogénéité : Les dispositifs utilisés dans l'IoT sont hétérogènes, car ils utilisent des plateformes matérielles et des réseaux différents. Ils peuvent interagir avec d'autres dispositifs ou plateformes de services via différents réseaux.
4. Changements dynamiques : L'état des dispositifs, tels que leur état de veille/réveil ou leur état de connexion/déconnexion, change de manière dynamique, tout comme le contexte dans lequel ces dispositifs fonctionnent (emplacement, vitesse, etc.). De plus, le nombre de dispositifs peut également évoluer de manière dynamique.
5. Très grande échelle : Le nombre de dispositifs à gérer et qui communiqueront entre eux dans l'IoT sera au moins dix fois supérieur à celui des dispositifs actuellement connectés à Internet. La proportion de communications établies par des dispositifs par rapport à celles établies par des personnes deviendra nettement plus favorable aux dispositifs. La gestion des données générées et leur interprétation pour les besoins des applications deviendront donc des aspects critiques, notamment en termes de sémantique et de traitement efficace des données.

Domaines d'application de l'IoT

Les applications potentielles de l'IdO sont nombreuses et variées, pénétrant dans pratiquement tous les domaines de la vie quotidienne des individus, des entreprises et de la société dans son ensemble [9].

1. Ville intelligente : Les technologies de l'information et de la communication (TIC) sont utilisées pour améliorer les services urbains et réduire les coûts dans les villes intelligentes. Cela comprend la gestion des déchets, l'éclairage public intelligent, la gestion du trafic, la surveillance de l'environnement, etc.
2. Médecine intelligente : L'IoT trouve des applications dans le domaine médical, telles que l'aide à la médication, le suivi à distance des patients (rythme cardiaque, pression artérielle, etc.), l'identification des allergies et des médicaments administrés, la localisation des médecins et des patients dans les hôpitaux, etc.
3. Industrie intelligente : L'industrie intelligente, également connue sous le nom d'industrie 4.0, utilise l'IoT et les systèmes cyber-physiques pour créer des usines

intelligentes. Cela permet une production plus adaptable et une allocation plus efficace des ressources, ouvrant ainsi la voie à une nouvelle révolution industrielle.

4. Domotique : La domotique regroupe les technologies de l'électronique, de la physique du bâtiment, de l'automatisation, de l'informatique et des télécommunications pour centraliser le contrôle des différents systèmes d'une maison ou d'une entreprise. Cela comprend la gestion de l'énergie, l'optimisation de l'éclairage et du chauffage, la sécurité (alarme), les commandes à distance, etc.
5. Agriculture intelligente : L'agriculture intelligente utilise l'IoT pour cultiver des aliments de manière propre et durable. Les technologies de l'information et de la communication modernes sont appliquées à l'agriculture pour améliorer la productivité, la gestion de l'eau, la surveillance des cultures, etc.

Figure II.1 présente les domaines d'application de l'IoT.



Figure II.1 : Domaines d'application de l'IoT

Applications de l'IoT en compteur d'énergie

L'IoT offre de nombreuses applications dans le domaine des compteurs d'énergie, permettant une gestion plus efficace et intelligente de la consommation d'énergie. Voici quelques exemples d'applications de l'IoT dans les compteurs d'énergie :

1. Mesure en temps réel : Les compteurs d'énergie IoT permettent de mesurer la consommation d'énergie en temps réel, offrant ainsi une visibilité instantanée de la consommation énergétique. Cela permet aux utilisateurs de surveiller leur consommation et d'ajuster leur comportement en conséquence.
2. Suivi de la consommation : Les compteurs d'énergie IoT peuvent collecter et stocker des données sur la consommation d'énergie sur une période de temps donnée. Ces données peuvent être utilisées pour analyser les schémas de consommation, identifier les goulots d'étranglement et prendre des mesures pour optimiser l'efficacité énergétique.
3. Facturation précise : Les compteurs d'énergie IoT facilitent la collecte automatique des données de consommation, ce qui permet d'éliminer les erreurs de relevé manuel. Cela garantit une facturation précise en fonction de la consommation réelle, évitant ainsi les litiges liés à la facturation.
4. Gestion à distance : Les compteurs d'énergie IoT peuvent être gérés à distance, ce qui permet aux fournisseurs de services d'énergie de surveiller et de contrôler les compteurs à distance. Cela simplifie la gestion des compteurs et facilite les opérations de maintenance.
5. Intégration avec les réseaux intelligents : Les compteurs d'énergie IoT peuvent être intégrés aux réseaux intelligents (smart grids), permettant une communication bidirectionnelle entre le fournisseur d'énergie et le compteur. Cela facilite la gestion de la demande d'énergie, l'intégration des énergies renouvelables et la mise en œuvre de tarifications différenciées.

L'application de l'IoT dans les compteurs d'énergie offre de nombreux avantages, tels que la surveillance en temps réel, l'optimisation de la consommation d'énergie et la réduction des coûts. Elle contribue également à la transition vers une utilisation plus durable et efficace de l'énergie.

II.3 Partie Hardware

II.3.1 Présentation du module nRF24L01

Le module nRF24L01 (Figure II.2) est un dispositif de communication sans fil utilisé dans de nombreux projets d'Internet des objets (IoT) et d'autres applications nécessitant une transmission de données à courte distance. Il offre une solution fiable et économique pour établir des connexions sans fil entre différents nœuds d'un réseau. [5]

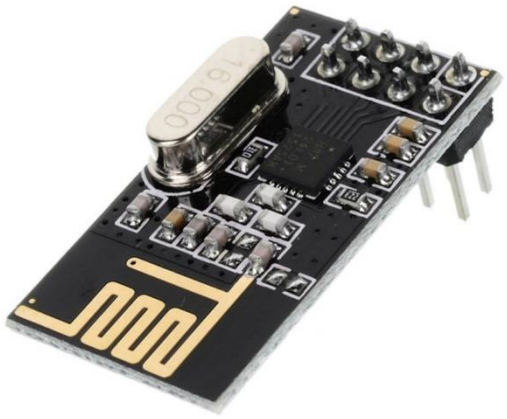


Figure II.2 : Module nRF24L01

II.3.2 Caractéristiques techniques du module nRF24L01

Le module nRF24L01 présente les caractéristiques techniques suivantes :

- **Protocole de communication**: Il utilise un protocole sans fil à faible consommation d'énergie basé sur des paquets de données.
- **Fréquence de fonctionnement**: Il opère dans la bande de fréquence de 2,4 GHz, ce qui le rend compatible avec d'autres normes sans fil comme Bluetooth et Wi-Fi.
- **Caractéristiques de transmission**: Il offre des options de puissance de transmission réglables, permettant d'ajuster la portée de communication. Il prend également en charge des taux de transfert de données allant jusqu'à 2 Mbps pour une transmission rapide et efficace.

- **Antenne**: Certains modules sont équipés d'une antenne intégrée, tandis que d'autres nécessitent une antenne externe pour étendre la portée de communication.
- **Interfaces**: Il utilise une interface de communication SPI (Serial Peripheral Interface) pour se connecter à d'autres dispositifs électroniques.
- **Faible consommation d'énergie**: Le module est conçu pour être économe en énergie, ce qui le rend adapté aux applications IoT alimentées par batterie.

Tableau II.1 présente les caractéristiques techniques du module Nrf24L01

Tableau II-1 : Caractéristiques techniques du module nRF24L01[5]

Caractéristiques	Description
Protocole de communication	Protocole sans fil à faible consommation d'énergie basé sur des paquets de données.
Fréquence de fonctionnement	2,4 GHz, compatible avec d'autres normes sans fil telles que Bluetooth et Wi-Fi.
Caractéristiques de transmission	Options de puissance de transmission réglables, taux de transfert de données jusqu'à 2 Mbps.
Antenne	Certains modules avec antenne intégrée, d'autres nécessitent une antenne externe.
Interfaces	Interface de communication SPI (Serial Peripheral Interface).
Faible consommation d'énergie	Conçu pour une faible consommation d'énergie, adapté aux applications alimentées par batterie.

II.3.3 Description des pins du nRF24L01

Tableau II.2 présente la description des pins du module Nrf24L01

Tableau II-2 : Description des pins du nRF24L01 [5]

BROCHE	SIGNIFICATION	RÔLE
VCC	–	Alimentation du module (1,9 et 3,6 V)
GND	–	Alimentation du module (0 V)
SCK	Serial ClocK	Horloge communication SPI
MISO	Master In Slave Out	Voie de communication nRF24 ->Arduino
MOSI	Master Out Slave In	Voie de communication Arduino -> nRF24
CE	Chip Enable	Active le mode RX ou TX (émission / réception)
CSN	Chip Select Not	Active le module nRF24, lorsque mis à la masse (port SPI)
IRQ	InterruptReQuest	Permet d'interagir avec un microcontrôleur, si besoin

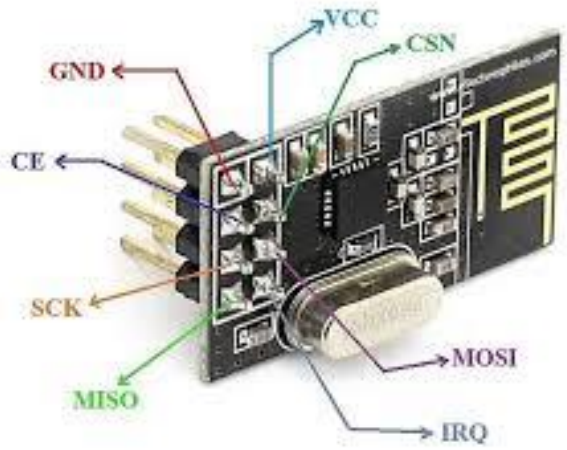


Figure II.3 : Pins du nRF24L01

II.3.4 Utilisations courantes du module nRF24L01

Le module nRF24L01 est largement utilisé dans différents domaines d'application, notamment :

- Domotique: Il est utilisé pour la communication sans fil entre les différents dispositifs domestiques intelligents.

- Suivi d'objets: Il permet de suivre et de localiser des objets à l'aide de capteurs sans fil.
- Contrôle à distance: Il est utilisé pour le contrôle à distance de divers appareils et systèmes.
- Réseaux de capteurs sans fil: Il permet la mise en place de réseaux de capteurs sans fil pour la collecte et le transfert de données.

II.3.5 Module Arduino nano

Carte Arduino

La carte Arduino est une plateforme de prototypage électronique open-source qui permet de créer des projets interactifs. Elle est composée d'un microcontrôleur programmable et de broches d'entrée/sortie qui permettent de connecter des composants électroniques tels que des capteurs, des actionneurs et des modules de communication. La carte Arduino est programmable à l'aide de l'IDE Arduino (Integrated Development Environment), qui utilise un langage de programmation simplifié basé sur le C/C++. Elle offre une interface conviviale pour écrire, télécharger et exécuter du code sur la carte Arduino. Les cartes Arduino sont utilisées dans de nombreux domaines, tels que l'électronique, l'automatisation, les objets connectés, l'Internet des objets (IoT), la robotique, l'art interactif, etc. Grâce à sa popularité, la communauté Arduino est très active et propose une vaste documentation, des exemples de code et des tutoriels pour aider les utilisateurs à réaliser leurs projets.[6]

Arduino Nano

Le module Arduino Nano (Figure II.4) est un composant électronique compact et polyvalent largement utilisé dans les projets électroniques et les applications IoT. Sa petite taille, sa facilité d'utilisation et sa compatibilité en font un choix populaire parmi les développeurs et les amateurs.



Figure II.4 : Module Arduino Nano

Caractéristiques techniques :

Tableau II.3 présente les caractéristiques techniques du module Arduino Nano

Tableau II-3 : caractéristiques technique du Ardiono Nano

Spécifications	Valeurs
Microcontrôleur	ATmega328P
Fréquence d'horloge	16 MHz
Mémoire flash	32 Ko (2 Ko utilisés par le bootloader)
Mémoire SRAM	2 Ko
Mémoire EEPROM	1 Ko
Entrées/sorties numériques	14 (6 avec sortie PWM)
Entrées analogiques	8
Interfaces de communication	UART, SPI, I2C
Tension d'alimentation	5 V
Courant de sortie maximum par broche	40 mA
Dimensions	18.5 mm x 43.2 mm

Description des pins du Arduino Nano

Voici la description des broches (pins) de la carte Arduino Nano [7] :

1. VCC : Alimentation. Connectez cette broche à la source d'alimentation +5V.
2. GND : Masse. Connectez cette broche à la masse (0V) de votre système.

3. AREF : Tension de référence analogique. Utilisée pour définir la tension de référence pour les entrées analogiques.
4. D0 à D13 : Broches d'entrées/sorties numériques. Ces broches peuvent être configurées en tant qu'entrées ou sorties numériques.
5. RXD : Broche de réception série. Utilisée pour recevoir des données en série.
6. TXD : Broche de transmission série. Utilisée pour envoyer des données en série.
7. D14 (A0) à D19 (A5) : Broches d'entrées analogiques. Utilisées pour mesurer des tensions analogiques.
8. RESET : Broche de réinitialisation. Utilisée pour réinitialiser le microcontrôleur.
9. GND : Broche de masse supplémentaire. Utilisée pour une connexion de masse supplémentaire si nécessaire.
10. Vin : Tension d'entrée. Cette broche peut être utilisée pour alimenter la carte Arduino Nano avec une tension externe de 7V à 12V.
11. AREF : Tension de référence analogique. Utilisée pour définir la tension de référence pour les entrées analogiques.

Figure II.5 présente le pins du module Arduino Nano.

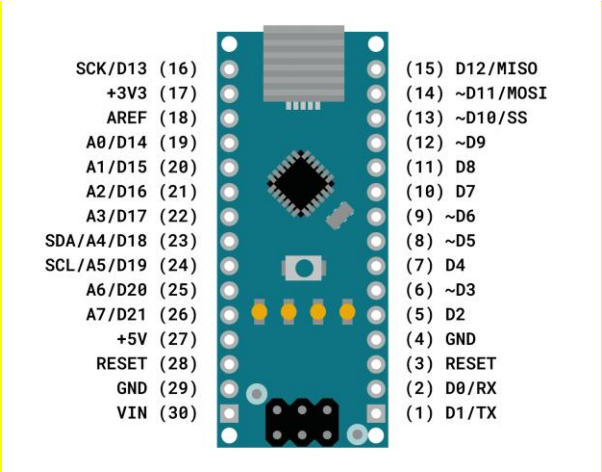


Figure II.5 : Pins du Arduino Nano

II.3.6 Communication Arduino Nano et nRF24L01

Le module NRF24L01 communique avec l'Arduino en utilisant le protocole SPI. Le module agit en tant qu'esclave SPI, ce qui signifie qu'il ne peut être utilisé qu'avec des appareils disposant de lignes de communication SPI dédiées. Cela signifie que les broches MOSI, MISO et SCK doivent être connectées à leurs broches correspondantes sur le microcontrôleur. Nous avons utilisé un Arduino Nano et ces broches sont les suivantes :

1. MOSI : Connectez cette broche du module NRF24L01 à la broche D11 de l'Arduino Nano.
2. MISO : à la broche D12 de l'Arduino Nano.
3. SCK : à la broche D13 de l'Arduino Nano.
4. CE : à la broche D9 de l'Arduino Nano.
5. CSN : à la broche D10 de l'Arduino Nano.
6. VCC : Alimentez cette broche du module NRF24L01 en utilisant une alimentation de 3,3V.
7. GND : à la broche de masse (GND) de l'Arduino Nano.

Figure II.6 présente le câblage du module Arduino Nano avec Nrf24L01

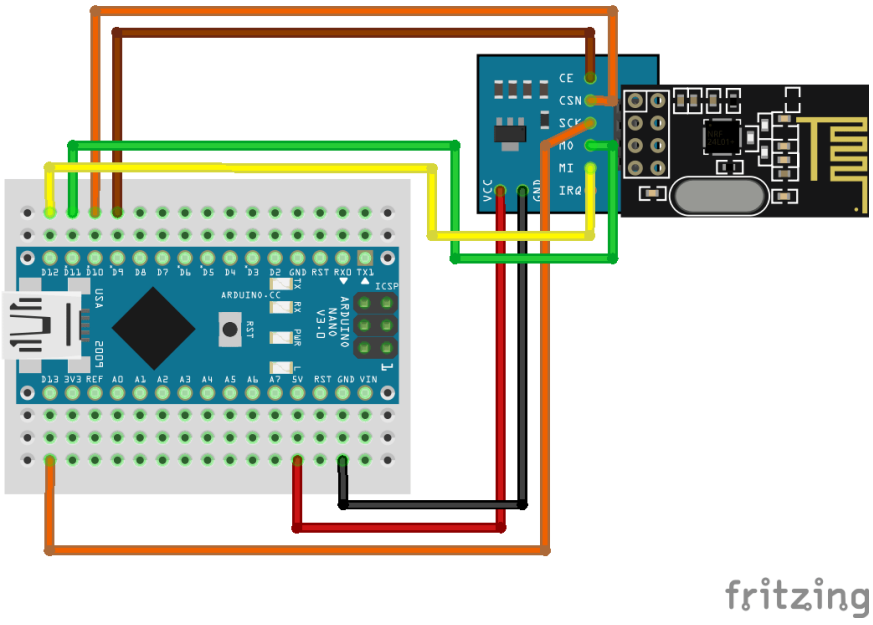


Figure II.6 : Cablage Arduino Nano avec nRF24L01

II.4 Partie Software

II.4.1 Arduino IDE

L'Arduino IDE (Integrated Development Environment) (Figure II.7) est un logiciel open-source utilisé pour programmer les cartes Arduino. C'est un environnement de développement convivial conçu spécifiquement pour faciliter la programmation et le téléchargement de code sur les cartes Arduino.

L'IDE Arduino propose une interface graphique simple et intuitive, ce qui en fait un outil populaire pour les débutants et les amateurs. Il est disponible pour différentes plates-formes, notamment Windows, macOS et Linux.

Avec l'IDE Arduino, vous pouvez écrire des programmes en utilisant le langage de programmation Arduino basé sur Wiring. Il comprend un éditeur de code avec des fonctionnalités telles que la coloration syntaxique, la complétion automatique et la vérification du code. Vous pouvez également compiler et télécharger facilement votre code sur une carte Arduino connectée via un câble USB.

En résumé, l'Arduino IDE est un logiciel essentiel pour développer des projets avec des cartes Arduino, offrant un environnement de programmation convivial et un moyen pratique de télécharger du code sur les cartes Arduino.



Figure II.7 : Arduino IDE

Interface Arduino IDE

L'IDE Arduino permet les fonctionnalités suivantes :

1. Édition de programme : L'IDE Arduino propose un éditeur de code où vous pouvez écrire vos croquis (sketches) Arduino. Les programmes sont rédigés en langage C, avec quelques fonctionnalités spécifiques au langage Arduino.
2. Compilation du programme : L'IDE Arduino prend en charge la compilation du programme écrit en langage C dans le langage "machine" spécifique à l'Arduino. La compilation traduit le code en instructions compréhensibles par le microcontrôleur de la carte Arduino. La console de l'IDE affiche des informations sur le déroulement de la compilation et les éventuels messages d'erreur.
3. Téléversement du programme : Une fois compilé, le programme peut être téléversé (uploadé) dans la mémoire de la carte Arduino. Cela se fait généralement via le port USB de l'ordinateur. Une fois que le programme est chargé dans la mémoire de l'Arduino, il est appelé micrologiciel (firmware). La console de l'IDE affiche des informations sur le déroulement du téléversement et les éventuels messages d'erreur.
4. Communication avec la carte Arduino : L'IDE Arduino permet de communiquer avec la carte Arduino via le terminal (ou moniteur série). Cela permet d'échanger des données entre l'Arduino et l'ordinateur pendant l'exécution du programme en mémoire sur l'Arduino. La connexion doit être active, généralement établie via un câble USB. Le terminal affiche les messages envoyés depuis l'Arduino et permet d'envoyer des commandes ou de recevoir des données de l'Arduino.

Ces fonctionnalités de l'IDE Arduino en font un outil puissant pour le développement et la programmation des cartes Arduino, offrant aux utilisateurs la possibilité d'écrire, compiler, téléverser et communiquer avec leurs programmes Arduino de manière pratique et conviviale.

Figure II.8 présente l’interface de l’Arduino IDE.



Figure II.8 : Interface Arduino IDE

II.4.2 VSCode

VSCode (Visual Studio Code) (Figure II.9) est un éditeur de code source polyvalent et populaire développé par Microsoft. Disponible gratuitement sur plusieurs plateformes, il est conçu pour être léger, flexible et riche en fonctionnalités. Grâce à sa popularité croissante, VSCode est devenu l'un des éditeurs de choix pour de nombreux développeurs. [10]



Figure II.9 : VSCode

Caractéristiques clés de VSCode

L'une des principales caractéristiques de VSCode est son interface utilisateur intuitive et conviviale. Il offre une expérience de développement agréable avec des fonctionnalités telles que la coloration syntaxique, la numérotation des lignes et une barre latérale pratique pour naviguer dans le code.

VSCode prend en charge de nombreux langages de programmation couramment utilisés, tels que JavaScript, Python, C++, Java, et bien d'autres. Cela en fait un choix polyvalent pour les développeurs travaillant sur différents projets.

Une autre caractéristique puissante de VSCode réside dans son écosystème d'extensions. Il dispose d'une vaste bibliothèque d'extensions qui permettent aux développeurs de personnaliser leur expérience de développement en ajoutant des fonctionnalités supplémentaires. Que ce soit pour le linting, le formatage automatique du code, les outils de test ou les intégrations avec des frameworks spécifiques, les extensions de VSCode offrent une flexibilité sans précédent.

Fonctionnalités avancées du VSCode

Parmi les fonctionnalités avancées de VSCode, on trouve un débogueur intégré qui permet de déboguer le code directement depuis l'éditeur. Il est également possible d'exécuter des commandes et des scripts directement à partir du terminal intégré, ce qui facilite les tâches de développement courantes.

Les développeurs web bénéficient également d'une expérience améliorée grâce à des fonctionnalités telles que l'aperçu en direct des modifications apportées au code HTML et CSS, ainsi que des extensions spécifiques pour ces langages.

Performances et optimisations du VSCode

VSCode est connu pour ses performances rapides et fluides, même avec des projets de grande taille. Cela est dû en partie à son moteur de rendu basé sur le rendu virtuel, qui n'actualise que les parties du code qui ont été modifiées. De plus, il gère intelligemment la mémoire, évitant ainsi les problèmes de performances liés à la consommation excessive de ressources.

En termes de productivité, VSCode propose des fonctionnalités de recherche et de remplacement avancées, ce qui permet aux développeurs de trouver rapidement et facilement des portions spécifiques de code, de les modifier et de les remplacer dans tout le projet.

En résumé, VSCode est un éditeur de code source polyvalent et populaire offrant une expérience de développement conviviale et flexible. Avec son écosystème d'extensions, ses fonctionnalités avancées, ses performances optimisées et sa personnalisation approfondie, il est devenu un choix populaire parmi les développeurs pour leurs projets de développement de logiciels.

II.4.3 Appwrite

Appwrite (Figure II.10) est une plateforme de développement backend open-source qui permet aux développeurs de construire plus rapidement et plus facilement des applications web et mobiles. Il offre une gamme complète de fonctionnalités backend prêtes à l'emploi, telles que l'authentification des utilisateurs, la gestion des bases de données, le stockage de fichiers et bien plus encore. [11]



Figure II.10 : Appwrite

Caractéristiques clés d'Appwrite

L'une des caractéristiques clés d'Appwrite est sa simplicité d'utilisation. Il fournit une API conviviale et bien documentée qui facilite l'intégration des fonctionnalités backend dans les applications. Les développeurs peuvent rapidement configurer et gérer des projets, des utilisateurs et des bases de données à l'aide de l'interface intuitive d'Appwrite.

Appwrite offre également une sécurité robuste avec des fonctionnalités d'authentification et d'autorisation intégrées. Il prend en charge plusieurs méthodes d'authentification, telles que l'authentification par e-mail/mot de passe, l'authentification par jeton et l'authentification sociale, ce qui permet aux développeurs de créer des systèmes d'identification personnalisés pour leurs applications.

Flexibilité et personnalisation

Appwrite est conçu pour être flexible et adaptable aux besoins spécifiques des développeurs et de leurs projets. Il prend en charge plusieurs bases de données populaires, notamment MySQL, MongoDB, PostgreSQL et Redis, permettant aux développeurs de choisir la solution qui convient le mieux à leurs besoins de stockage de données.

De plus, Appwrite est facilement extensible grâce à son architecture modulaire. Les développeurs peuvent créer et intégrer leurs propres plugins personnalisés pour ajouter des fonctionnalités supplémentaires à la plateforme et l'adapter aux exigences spécifiques de leurs projets.

Performances et optimisations

Appwrite est conçu pour offrir des performances élevées et une évolutivité. Il est basé sur des technologies modernes et utilise des mécanismes de mise en cache pour réduire la latence et améliorer les temps de réponse.

De plus, Appwrite propose des outils de surveillance et de journalisation intégrés, ce qui permet aux développeurs de surveiller les performances de leurs applications et de diagnostiquer rapidement les problèmes éventuels.

En résumé, Appwrite est une plateforme de développement backend puissante et flexible qui facilite la création d'applications web et mobiles. Avec ses fonctionnalités clés, sa facilité d'utilisation, sa sécurité solide et sa capacité d'extension, Appwrite est un choix intéressant pour les développeurs qui souhaitent accélérer leur processus de développement et offrir des fonctionnalités backend solides à leurs applications.

II.5 Conclusion

Les avancées technologiques et l'essor du numérique ont un impact significatif sur différents aspects de notre vie. Dans ce chapitre, nous avons examiné en détail les différents outils matériels et logiciels utilisés dans notre projet. Chacun de ces outils a été sélectionné en

fonction de ses caractéristiques et de ses avantages, tels que leur disponibilité, leur simplicité d'utilisation, leur support technique et leur coût.

. Dans le prochain chapitre, nous allons expliquer les différentes étapes de ce projet pratique et discuter des résultats que nous avons obtenus.

Chapitre 3

Conception et implémentation

III. Chapitre 03

III.1 Introduction

La révolution numérique a engendré un impact majeur dans tous les secteurs. Parmi les avancées significatives, les compteurs d'énergie intelligents se démarquent en permettant la collecte automatisée des données de consommation d'énergie, une communication bidirectionnelle entre les utilisateurs et les fournisseurs d'énergie, ainsi qu'une gestion plus efficace de la distribution énergétique.

L'objectif de ce chapitre est de présenter en détail l'implémentation technique et pratique de la solution proposée.

III.2 Description du système

Notre projet de compteur intelligent comprend trois parties distinctes :

- Collection de données : La première partie du projet consiste à collecter les données à partir du compteur intelligent en utilisant une carte électronique installée à côté du compteur. Cette carte est connectée à un serveur via la communication RF (Radio Fréquence). Cette étape permet d'obtenir des informations de consommation d'énergie précises et en temps réel.
- Stockage des données : La deuxième partie du projet concerne la conception et le développement d'une base de données pour stocker les données collectées dans le cloud. Cette base de données centralise les informations et les organise de manière sécurisée et accessible à tout moment.
- Suivi de la consommation : La dernière partie du projet consiste à concevoir une application web conviviale qui permettra aux fournisseurs et aux consommateurs de suivre facilement la consommation quotidienne d'énergie. Cette application fournira des visualisations, des statistiques et des alertes pour aider les utilisateurs à prendre des décisions éclairées concernant la consommation d'énergie.

III.3 Principe de fonctionnement de système proposé

Figure III.1 présente l'architecture générale de notre système novateur. Nous avons conçu une solution basée sur l'utilisation de cartes électroniques Arduino Nano, installées à côté de chaque compteur d'énergie dans les maisons des utilisateurs. Ces cartes permettent de récupérer les données de consommation d'énergie du client de manière précise. Les données sont ensuite transférées vers un nœud collecteur, appelé "Sink", en utilisant la communication RF via le module nRF24L01.

Le Sink joue un rôle essentiel en transférant ces données vers un serveur (Raspberry Pi ou PC) via une communication série. Les données de consommation sont stockées de manière sécurisée dans la base de données APPWRITE, qui représente notre solution de cloud. Parallèlement, nous avons développé une application web conviviale basée sur la plateforme React, permettant aux utilisateurs d'accéder facilement à leurs données de consommation individuelles. L'application présente ces données sous forme de jauges et de graphiques, offrant une visualisation claire de l'évolution des mesures récentes.

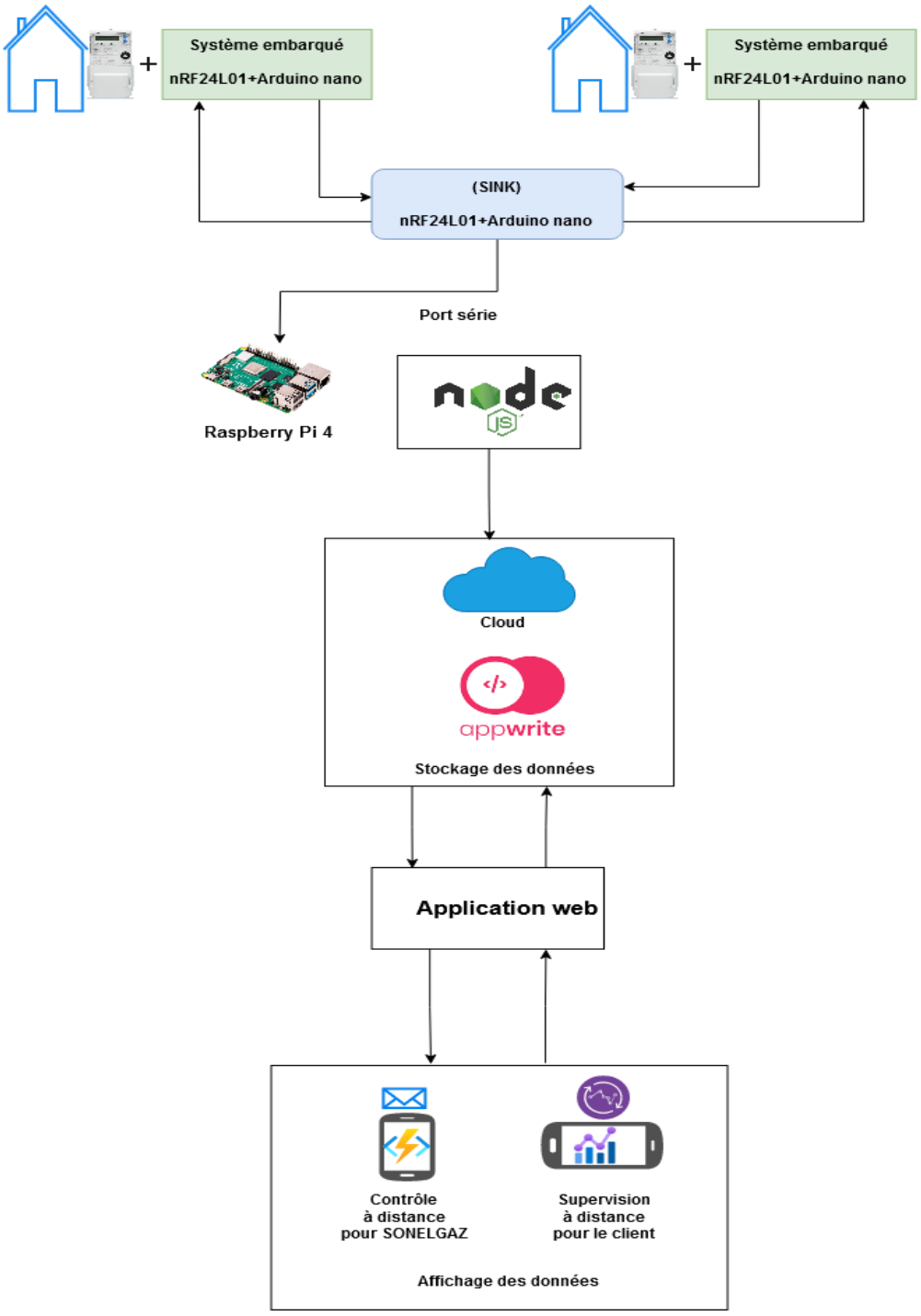
Notre application offre une multitude de fonctionnalités avancées. Chaque minute, une requête est soumise à la base de données pour récupérer les dernières données à afficher. Nous utilisons des nœuds créés avec l'environnement de développement VSCODE pour assurer une communication fluide avec la base de données. L'interface graphique de l'application web est conviviale et intuitive, permettant aux utilisateurs de surveiller facilement leur consommation quotidienne d'électricité et de prendre des mesures de contrôle pour optimiser leur utilisation énergétique.

L'application propose également une analyse des tendances, grâce à des graphiques et des diagrammes permettant aux utilisateurs de visualiser les fluctuations de leur consommation d'énergie sur une période donnée. Ils peuvent ainsi identifier les périodes de consommation élevée ou anormale, et ajuster leur comportement en conséquence pour des économies d'énergie significatives.

Pour une gestion encore plus personnalisée, les utilisateurs ont la possibilité de définir des seuils de consommation personnalisés et de recevoir des alertes lorsque ces seuils sont approchés ou dépassés. Cela leur permet de rester informés en temps réel et de prendre des mesures proactives pour optimiser leur consommation d'énergie.

Enfin, notre application offre la fonctionnalité de comparaison de la consommation, permettant aux utilisateurs de comparer leur consommation d'énergie actuelle avec des périodes précédentes. Cela leur permet de suivre leur progrès, d'identifier les améliorations réalisées et de renforcer leur engagement en matière de gestion énergétique.

Pour une analyse plus approfondie ou pour partager les informations avec d'autres parties prenantes, notre application permet également l'exportation des données de consommation dans différents formats tels que CSV ou Excel.



FigureIII.1: Architecture du système proposé

III.4 Solutions hardware et software

III.4.1 Hardware

Notre système utilise les composants matériels suivants :

1. Un module de capture des informations :
    - Un Arduino Nano connecté aux compteurs pour collecter les données de consommation.
2. Un module de communication :
    - Nous utilisons deux types de liaisons :
        - Une communication série entre le nœud collecteur (Sink) et le serveur.
        - Une communication sans fil (RF) entre le Sink et les nœuds.
3. Les nœuds :
    - Chaque nœud est constitué d'un Arduino Nano en tant qu'esclave et d'un module nRF24L01+.
    - Leur rôle est de communiquer avec le Sink en utilisant une communication radiofréquence.
4. Le nœud de collecte des données (Sink) :
    - Il est composé d'un Arduino en tant que maître et d'un module nRF24L01+.
    - Son rôle est de recevoir les données provenant des différents nœuds et de les transférer vers l'unité de traitement du PC en utilisant la communication série.

III.4.2 Software

Notre système utilise les technologies logicielles suivantes :

- JavaScript : Utilisé pour le développement de la logique côté client dans l'application web.
- HTML : Utilisé pour la structuration et la création des éléments de contenu dans l'application web.

- VS Code : Utilisé comme environnement de développement intégré (IDE) pour écrire et éditer le code source des différents composants du système.
- Appwrite : Une plateforme de développement qui fournit une base de données et des fonctionnalités de gestion des données pour stocker et récupérer les informations nécessaires dans le système.

III.5 Algorithmes

Avant de passer à l'implémentation, nous devons réaliser un organigramme (Figure III.2) qui explique le déroulement des différentes séquences de notre système proposé.

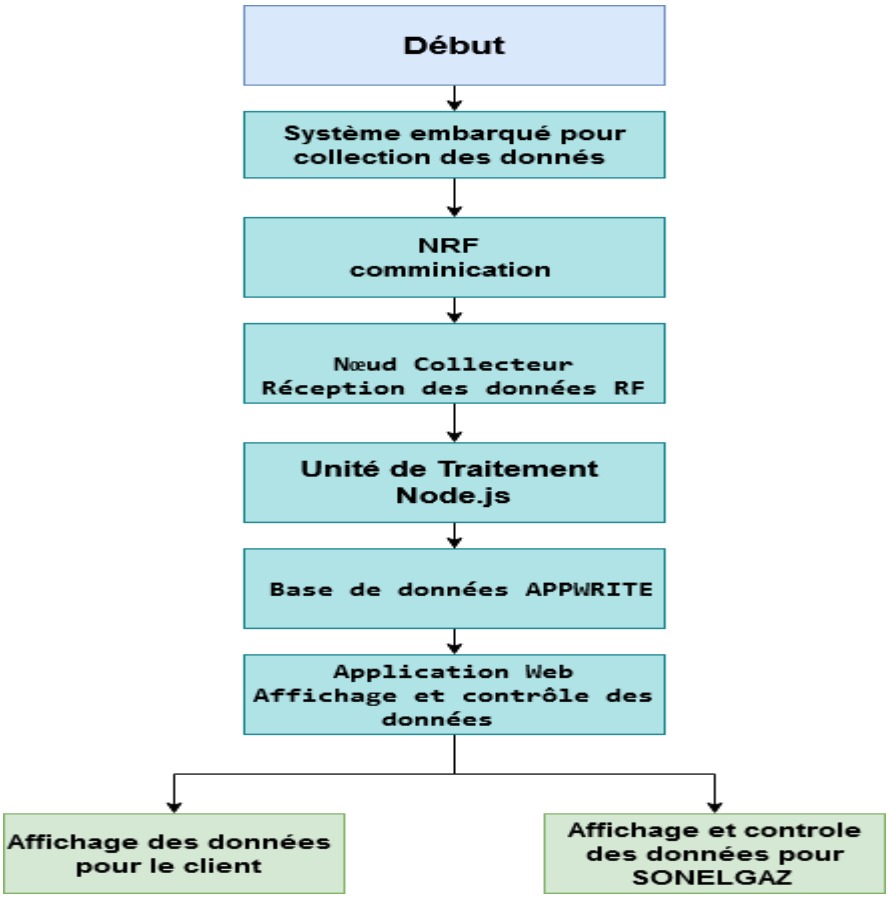


Figure III.2 : Organigramme de fonctionnement de notre système

III.6 Implémentation du système

III.6.1 Captage et transfert des données

Dans notre projet, nous avons utilisé l'Arduino Nano et le module nRF24L01+ comme des nœuds pour collecter les données. De plus, nous avons utilisé un autre Arduino Nano connecté à un NRF en tant que nœud collecteur (sink).Ainsi, nous avons pris en considération deux nœuds de collecte, un pour chaque compteur, afin de réaliser les tests.

A) Programmation de la carte Arduino Nano

Pour programmer le module nRF24L01 avec l'Arduino IDE, les étapes suivantes doivent être suivies :

- Connecter physiquement le module nRF24L01+ à l'Arduino Nano en respectant le schéma de câblage mentionné précédemment. Assurez-vous que toutes les connexions sont correctes et sécurisées.
- Ouvrir notre Arduino IDE sur l'ordinateur.
- Dans l'IDE, sélectionner le type de carte "Arduino Nano" dans le menu "Outils" -> "Type de carte". Assurez-vous également de sélectionner le port série approprié pour l'Arduino Nano dans le menu "Outils" -> "Port".
- Une fois ces étapes suivies, nous pouvons commencer à programmer le module nRF24L01+ avec l'Arduino IDE pour permettre la communication entre les nœuds de capteurs et le nœud de collecte des données (Sink). Cette configuration permettra le transfert efficace des données de consommation d'énergie vers l'unité de traitement du PC pour un traitement ultérieur. Figure III.3 montre comment choisir le type de la carte Arduino.

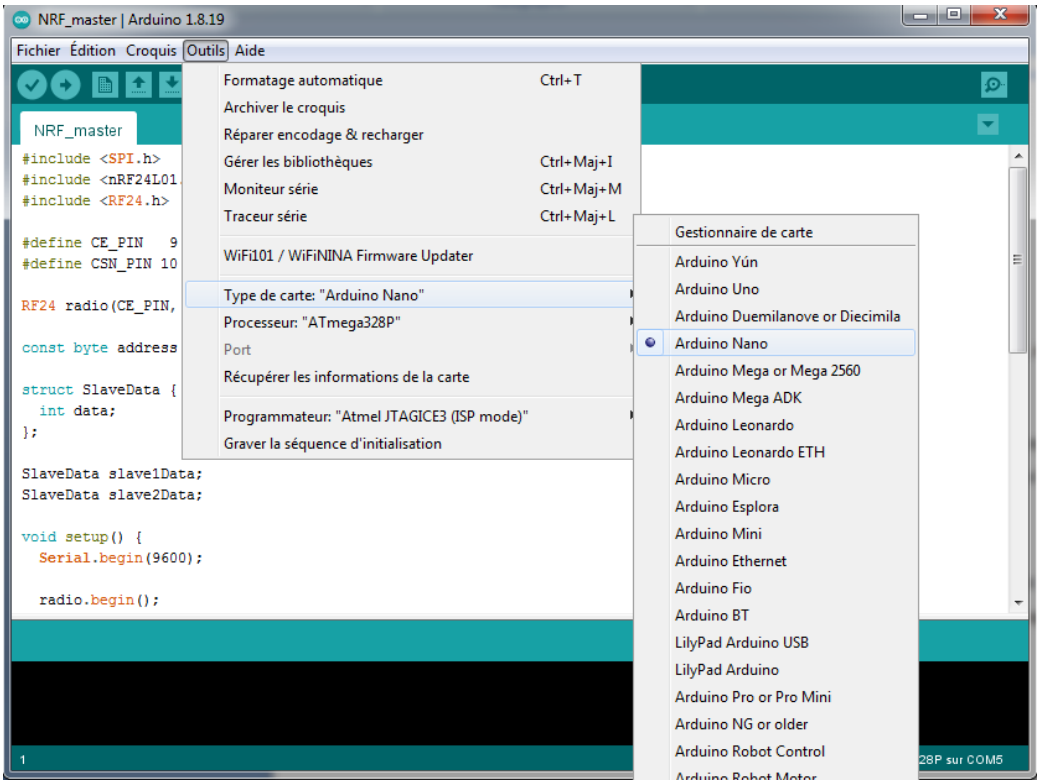


Figure III.3 : Type de carte Arduino choisie

- Avant de pouvoir utiliser le module NRF24L01, on doit installer la bibliothèque RF24 dans l'Arduino IDE (Figure III.4).

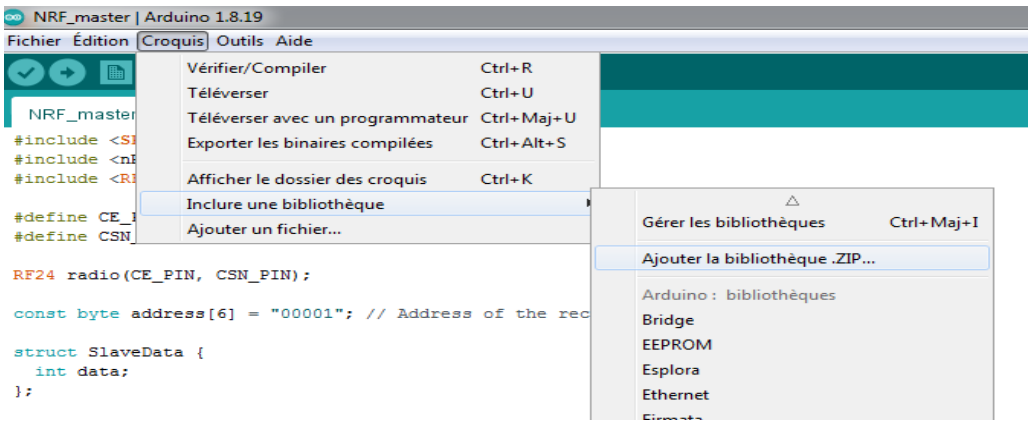


Figure III.4 : Ajouter Bibliothèque RF24

- Dans la fenêtre de gestion des bibliothèques, nous recherchons "RF24" et cliquons sur le résultat correspondant. Ensuite, nous cliquons sur le bouton "Installer" pour installer la bibliothèque RF24 (Figure III.5).

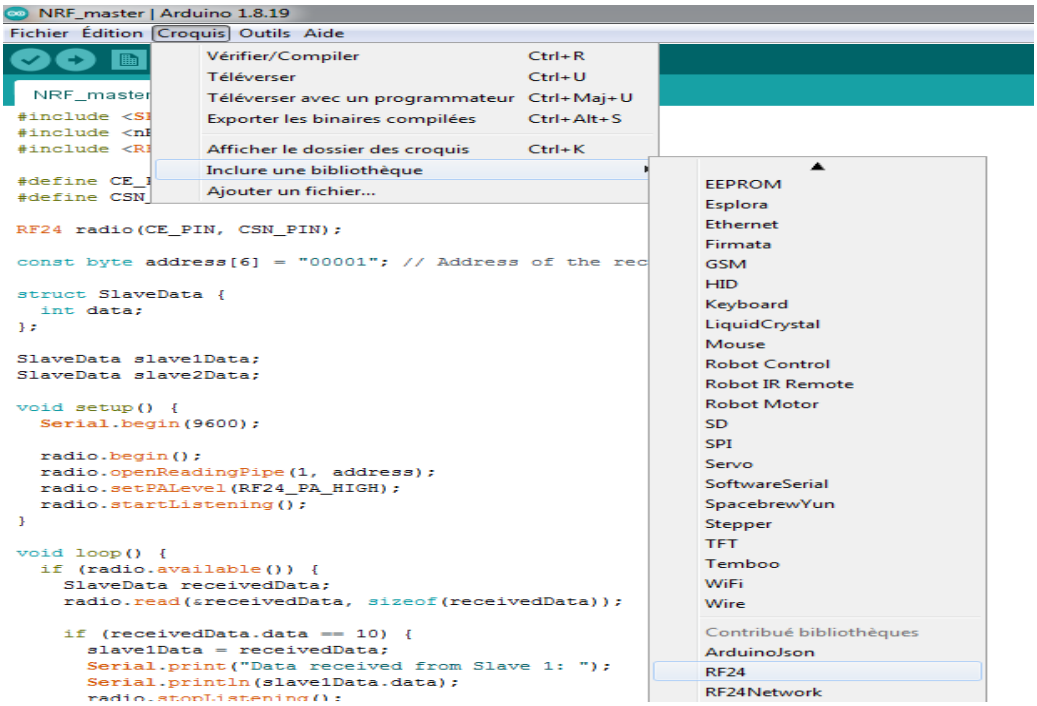


Figure III.5 : Installer bibliothèque RF24 sur Arduino IDE

- Téléverser le code sur Arduino Nano en cliquant sur le bouton de téléversement (flèche vers la droite) dans l'IDE.
- Une fois le téléversement terminé, on ouvre le "Console"(Figure III.6) de l'IDE pour voir les messages de sortie. Cela nous permettra de vérifier si la communication entre l'Arduino Nano et le module NRF24L01 fonctionne correctement.

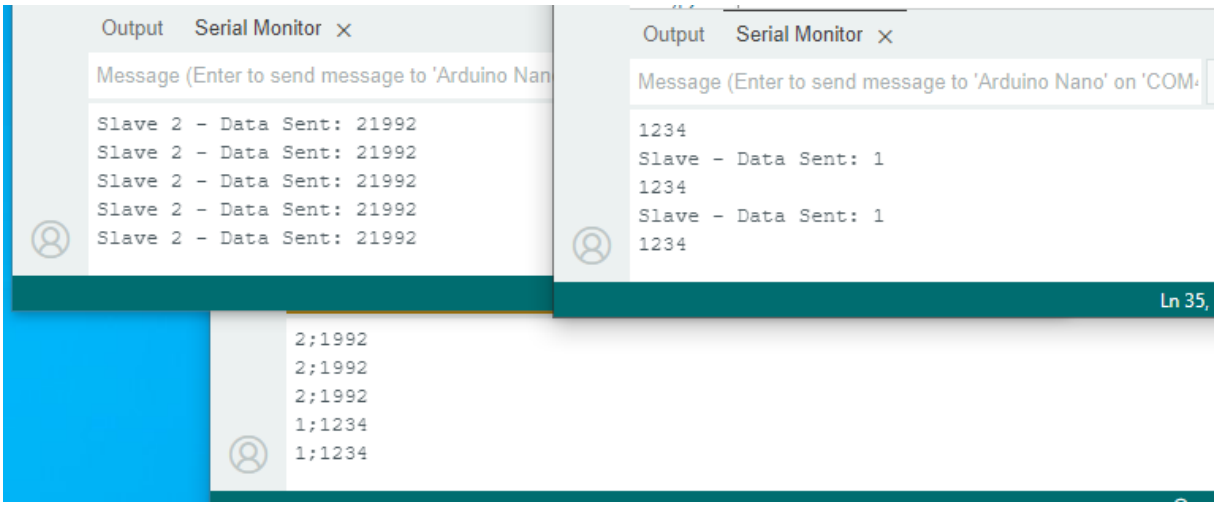


Figure III.6 : Transmission des données sur les serials monitors

B) Protocole de communication

➢ Communication Nœuds-Sink

Dans notre projet on a adopté le mode de transaction avec accusé comme solution efficace de la problématique posée, nous exploitons l'accusé de réception (ACK) pour transmettre les données des deux noeuds installés au Sink (ça permet au noeud de communique en deux modes et en même temps) autrement, il fait l'émission et la réception en même temps. Le principe de ce scénario est illustré sur Figure III.8. Le Sink commence la communication avec les noeuds un par un sur le même pipe où chaqu'un possède une adresse. Le Sink commence le premier cycle de communication en envoyant un paquet de données (TX : PID1_1) au premier noeud (Noeud1). Une fois que ce dernier reçoit tout le paquet, il attend 130 µs puis envoie un accusé de réception (ACK1_1) au Sink. Après la réception du premier accusé le Sink commence la communication avec l'autre nœud[12].

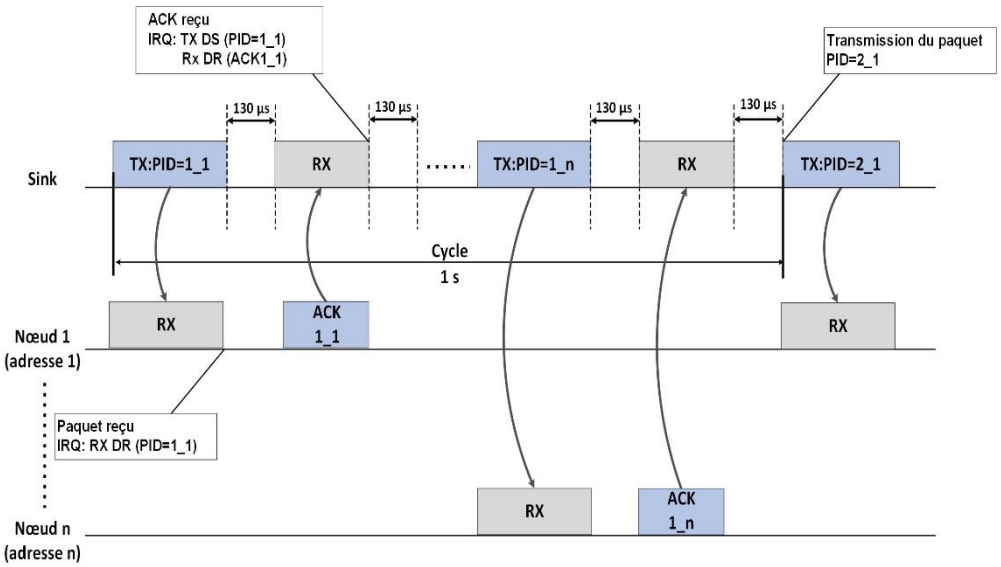


Figure III.7 : Communication Nœud-Sink [12]

➢ Communication Sink-Unité de traitement

Comme mentionné dans notre projet, nous avons utilisé la communication série directe entre le Sink (Arduino Nano) et l'unité de traitement (Raspberry Pi). Le principe de la

communication série repose sur l'utilisation d'un câble USB de type A vers micro USB (ou mini USB).

Le câble USB de type A vers micro USB (ou mini USB) est utilisé pour connecter la prise USB de type A à l'unité de traitement (Raspberry Pi) et la prise micro USB (ou mini USB) au port USB du Sink (Arduino Nano). Ce câble USB assure à la fois le transfert des données et la fourniture d'une source d'alimentation pour le Sink. Lorsqu'ils sont connectés via ce câble USB, le Raspberry Pi envoie les données au Sink à travers les lignes de transmission série (RX/TX) du câble. De plus, le câble USB fournit également l'alimentation nécessaire au bon fonctionnement du Sink, éliminant ainsi le besoin d'une source d'alimentation externe distincte pour le Sink.

Cette configuration simplifie la connexion et l'alimentation du Sink, tout en permettant le transfert des données entre l'Arduino Nano et le Raspberry Pi via la communication série.

III.6.2 Création de la base de données

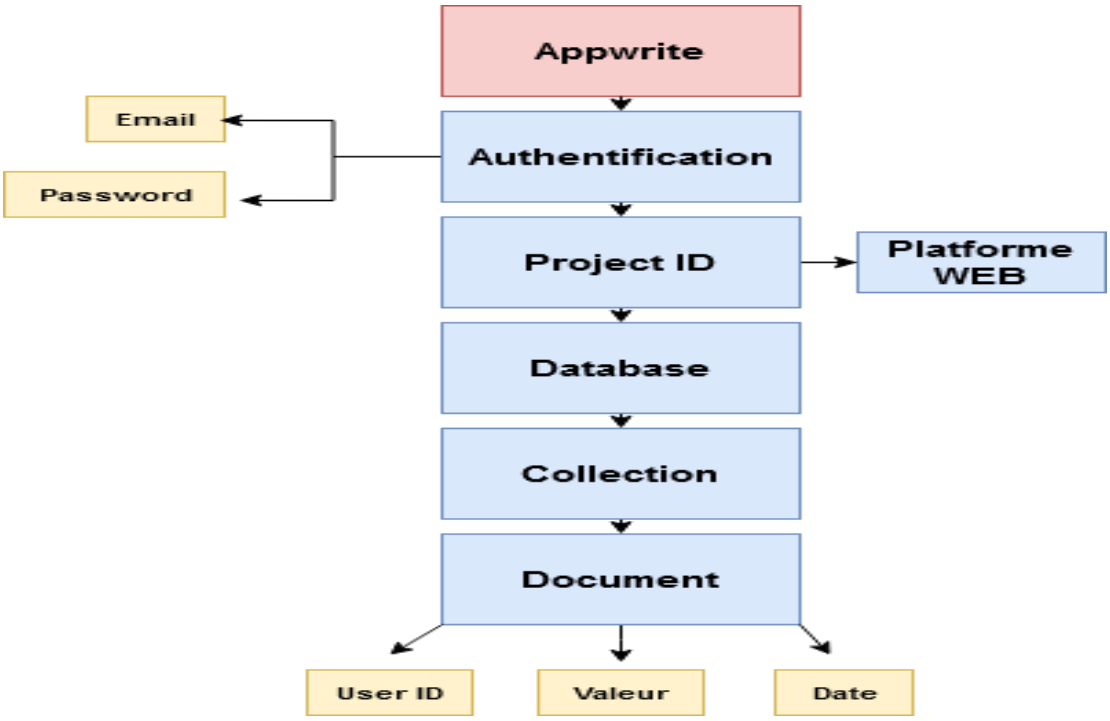
La création de la base de données est une étape essentielle de notre projet. Son objectif est de stocker les informations collectées par le biais de divers systèmes embarqués installés sur différents compteurs. Cette base de données nous permettra de faciliter l'interprétation de ces informations ultérieurement.

Nous avons choisi d'utiliser APPWRITE comme base de données en raison de ses avantages, notamment la possibilité de créer un serveur local.

Pour créer une base de données avec APPWRITE, nous suivons les étapes suivantes :

- Création d'un identifiant de projet (Project ID).
- Création d'une plateforme web.
- Création d'une base de données.
- Création d'une collection qui contient des documents avec deux champs : ID et valeur.

Figure III.8 illustre un organigramme de la création de la base de données



FigureIII.8 : Création de la base de données

A) Aperçu sur la base de données

Figure III.9 présente l'interface du la base de données Appwrite.

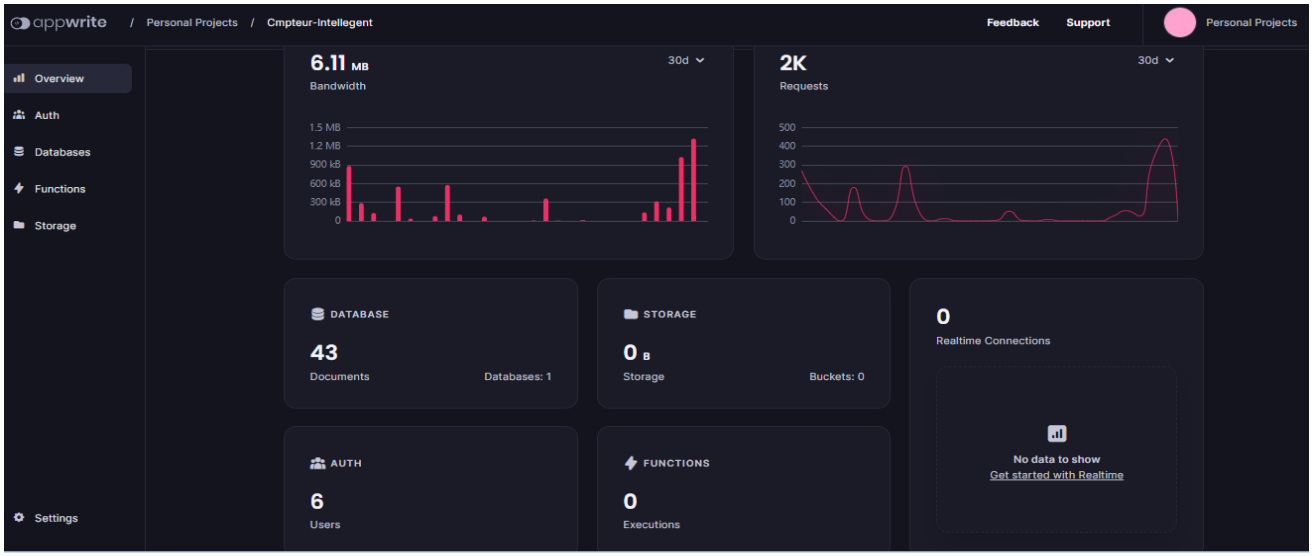


Figure III.9 : Overview sur Appwrite

B) Connexion l'unité de traitement avec le cloud

Pour établir la communication entre l'unité de traitement (serveur) et le cloud (Appwrite), nous avons programmé en Node.js en suivant les étapes suivantes :

1. Configuration de l'environnement : Nous avons installé Node.js sur l'unité de traitement (serveur) pour exécuter des applications JavaScript côté serveur. Cela a impliqué l'installation de Node.js et de son gestionnaire de paquets, npm.
2. Importation des bibliothèques nécessaires : Nous avons importé les bibliothèques requises dans notre application Node.js pour commencer à programmer (Figure III.10).

```
// BibliothèqueS

import { SerialPort } from 'serialport';
import { DelimiterParser } from '@serialport/parser-delimiter'
import { Client, Databases } from 'node-appwrite';
import { nanoid } from 'nanoid';
```

Figure III.10 : Importation les bibliothèques Node.js

3. Connexion de Node.js à la base de données Appwrite : Nous avons mis en place la connexion entre notre application Node.js et la base de données Appwrite. Cela implique de fournir les informations d'authentification appropriées, telles que l'URL de l'API Appwrite et les clés d'API nécessaires.(Figure III.11)

```
// connecter le noeud avec base de données "appwrite"

const client = new Client()
    .setEndpoint('https://cloud.appwrite.io/v1') // Your Appwrite Endpoint
    .setProject('6467c41de32e285efb9a')              // Your project ID
    .setKey('d44b45bec2954d03f473696cb668b6929a2d6aeda95be375657cf850fd6cd666d89f0da357a2ac876077be46858456c1fe96ce0dfe9d8aeedbaa4a3a36b26e3a9317
```

Figure III.11 : Connexion de Node.js avec la base données Appwrite

4. Connexion de l'unité de traitement (PC ou Raspberry Pi) à la base de données : Nous avons également établi la connexion entre notre unité de traitement et la base de données Appwrite. Cela peut être réalisé en utilisant les fonctionnalités de l'Appwrite SDK appropriées pour la communication avec la base de données (Figure III.12)

```
// connection de serveur "unite de traitement" "Rasberie PI" avec  la base de données

    const databases = new Databases(client);

const port = new SerialPort({ path: 'COM3', baudRate: 9600 })

const parser = port.pipe(new DelimiterParser({ delimiter: '\n' }))
parser.on('data', function (data) {
const id = data.toString().split(";")[0] ?? null;
const value = data.toString().split(";")[1] ?? null;

    console.log(id, value);

    if (id && value) {
        writeData(databases, id, value);
    }
}) // emits data after every '\n'

// Open errors will be emitted as an error event
port.on('error', function (err) {
    console.log('Error: ', err.message)
})
```

Figure III.12 : Connexion d'unité de traitement avec la base de données

5. Création d'une base de données dans Appwrite : Nous avons créé une base de données dans Appwrite pour stocker les données pertinentes. Cela peut impliquer la définition des schémas, des collections et d'autres configurations nécessaires pour la base de données(Figure III.13)

```
// crée une document au niveau de base de données "appwrite" qui contient le ID de utilisateur et la valeur de consommation

const writeData = (db, uid, value) => {
    const promise = db.createDocument('646a224b9c7d1e4608a1', '648f45a9357a3663f0e4', nanoid(12), { uid: uid, value: value, date: new Date().toIS

    promise.then(function (response) {
        console.log(response);
    }, function (error) {
        console.log(error);
    });
}
```

Figure III.13 : Création de la base de données sur Appwrite

La Figure III.14 illustre que nous avons réussi à établir une connexion à la base de données Appwrite et que nous sommes en mesure d'écrire des données sur Appwrite et de les stocker.

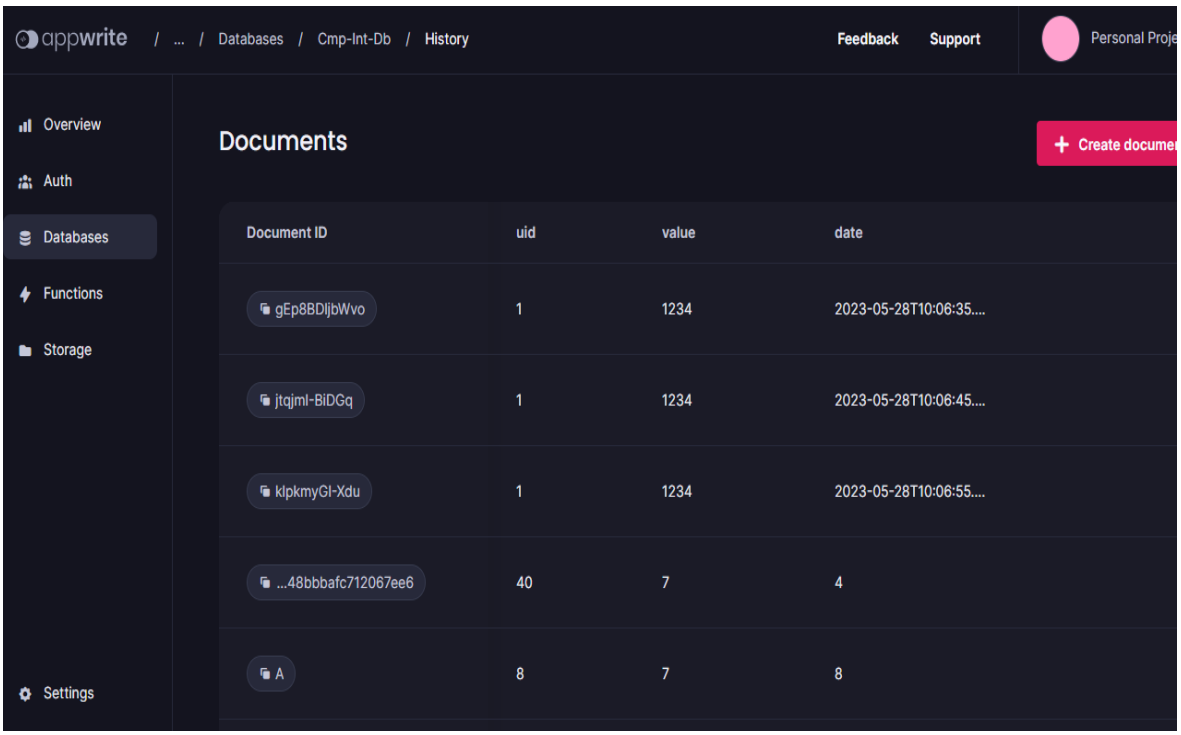


Figure III.14 : Création de document Appwrite

III.6.3 Interface Homme Machine IHM

A) Application web

La Figure III.15 présente l'interface de notre application web, où le consommateur/fournisseur peut visualiser la consommation d'énergie ainsi que l'historique de consommation sur une certaine période de temps.

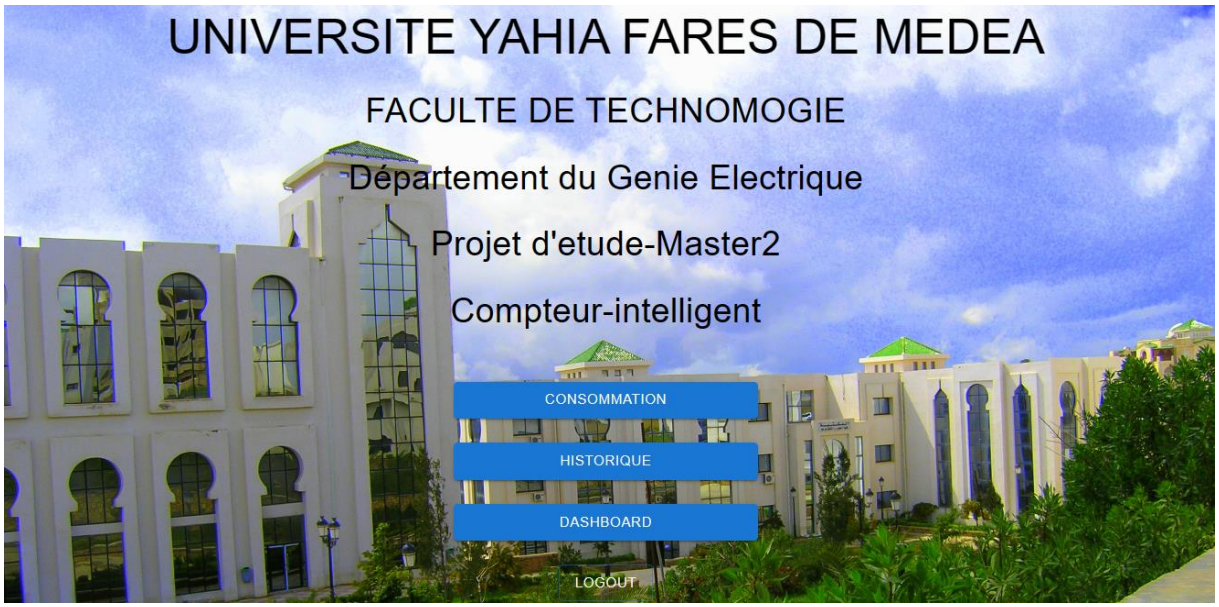


Figure III.15 : Interface d'application web

B) Organisation de notre projet React

La Figure III.16 présente l'organisation de la création de notre interface pour l'application web. Dans le cadre de notre projet React, nous avons suivi une structure bien définie pour organiser la création de notre interface. Nous avons mis en place les différentes parties de notre projet en suivant cette structure.

En utilisant cette approche d'organisation, nous facilitons le développement, la maintenance et la compréhension de notre projet React. Chaque fonctionnalité de notre application a été placée dans des composants spécifiques, ce qui favorise la modularité et permet une séparation claire des responsabilités.

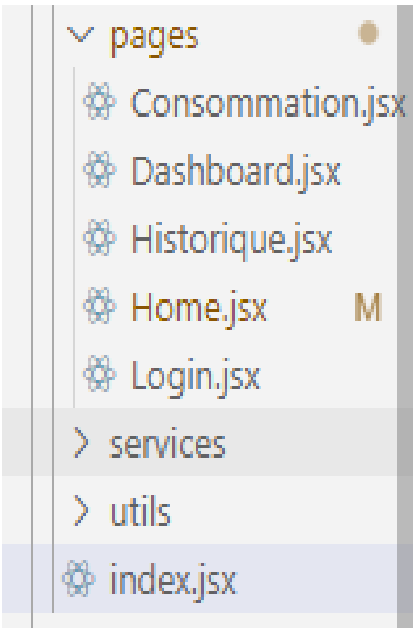


Figure III.16 : Organisation de notre projet React

- **Index.jsx**

Pour effectuer une navigation entre les pages, nous utilisons le fichier index.jsx, comme illustré dans la Figure III.17.

```
import React from "react";
import { createBrowserRouter, RouterProvider } from "react-router-dom";
import ReactDOM from "react-dom/client";
import Home from "./pages/Home";
import Login from "./pages/Login";
import { AuthProvider } from "./context/AuthContext";
import AuthGuard from "./guard/AuthGuard";
import Consommation from "./pages/Consommation";
import Dashboard from "./pages/Dashboard";
import Historique from "./pages/Historique";
```

Figure III.17 : Bibliothèques utilisés sur le programme React

Figure III.18 montre le bouton de navigation « consommation » dans l'interface de l'application web.

```
{
  path: "consommation",
  element: (
    <AuthGuard>
      <Consommation />
    </AuthGuard>
  ),
},
```

Figure III.18 : Chemin de navigation vers la page consommation

La Figure III.19 montre le chemin de navigation vers la page consommation.jsx à partir de ce bouton.



Figure III.19 : Bouton consommation

- **Login.jsx**

Figure III.20 illustre les bibliothèques utilisées pour la page login

```
import React from "react";
import { styled } from "@mui/material/styles";
import { Box, Stack, Container, Typography } from "@mui/material";
import LoginForm from "../components/LoginForm";
```

Figure III.20 : Bibliothèques de login

On importe la bibliographie de login (Figure III.20).

Après cela, nous avons sélectionné le type d'écriture, la taille et le fond d'écran, comme indiqué dans la Figure III.21.

```
const ContentStyle = styled("div")(({ theme }) => ({
  maxWidth: "480",
  margin:"auto",
  display: "flex",
  minHeight: "100vh",
  flexDirection: "column",
  justifyContent: "center",
  padding: theme.spacing(12, 0),
  backgroundImage:"url(/logo-univ.jpg)",backgroundPosition:"top-right",
  backgroundRepeat:"no-repeat",
}));
```

Figure III.21 : le choix de type d'écriture

Figure III.22 montre la programmation de la page Login.

```
const Login = () => {
  return (
    <RootStyle>

      <Container maxWidth="sm" >
        <ContentStyle>
          <Stack direction="row" alignItems="center" sx={{ mb: 4 }} >
            <Box sx={{ flexGrow: 1 }}>
              <Typography variant="h3" sx={{ color: "text.secondary", my: 2 }}>
                Bonjour,
              </Typography>
              <Typography sx={{ color: "text.secondary" }}>
                Entrez vos identifiants ci-dessous.
              </Typography>
            </Box>
          </Stack>
          <LoginForm />
        </ContentStyle>
      </Container>
    </RootStyle>
```

Figure III.22 : Programmation de la page Login

Figure III.23 montre l'interface de la page Login.



Figure III.23 : Interface de la page Login

Chaque utilisateur a la capacité de voir uniquement sa consommation et son historique grâce à la fonction d'authentification au niveau du Cloud, comme le montre la Figure III.24.

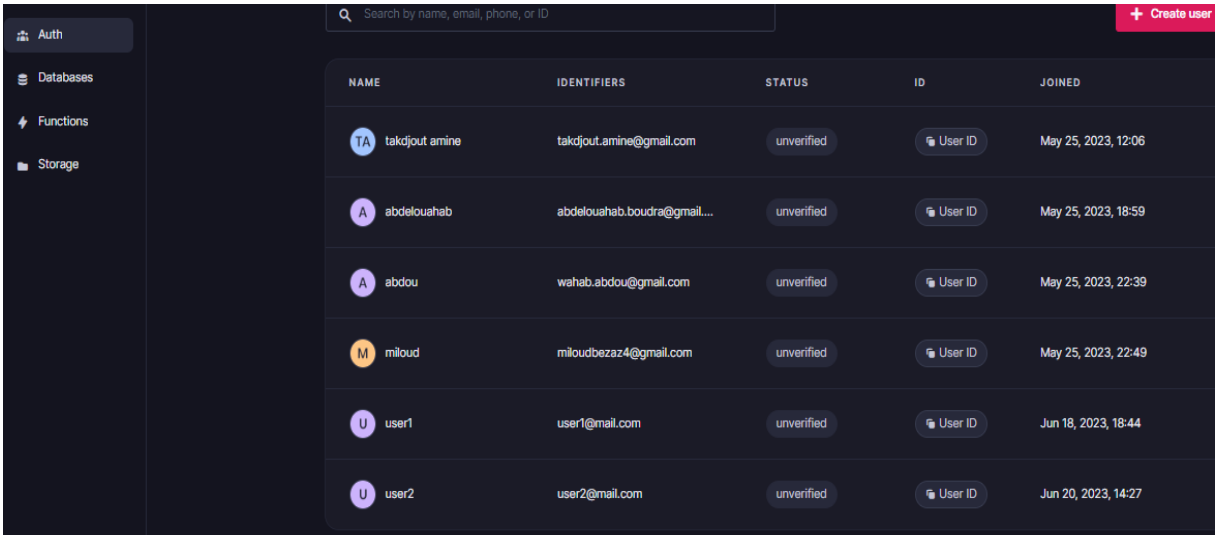


Figure III.24 : authentification d'utilisateur

- **Home.jsx**

Figure III.25 montre la programmation de la page Home.

```
<Stack spacing={3}>
  <Typography color="black" text-shadow="4px 4px 4px" align="center" variant="h3"
    {" "}
    UNIVERSITE YAHIA FARES DE MEDEA{" "}
  </Typography>
  <Typography color="black" align="center" variant="h4">
    {" "}
    FACULTE DE TECHNOlOGIE{" "}
  </Typography>
  <Typography color="black" align="center" variant="h4">
    {" "}
    Département du Génie Electrique{" "}
  </Typography>
  <Typography color="black" align="center" variant="h4">
    {" "}
    Projet fin d'étude-Master2 {" "}
  </Typography>
  <Typography color="black" align="center" variant="h4">
    {" "}
    Compteur-intelligent{" "}
  </Typography>
```

Figure III.25 : Programmation de la page Home

Figure III.26 montre l'interface d'accueil.



Figure III.26 : Interface de l'accueil

- **Consommation.jsx**

On importe la partie de la page consommation (Figure III.27).

```
import React, { useEffect, useState } from "react";
import Button from "@mui/material/Button";
import { Stack, Typography } from "@mui/material";
import { useNavigate } from "react-router-dom";
import useAuth from "../hooks/useAuth";
import { client } from "../services/appwrite";
```

Figure III.27 : Bibliographie pour la page consommation

Figure III.28 présente la programmation de l'affichage de la valeur de consommation.

```
    <Typography color="white" align="center" variant="h3">
      {data?.value}
    </Typography>
    <Typography color="white" align="center" variant="h3">
      KW/h
    </Typography>
  </Stack>
  <Typography color="white" align="center" variant="body1">
    Dernière mise a jour :{" "}
    {data?.date ? new Date(data?.date).toLocaleString() : "----"}
  </Typography>
</Stack>
<Stack alignItems="center">
  <Typography color="success" align="center">
    <Button
      onClick={() => navigate("/")}
      sx={{ width: "30%", margin: 2 }}
      variant="contained"
    >
      Home
    </Button>
    <Button
      onClick={() => logout()}
      sx={{ width: "15%", margin: 2, color: "success", size: "large" }}
      variant="outlined"
    >
      logout
    </Button>
  </Typography>
```

Figure III.28 : Programmation de l'affichage de la valeur de consommation

Figure III.29 présente l'affichage de la valeur de consommation d'énergie.

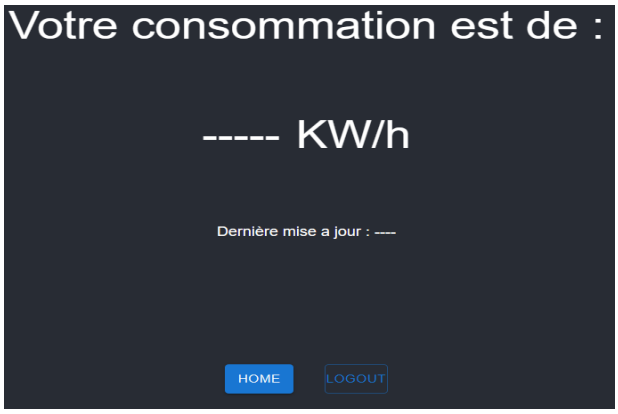


Figure III.29 : Affichage de la valeur de consommation

La Figure III.30 présente le processus d'inscription au niveau du Cloud pour récupérer les données en temps réel en se connectant à Appwrite à l'aide de l'ID de la base de données, de l'ID de la collection et de l'ID du document.

```
useEffect(() => {
  return client.subscribe(
    "databases.646a224b9c7d1e4608a1.collections.648f45a9357a3663f0e4.documents",
    (response) => {
      if (
        response.events.includes(
          "databases.*.collections.648f45a9357a3663f0e4.documents.*.create"
        )
      ) {
        if (user.$id === response.payload?.uid) {
          setData(response.payload);
          console.log(response.payload, user);
        }
      }
    }
```

Figure III.30 : Inscription au niveau de Cloud pour récupère la données

- **Historique.jsx**

Figure III.31 présente l'importation des bibliothèques de la page historique.

```
import React, { useCallback, useEffect, useState } from "react";
import Button from "@mui/material/Button";
import { Box, Stack, Typography } from "@mui/material";
import { useNavigate } from "react-router-dom";
import useAuth from "../hooks/useAuth";
import { DataGrid } from "@mui/x-data-grid";
import { Databases } from "appwrite";
import { client } from "../services/appwrite";
import Loading from "../components/Loading";
```

Figure III.31 : Importation des bibliothèques de la page historique

La Figure III.32 présente la récupération en temps réel des données de la base de données à partir du cloud en utilisant l'ID de la base de données et l'ID de la collection.

```
const databases = new Databases(client);

const promise = databases.listDocuments(
  "646a224b9c7d1e4608a1",
  "648f45a9357a3663f0e4"
```

Figure III.32 : Récuperation database realtime

Figure III.33 présente l'affichage de l'historique.

Historique

ID	Value	Date
1	1234	20/06/2023 14:07:09
2	1234	20/06/2023 14:07:29
3	1234	20/06/2023 14:15:15
4	1234	20/06/2023 14:15:35

Figure III.33 : Affichage de l'historique

Chaque fois que nous souhaitons accéder à notre application, nous avons utilisé la plateforme de déploiement et d'hébergement d'applications web appelée Vercel. Vercel est spécialement conçu pour faciliter le déploiement de projets React, Next.js et d'autres applications front-end. Voici les étapes que nous avons suivies :

1. Nous avons poussé notre code vers un dépôt Git tel que GitHub, GitLab ou BitBucket.
2. Ensuite, nous avons importé notre projet React dans Vercel. La plateforme a automatiquement configuré les paramètres appropriés pour notre déploiement, comme le montre la Figure III.33.

3. Grâce à Vercel, nous avons obtenu un lien d'accès direct vers notre application. Cela a éliminé la nécessité d'exécuter le programme localement à chaque fois que nous voulions y accéder.
4. Maintenant, notre application est déployée à l'adresse suivante : 'https://web-app-cmpt-int.vercel.app/'. Nous pouvons y accéder depuis n'importe quel appareil en utilisant ce lien, ce qui la rend facilement accessible et utilisable.
5. La Figure III.34 montre la création du projet React sur Vercel. Une fois connectés au site, nous avons accès à la fenêtre de connexion.
6. Grâce à Vercel, nous avons pu déployer notre application de manière efficace, ce qui nous permet de la rendre accessible à tout moment et depuis n'importe où.

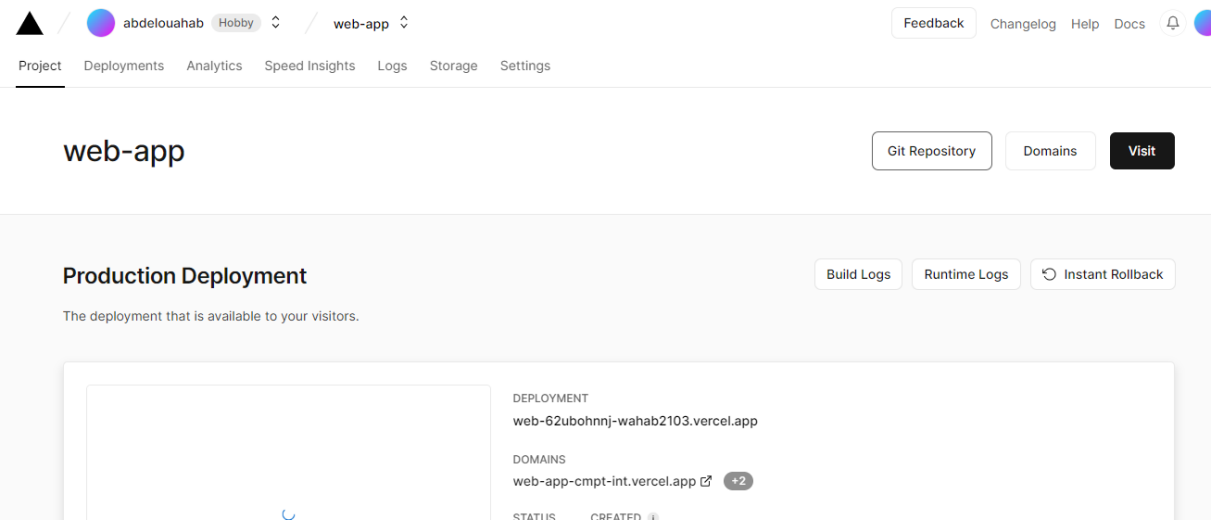


Figure III.34 : Plateforme web de déploiement l'application web

III.7 Résultats des tests

Figure III.35 illustre un prototype de notre solution proposée, où les différents dispositifs électroniques sont montrés, tels que : les cartes "Arduino Nano", les modules de communication (nRF24L01+) et le serveur (Raspberry Pi).Nous avons effectué les tests illustrés par les figures suivantes :

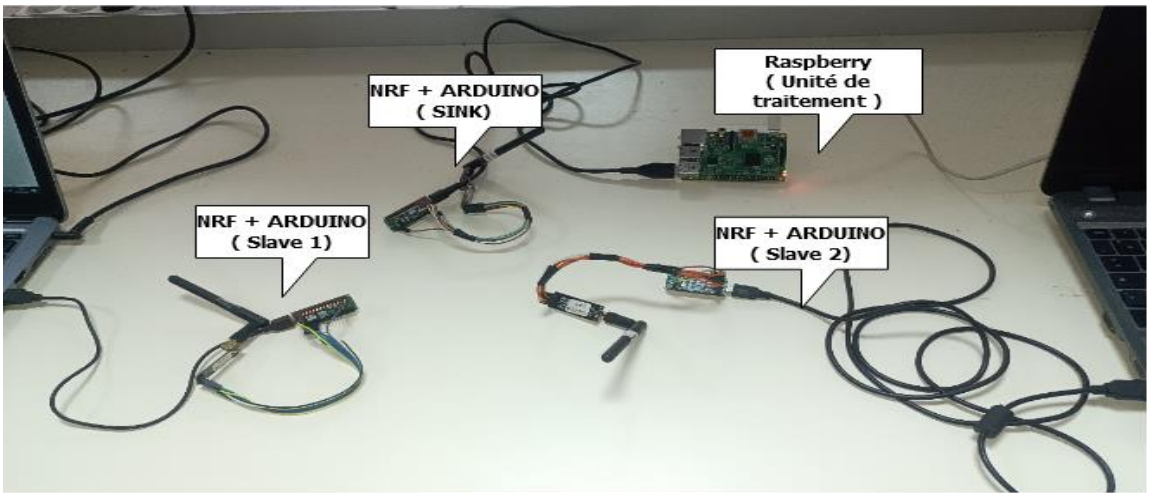


Figure III.35 : Réalisation du montage

- Slave 1 envoie une donnée de valeur fixe (170)au Sink (Figure III.36).

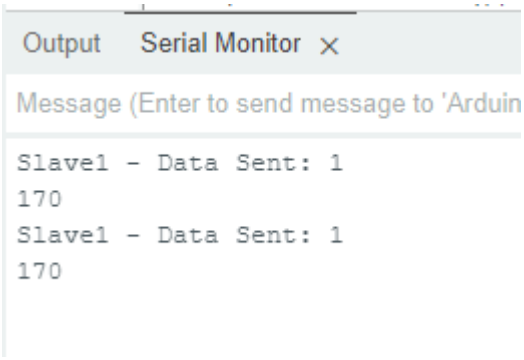


Figure III.36 : Donnée envoyé par NRF slave 1

- Slave2 envoie une donnée de valeur fixe (420) vers le Sink (Figure III.37).

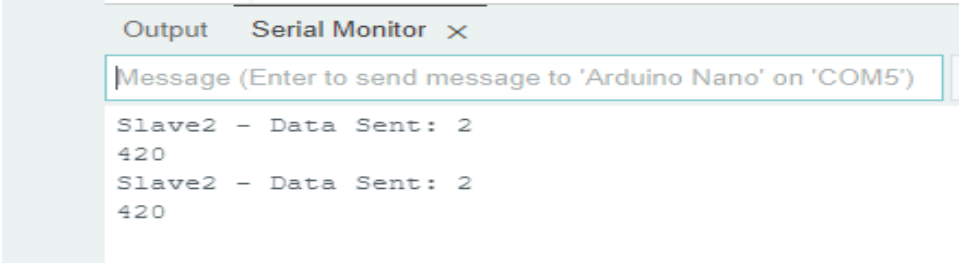


Figure III.37 : Donnée envoyé par NRF slave 2

- Les deux données ont été réceptionnées par l'Arduino collecteur "Sink" (Figure III.38).

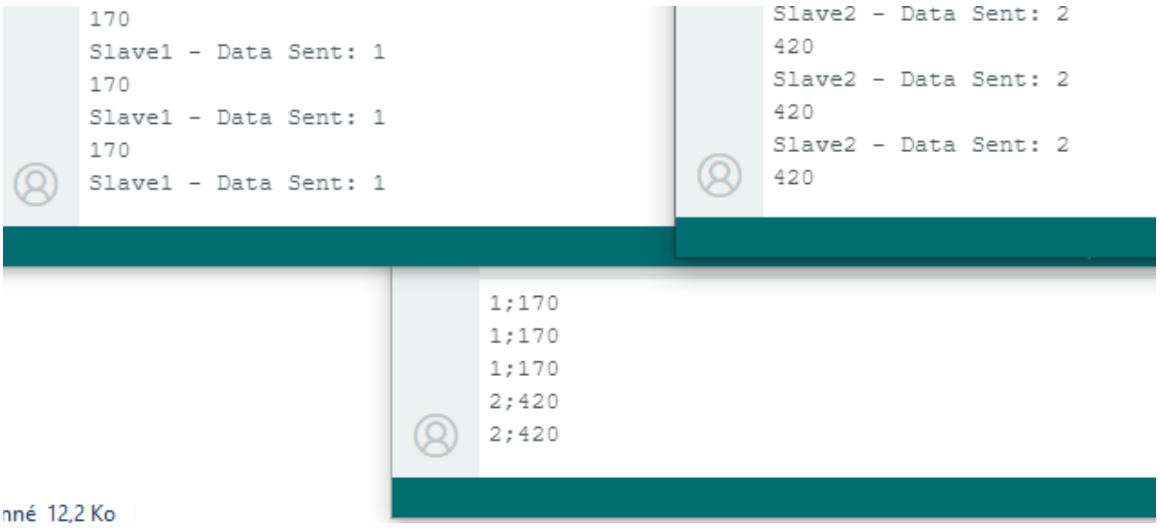


Figure III.38 : Réception des données par Arduino Sink

- Nous utilisons Node.js pour nous connecter à la base de données Appwrite (Figure III.39).

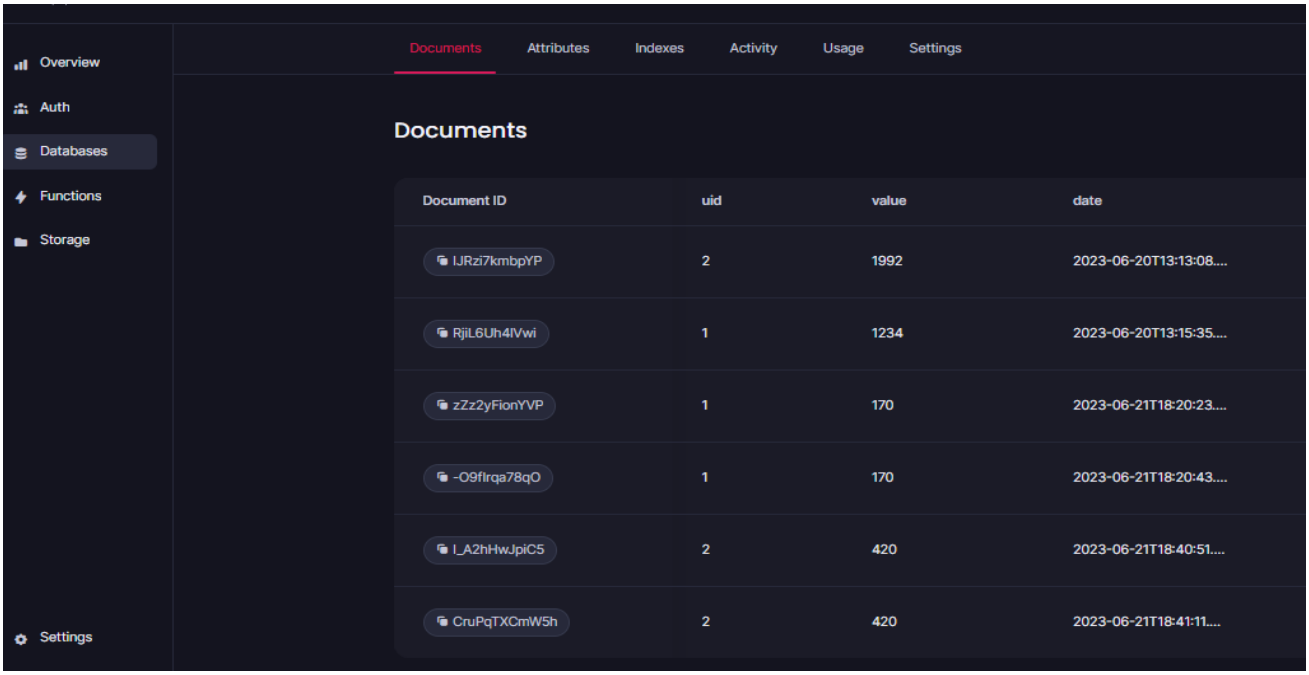


Figure III.39 ; Connexion à la base de données

Application web

Nous devons récupérer deux valeurs à partir de la base de données, donc nous avons deux utilisateurs, ‘user1’ et ‘ user2’ (Figure III.40).

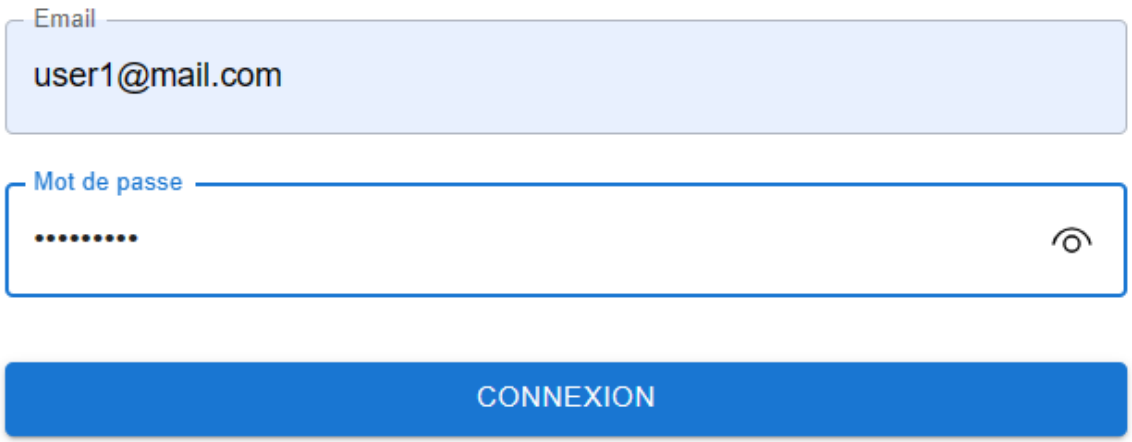


Figure III.40 : Login d'utilisateur 1

Figure III.41 présente la consommation de user1.



Figure III.41 : Consommation de user 1

Figure III.42 présente l'historique de user1.

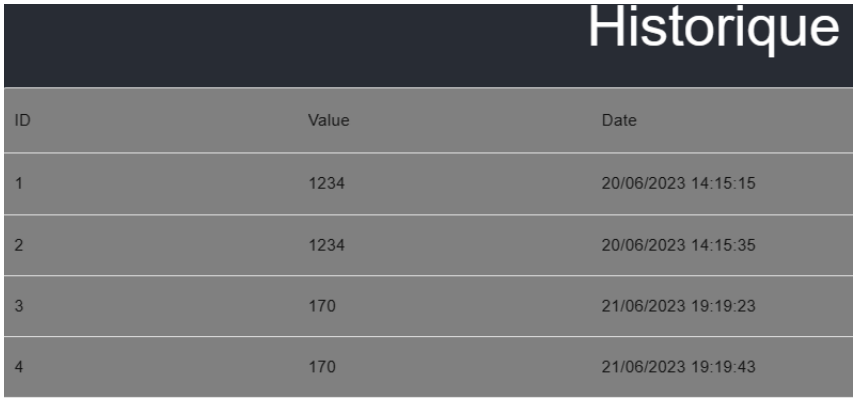


Figure III.42 : Affichage de l'historique de user1

La même chose pour l'utilisateur 2. Figure III.43 présente le login de user2.

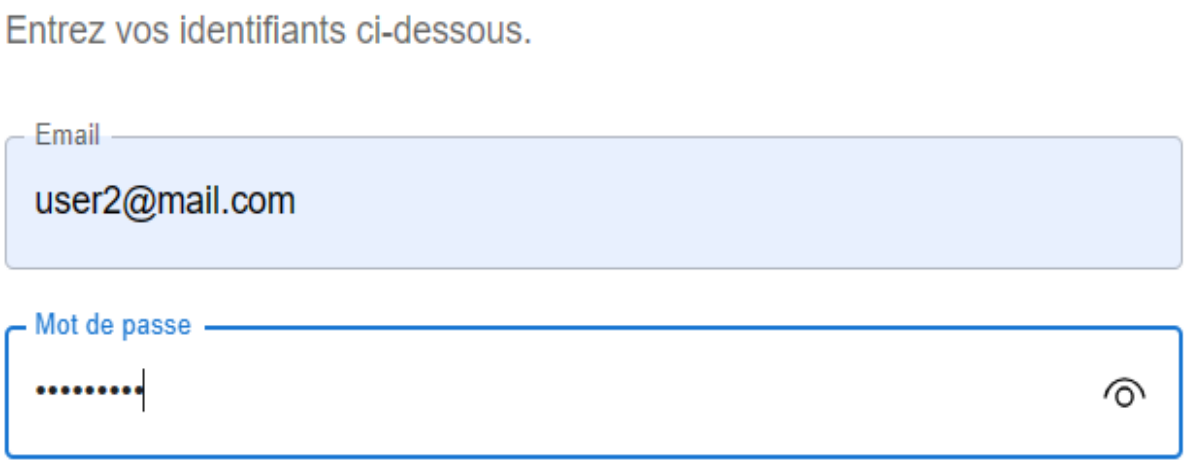


Figure III.43 : Login pour user2

Figure III.44 présente la valeur de consommation de user2.



Figure III.44 : valeur de consommation pour user2

Figure III.45 présente l'historique de consommation de user2.

Historique

ID	Value	Date
1	1992	20/06/2023 14:08:08
2	1992	20/06/2023 14:13:08
3	420	21/06/2023 19:40:51
4	420	21/06/2023 19:41:11

Figure III.45 : Historique de consommation de user 2

III.8 Conclusion

Dans ce chapitre, nous avons abordé la conception et l'implémentation du système proposé. Nous avons commencé par une introduction, suivie d'une description détaillée du système. Ensuite, nous avons expliqué le principe de fonctionnement du système et présenté les solutions matérielles et logicielles adoptées.

Au niveau matériel, nous avons discuté des composants et des dispositifs utilisés pour la mise en place du système. Du côté logiciel, nous avons examiné les différentes applications et programmes nécessaires au bon fonctionnement du système.

Nous avons également présenté les algorithmes utilisés dans le système, en expliquant leur rôle et leur fonctionnement. Ces algorithmes sont essentiels pour traiter les données collectées et fournir les résultats souhaités.

En ce qui concerne l'implémentation du système, nous avons détaillé les étapes de collecte des données, de création de la base de données et de conception de l'interface homme-machine (IHM). Ces éléments sont cruciaux pour assurer le bon fonctionnement et l'interaction conviviale du système.

Enfin, nous avons discuté des résultats des tests obtenus après l'implémentation du système. Ces résultats ont permis de vérifier l'efficacité et la performance du système, en mettant en évidence ses avantages et ses éventuelles limites.

En conclusion, ce chapitre a couvert tous les aspects de la conception et de l'implémentation du système proposé. Il a fourni une vision globale du fonctionnement du système, en mettant en évidence les choix matériels et logiciels, ainsi que les étapes clés de sa réalisation. Les résultats des tests ont confirmé l'efficacité du système, ouvrant ainsi la voie à la suite de notre étude.

Conclusion générale

Conclusion générale

Notre projet visait à mettre en place un système de compteurs intelligents pour les compteurs électriques. L'objectif principal était de créer un système connecté et intelligent permettant le contrôle et la supervision de la consommation électrique de chaque utilisateur, tout en offrant une application web basée sur React pour faciliter la visualisation et le contrôle des données. Grâce à ce système, les utilisateurs pourraient surveiller et gérer leur consommation d'électricité depuis n'importe où via leurs smartphones.

Au cours de notre étude, nous avons réussi à collecter des données à partir de deux Arduino Nano vers un Arduino Nano maître, puis à les stocker dans une base de données. Parallèlement, nous avons développé une application web réactive et interactive basée sur React, permettant aux utilisateurs de visualiser en temps réel leurs données de consommation ainsi que l'historique des données enregistrées.

Nous avons pris beaucoup de plaisir à travailler sur ce projet, car il représente l'avenir du secteur de l'énergie en Algérie. Notre système offre une solution qui peut contribuer à améliorer la gestion de l'énergie en facilitant l'accès aux informations nécessaires grâce à notre application développée. Ce projet nous a également permis de nous familiariser avec différents domaines tels que l'automatique, l'électronique et l'informatique, ce qui nous a permis d'acquérir des connaissances précieuses tout au long du processus.

L'un des principaux défis auxquels nous avons été confrontés dans ce projet était la programmation de l'application web en JavaScript. Le développement d'une application basée sur React et JavaScript a présenté des défis techniques importants. JavaScript est un langage puissant mais complexe, qui exige une compréhension approfondie de ses concepts et fonctionnalités pour créer une application web efficace et réactive. Nous avons dû relever des défis tels que la collecte des données, la gestion de l'état de l'application et la manipulation des données en temps réel. Cela nécessitait une solide maîtrise des concepts de JavaScript ainsi que des bibliothèques complémentaires telles que React.

Malgré tous nos efforts pour mener à bien ce projet, nous reconnaissons que notre contribution n'est qu'un point de départ sur un long chemin à parcourir. Il reste encore de

nombreuses possibilités d'amélioration, de complétion et de développement ultérieur de ce travail, notamment approfondir nos connaissances en JavaScript pour développer davantage l'application web, créer une interface pour Sonelgaz et connecter notre système au compteur homologue de Sonelgaz.

En résumé, notre projet a abouti à la création d'un système de compteurs intelligents avec une application web associée. Nous sommes conscients des perspectives d'amélioration et de développement futur, et nous espérons que notre travail servira de base solide pour de futures avancées dans le domaine des compteurs intelligents en Algérie.

Références bibliographiques

[1] A. Djebiri and H. Bakhaled, “Etude et réalisation d’un compteur d’énergie intelligent,” Mémoire Master, UNIVERSITE KASDI MERBAH OUARGLA, 2019.

[2] S. Abderrahim, “ETUDE ET REALISATION D’UN COMPTEUR ELECRIQUE,” Mémoire Master, UNIVERSITE YAHIA FARES DE MEDEA, 2021.

[3] « Définition: Compteur intelligent - smart meter: Futura Planète »,” [En ligne]. Disponible sur : https://www.futurasciences.com/planete/definitions/developpement-durable-compteur-intelligent-6952/ [Consulté le: 20/05/2023].

[4] Z. Rayane, “ETUDE ET REALISATION D’UN SMART METER BI-DIRECTIONNEL POUR APPLICATION SOLAIRE PHOTOVOLTAÏQUE,” Mémoire Master, Université Saad Dahlab Blida-, 2021.

[5] « Nrf24 séries, » Nordic Semiconductor, [En ligne]. Disponible sur : https://www.nordicsemi.com/products/nrf24-series [Consulté le: 23/05/2023].

[6] https://www.arduino.cc/ [En ligne]. [Consulté le: 23/05/2023].

[7] Arduino. (n.d.). "Arduino Nano Pinout & Schematics". Arduino, [En ligne] . Disponible sur :https://www.arduino.cc/en/uploads/Main/ArduinoNano30Schematic.pdf [Consulté le: 24/05/2023].

[8] « Internet des objets : FUTURA TECH » . [En ligne]. Disponible sur :https://www.futura-sciences.com/tech/definitions/internet-internet-objets-15158/ [Consulté le: 26/05/2023].

[9] H. MOULAI MOSTEFA and M. ALLAOUI, “Conception et développement d’une base de données et d’une IHM pour la supervision d’un réseau de capteurs sans fils implanté dans une serre agricole,” Mémoire Master, UNIVERSITE YAHIA FARES DE MEDEA, 2020.

[10]https://code.visualstudio.com/docs[En ligne]. [Consulté le: 01/06/2023].

[11] https://appwrite.io/docs [En ligne]. [Consulté le: 01/06/2023].

[12] H. Benyezza, “Fusion de données multi-capteurs et prise de décision intelligente : Application à un système de surveillance et contrôle d’un environnement agricole,” Thèse de Doctorat, Université de Médéa, 2021.