

REPUBLIQUE DU CAMEROUN

*Paix – Travail – Patrie*

\*\*\*\*\*

UNIVERSITE DE YAOUNDE I  
ECOLE NORMALE SUPERIEUR  
D'ENSEIGNEMENT TECHNIQUE  
D'EBOLOWA  
DEPARTEMENT DE DE GENIE  
INFORMATIQUE

\*\*\*\*\*



REPUBLIC OF CAMEROUN

*Peace – Work – Fatherland*

\*\*\*\*\*

UNIVERSITY OF YAOUNDE I  
HIGHER TECHNICAL TEACHER  
TRAINING COLLEGE OF  
EBOLOWA  
DEPARTMENT OF OF  
COMPUTER ENGINEERING

\*\*\*\*\*

**Filière  
Informatique Industrielle**

**CONCEPTION ET REALISATION D'UN COMPTEUR  
D'EAU INTELLIGENT EN MODE PREPAYE**

Mémoire présenté et soutenu en vue de l'obtention du  
Diplôme de Professeur d'Enseignement Technique deuxième grade  
(DIPET II)

Par : **BEIDI ADAMA Alain Serge**  
Licence en Techniques d'Analyse et Contrôle de Qualité

Sous la direction de  
**Dr. OLLE OLLE Daniel**  
Chargé de Cours

**Année Académique : 2019 - 2020**



**DEDICACES**

*A la famille ADAMA*

## REMERCIEMENTS

Nos sincères remerciements s'adressent :

- A DIEU Tout Puissant, pour nous avoir accordé, santé, intelligence et force pendant notre formation ;
- A Madame le Directeur de l'ENSET d'Ebolowa **Pr. NDJAKOMO ESSIANE Salomé**, pour tous les efforts déployés pour nous assurer une formation de qualité ;
- A Monsieur le Chef du Département de Génie Informatique et Encadreur, **Dr OLLE OLLE Daniel**, pour son encadrement, ses conseils et ses remarques ;
- A **Monsieur MEDZO ABA Charles**, Membre de notre équipe d'encadrement, pour son assistance, et ses conseils ;
- A tout le Staff Administratif et Professoral de l'ENSET d'Ebolowa pour la formation reçue tout au long de notre séjour ;
- A nos camarades de promotion, en particulier ceux du département génie informatique filière informatique industrielle niveau 5 ;
- A mes chers parents **Monsieur & Madame ADAMA** qui n'ont cessé de m'accompagner, me conseiller et me guider depuis toujours ;
- A mes sœurs et frères, **Dr. Djoda ADAMA, Ing. Hawa ADAMA, Ing. Nouho ADAMA** et **Gouara ADAMA** pour tout leur soutien.
- A Monsieur **TOUOYEM DIMITRI** & Monsieur **TCHASSEM William** pour leurs aide et conseils précieux pour le choix du matériel et la compréhension du fonctionnement de ces derniers ;
- A Monsieur **TAKODJOU Michel** & Monsieur **Guy**, pour leur aide dans les branchements électriques et de plomberie de notre projet ;
- A la famille **MBRING Lassou** pour son accueil à notre arrivée à EBOLOWA ;
- A toutes les personnes qui ont œuvré de près ou de loin pour le bon déroulement de ce projet et dans tous les aspects de ma vie.

# SOMMAIRE

DEDICACES .....	i
REMERCIEMENTS .....	ii
SOMMAIRE .....	iii
LISTE DES FIGURES .....	v
LISTE DES TABLEAUX .....	vii
LISTE DES ABBREVIATIONS .....	viii
RESUME.....	ix
ABSTRACT .....	x
INTRODUCTION GENERALE.....	1
CHAPITRE I : REVUE DE LA LITTERATURE.....	3
1. Le compteur intelligent .....	3
2. Le compteur intelligent en mode prépayé .....	4
2.1. Compteur Electrique Prépayé .....	4
2.2. Compteur D'eau Prépayé .....	5
3. Les moyens de paiement .....	6
3.1. Compteurs Intelligents Prépayés Avec Technologie GSM.....	6
3.2. Compteurs Intelligents Prépayés Basés Sur Carte A Puce .....	7
3.3. Compteurs Intelligents Prépayés Basé Sur La Radiofréquence (RFID) .....	7
4. Choix de la technologie de paiement .....	8
CHAPITRE II : CONCEPTION DU SYSTEME .....	9
1. Ecosystème du projet .....	9
2. Les capacités du compteur. ....	9
3. Organigramme de fonctionnement du compteur.....	12
4. Le matériels et les logiciels .....	15
4.1. Le matériel .....	15

iii

4.2. La partie Logicielle .....	50
CHAPITRE III : IMPLEMENTATION .....	53
1. Le compteur.....	53
1.1. Le compteur en action.....	53
1.2. Les captures d'écran SMS .....	55
2. Les codes de fonctionnement .....	56
2.1. Conversion de la somme d'argent de la carte RFID en volume d'eau .....	56
2.3. Réception par SMS du volume d'eau restant dans le compte du client.....	58
2.4. Affichage de l'état de consommations d'eau sur l'écran du compteur.....	59
2.5. Fermeture automatique de l'électrovanne à la fin de la consommation du volume d'eau acheté.....	60
2.6. Envoi de l'état de consommation sur le serveur à une heure précise .....	60
3. Le câblage .....	61
CHAPITRE IV : SIMULATION ET RESULTATS.....	63
CONCLUSION GENERALE .....	68
BIBLIOGRAPHIE .....	70
WEBOGRAPHIE.....	71
ANNEXES .....	I

## LISTE DES FIGURES

Figure I-1 : Synoptique Compteur Intelligent d'Electricité.....	4
Figure I-2 : Synoptique Compteur Intelligent d'eau.....	5
Figure I-3: Exemple de diagramme de fonctionnement de paiement par GSM pour un compteur électrique résumant le fonctionnement .....	6
Figure II-1 : Communication entre le compteur/le client/la compagnie des eaux .....	11
Figure II-2 : Organigramme du compteur au démarrage .....	13
Figure II-3 : Organigramme du compteur pendant son fonctionnement.....	14
Figure II-4 : Synoptique Global du compte .....	16
Figure II-5 : Représentation photographique de l'Arduino MEGA.....	18
Tableau II-1 : Fiche Technique Arduino MEGA .....	19
Figure II-6 : Vue de face SIM808 EVB 3.2 .....	24
Figure II-7 : Vue arrière SIM808 EVB 3.2 .....	25
Figure II-8 : Représentation du branchement de la SIM808 .....	27
Figure II-8 : Représentation du branchement de la SIM808.....	27
Figure II-9 : Broches essentiels au branchement de la SIM808.....	27
Figure II-10 : Schéma fonctionnement Commande AT.....	28
Tableau II-2 : Quelques Commandes AT et leurs actions sur le SIM808.....	31
Figure II-11 : Vue Extérieure du YF-S201 .....	32
Figure II-12 : Vue Intérieure du YF-S201.....	33
Figure II-13 : Principe de fonctionnement du YF-S201 [21].....	34
Figure II-14 : Branchement Standard du YF-S201 sur une carte Arduino .....	35
Figure II-13 : Module RFID RC522 avec Carte et Bracelet Mifare Classic 1k.....	36
Figure II-14 : Représentation 3D de la mémoire de la MIFARE 1K.....	38
Figure II-14 : Branchement de la RC522 avec une carte Arduino Uno .....	39
Figure II-15 : Afficheur LCD 16×2.....	40
Figure II-16 : Branchement LCD 16×2 avec I2C sur Arduino Méga .....	41
Figure II-17 : Modèle électrovanne 12V/DC .....	42
Figure II-18 : Fonctionnement d'une électrovanne .....	43
Figure II-19 : Branchement de l'électrovanne vers une carte Arduino .....	44
Figure II-20 : Relai 2 Modules.....	45
Figure II-21 : Coupe transversale interne d'un relai .....	45
Figure II-22 : Branchement relai 2 modules vers carte Arduino UNO .....	47
Figure II-22 : Relai - Contact Normalement Ouvert .....	47
Figure II-23 : Relai - Contact Normalement Fermé.....	48
Figure II-24 : RTC DS3231 de Keystudio .....	48
Figure II-25 : Branchement DS3231 vers carte .....	49
Figure II-26 : Interface du logiciel Arduino en version 1.8.12 .....	51
Figure II-27 : Interface Contrôle des données sur ThingSpeak .....	52
Figure III-1 : Phase configuration Compteur .....	54
Figure III-2 : Compteur lorsque le compte est vide en volume d'eau.....	54

Figure III-2 : Compteur lorsque le compte est vide en volume d'eau.....	54
Figure III-3 : Mauvaise Carte Utilisée .....	54
Figure III-2 : Compteur lorsque le compte est vide en volume d'eau.....	54
Figure III-3 : Mauvaise Carte Utilisée .....	54
Figure III-4 : Bonne Carte Utilisée.....	54
Figure III-5 : Le compteur en fonctionnement normal .....	55
Figure III-6 : Captures d'écran des messages reçus par le client .....	56
Figure II-: Extrait du code Relatif à la conversion de la somme d'argent de la carte RFID en Volume d'eau.....	57
Figure III-7 : Extrait du code Relatif à la conversion de la somme d'argent de la carte RFID en Volume d'eau.....	57
Figure II- : Extrait du code Relatif au Calcul du volume d'eau restant après consommation..	57
Figure III-8 : Extrait du code Relatif au Calcul du volume d'eau restant après consommation .....	57
Figure II-: Extrait du code relatif à la réception d'un SMS contenant le volume d'eau restant	59
Figure III-9 : Extrait du code relatif à la réception d'un SMS contenant le volume d'eau restant .....	59
Figure III-10 : Extrait du code relatif à l'affichage sur l'écran LCD des états de consommation .....	59
Figure III-11 : Extrait du code portant sur la fermeture automatique de l'électrovanne à la fin de la consommation du volume d'eau acheté.....	60
Figure III-13 : Câblage Interne.....	62
Figure III-14 : Schéma de câblage .....	62
Figure IV-1 : Synoptique compteur avec un capteur d'eau en amont du système.....	65
Figure IV-2 : Capteur de niveau d'eau ST045.....	65
Figure IV-3 : Synoptique idéal du compteur actuel .....	67

## LISTE DES TABLEAUX

Tableau II-1 : Fiche Technique Arduino MEGA .....	19
Tableau II-2 : Quelques Commandes AT et leurs actions sur le SIM808.....	31
Tableau II-2 : Quelques Commandes AT et leurs actions sur le SIM808.....	31
Tableau II-3 : Branchements des broches MOSI, MISO, SCK, SDA selon les cartes Arduino .....	40
Tableau II-4 : Prix du matériel .....	50
Tableau IV – 1 : Volume d’air mesuré par souffle humain .....	64
Tableau IV – 2 : Volume d’eau mesuré .....	66



## **LISTE DES ABBREVIATIONS**

<b>CPL</b>	Courant Porteur Logique
<b>LCD</b>	Liquid Crystal Display
<b>RFID</b>	Radio Frequency Identification
<b>CPU</b>	Central Processing Unit
<b>ROM</b>	Read Only Memory
<b>EEPROM</b>	Electrically Erasable Programmable Read Only Memory
<b>GSM</b>	Global System for Mobile Communications
<b>SMS</b>	Short Message System
<b>GPRS</b>	General Packet Radio Service
<b>GPS</b>	Global Positioning System
<b>RTC</b>	Real-Time Clock
<b>SPI</b>	Serial Peripheral Interface
<b>I2C</b>	Inter Integrated Circuit
<b>TTL</b>	Transistor-Transistor Logic
<b>M2M</b>	Machine-to-Machine
<b>PMW</b>	Pulse Width Modulation
<b>TCP</b>	Transmission Control Protocol
<b>HTTP</b>	Hypertext Transfer Protocol
<b>AT</b>	Attention
<b>IDE</b>	Integrated Development Environment
<b>IOT</b>	Internet Of Things

## **RESUME**

Au Cameroun, le fonctionnement des compteurs d'eau et par ricochet la remise des factures qui en découle est souvent sujet à des revendications de la part des clients qui n'acceptent pas les consommations qui leurs sont attribuées à la fin du mois, ceci pouvant causer des piratages des compteurs et des dégradations des installations d'eau. Alors vu cette situation et au vu de l'évolution technologique et de son accessibilité qui nous est plus aisé, nous avons été emmenés de nous demander comment améliorer les compteurs d'eau, les rendre plus intuitifs et automatiques par rapport aux compteurs d'eau actuel ? Ainsi donc, pour répondre à cette problématique, très souvent source d'incompréhension et de conflit entre les clients et la société en charge de la fourniture d'eau dans les ménages (CAMWATER), nous nous proposons de concevoir et de réaliser un compteur d'eau intelligent en mode prépayé. Celui-ci fonctionnera avec la technologie des systèmes embarquées à savoir Arduino dans notre cas, pour le prépaiement utiliser la technologie RFID et la technologie GSM pour la communication. Ce compteur devra permettre au client de payer juste la quantité d'eau dont il aura besoin, de contrôler son état de consommation à tout moment et de pouvoir fermer l'arrivée de son compteur d'eau à distance.

Mots Clés : RFID, Compteurs intelligents, Eau

## **ABSTRACT**

In Cameroon, the operation of water meters and by extension, the delivery of invoices which results therefrom is often subject to claims from customers who do not accept the consumption which is allocated at the end of the month. This brings along pirated meters and degraded water installations. So, given this situation, given also the technological evolution and its accessibility which is easier for us, we were led to ask ourselves how we could improve the water meters, make them more intuitive and automatic compared to current water meters? As a answer to this problem, very often source of misunderstanding and conflict between customers and the company in charge of the supply of water in households (CAMWATER), we propose to design and produce a smart water meter in prepaid mode. This will work with embedded systems technology, named Arduino in our case, for prepayment use RFID technology and GSM technology for communication. This meter will allow the customer to pay just the amount of water he will need, to control his consumption status at any time and to be able to close the arrival of his water meter remotely.

Keywords: RFID, Smart meters, Water

## **INTRODUCTION GENERALE**

Dans un contexte de vie difficile que nous vivons dans notre pays, les compteurs d'électricité ou d'eau sont sujet à des discussions, surtout lors de la remise des factures de consommation, moment qui est une phase de souci pour de nombreux ménages. Dans le cas particulier de l'eau ces ménages se disent souvent surpris des états de consommation d'eau qui leurs sont fournis à la fin du mois, rejetant le tort soit sur les agents releveurs d'index ou bien directement au fournisseur d'eau qui gonflerait les consommations d'eau. Et ceci pouvant entraîner un non-paiement des factures d'eau, un piratage des compteurs pour diminuer les valeurs mesurées ou pire sectionner les arrivées d'eau en amont des domiciles. Et cette situation pour le fournisseur peut entraîner un manque à gagner financier et logistique et en général une mauvaise image auprès de l'opinion public.

Au regard de ce constat qui crée très souvent une défiance entre les clients et la compagnie de gestion des eaux (CAMWATER), et compte tenu des moyens techniques appris pendant notre formation ou lors des recherches effectuées sur la toile et vu la facilité d'accès à la technologie avec l'acquisition facile et peu onéreuse de composants électroniques, il nous est alors apparu de se demander avec tous ces éléments à notre disposition, comment améliorer les compteurs d'eau, les rendre plus intuitifs et automatiques par rapport aux compteurs d'eau actuel ??

Afin de répondre à cette problématique, nous aurons comme objectif la conception et la réalisation d'un compteur d'eau intelligent en mode prépayé.

Pour sa fonction principale à savoir le prépaiement, il s'avère que plusieurs technologies s'offraient à nous à savoir le GSM à travers l'utilisation du SMS, l'utilisation de cartes à puce ou encore magnétiques et enfin l'utilisation de carte utilisant la technologie RFID (Radio Frequency Identification ou Radio-Identification en français), technologie qui est aussi utilisée par sa dérivée à savoir le NFC pour le paiement sans contact. Aussi pour faire tourner dans le bon sens et commander l'ensemble des composants qui vont constituer notre compteur nous allons utiliser une technologie des systèmes embarqués à travers l'un des plus connus à savoir Arduino qui est un système parfait pour créer d'innombrables circuits électroniques, les commander et faire du prototypage rapide. Du fait de son côté open-source et de l'obtention un peu aisé des composants compatibles avec sa programmation, il est parfaitement adapté pour les projets comme le nôtre.

Le compteur d'eau intelligent en mode prépayé, que nous allons conceptualiser et réaliser devra entre autres :

- Envoyer de manière quotidienne l'état de consommation sur le volume d'eau acheté au client via SMS et par ce même canal le client pourra connaître cette même information s'il le souhaite.
- Le compteur se bloquera de façon automatique à la fin du volume d'eau acheté et le client pourra bloquer son compteur à distance s'il le souhaite en cas de problème de fuite par exemple et s'il n'est pas présent sur les lieux.
- Envoyer les données de consommation et d'activation de compte en ligne pour permettre à la compagnie de connaître l'état de consommation des eaux dans une zone s'il le souhaite et aussi au client de voir ses états en ligne s'il le souhaite également.
- Permettre sa géolocalisation par les techniciens de la compagnie des eaux pour une éventuelle maintenance au cas où lesdits techniciens n'ont aucune information sur son lieu d'installation et surtout le retrouver s'il a été déplacé par exemple.

Le présent rapport qui synthétise nos travaux s'articule autour de quatre chapitres à savoir :

Chapitre 1 : Revue de la Littérature, qui va traiter des divers travaux effectués sur les compteurs intelligents et ceux en mode prépayé.

Chapitre 2 : Conception du Système qui traitera du fonctionnement de notre solution et des outils matériels et logiciels qui seront utilisés pour la phase de réalisation

Chapitre 3 : Implémentation, qui va s'atteler à présenter le produit fini de notre projet et la présentation d'extraits de codes qui vont expliquer sur le plan informatique certaines fonctionnalités du compteur.

Chapitre 4 : Simulation et Résultats, qui servira à tester la maquette en condition réelle afin de voir son comportement et d'en tirer certaines conclusions sur notre compteur.

Et enfin une Conclusion Générale.

# **CHAPITRE I : REVUE DE LA LITTERATURE**

## **1. Le compteur intelligent**

En se basant sur l'article « compteur communicant » publié par WILD Jean en 2018 dans son article intitulé « Compteur Communicant » [1], le compteur intelligent encore appelé compteur communicant est défini comme étant un compteur qui mesure de manière détaillée et précise, et éventuellement en temps réel, une consommation d'électricité, d'eau ou de gaz. La transmission des données s'effectue généralement par courants porteurs en ligne (CPL) au gestionnaire du réseau de distribution chargé du comptage.

Les compteurs intelligents servent à produire des factures sur la consommation réelle, à repérer des gaspillages, des pertes en ligne ou les postes qui coûtent le plus au client. Le compteur peut en outre être bidirectionnel et être programmable à distance. Ces fonctions sont à la base de l'évolution des réseaux (d'électricité, d'eau ou de gaz) vers les « réseaux intelligents ».

Mais malgré l'évolution des compteurs dans un modèle intelligent, nous avons de Grégoire Wallenborn de l'Université Libre de Bruxelles qui dans un rapport [2], à l'attention des parlementaires de la Wallonie en Belgique, présente les aspects négatifs de ces nouveaux types de compteur en ressortant une liste de griefs contre les compteurs communicants à savoir :

- Les compteurs communicants émettent des ondes électromagnétiques
- Les compteurs communicants n'aident pas les consommateurs à réduire leur facture (ou leur consommation).
- Le souci Protection de la vie privée et big data
- L'utilisation de nouvelles ressources par les compteurs communicants.
- Les compteurs communicants ne sont pas indispensables pour la transition énergétique.

Au vu de ces études, on pourrait, compte tenu de la liste des inconvénients ci-dessus relevés, décider de ne plus avancer dans nos recherches, mais nous signalons le fait que le monde actuel est de plus en plus connecté et donc dans le contexte camerounais, la réalisation d'un compteur communicant est vraiment important pour le client surtout que notre axe se porte sur la consommation de l'eau et non de l'électricité du fait que la compagnie nationale ENEO propose déjà à ses clients des compteurs de ce type et qui sont déjà acceptés par ceux-ci malgré les réticences enregistrées au début de cette opération.

Pour ce cas d'ENEO, nous constatons que les compteurs intelligents permettent bien la mesure des consommations d'énergie. Dans la majorité des cas ils servent plus à compter ou à relever la consommation de façon directe, c'est-à-dire à mesurer sans en fixer une limite. Or notre travail veut se concentrer sur l'aspect prépayé d'un compteur intelligent c'est-à-dire permettre au client de consommer la quantité d'eau exacte qu'il aura payé

## 2. Le compteur intelligent en mode prépayé

Pour ce type de compteur nous pouvons signaler qu'il en existe déjà au Cameroun, particulièrement dans la partie septentrionale (Adamaoua, Nord et Extrême-Nord), par l'entremise de la société Huawei qui construit des centrales solaires dans les zones rurales et a opté pour le prépayé par ses consommateurs.

Ainsi nous constatons que la majorité de travaux de recherche portant sur la conception des compteurs prépayés se concentrent sur le compteur électrique et avait au cœur la carte Arduino [10] pour gérer le fonctionnement desdits compteurs. Rares sont ceux qui traitent des compteurs prépayés dans le cas de l'eau.

Toutefois, en ayant le système embarqué comme base commune, le système de fonctionnement proposé est quasiment le même :

### 2.1. Compteur Electrique Prépayé

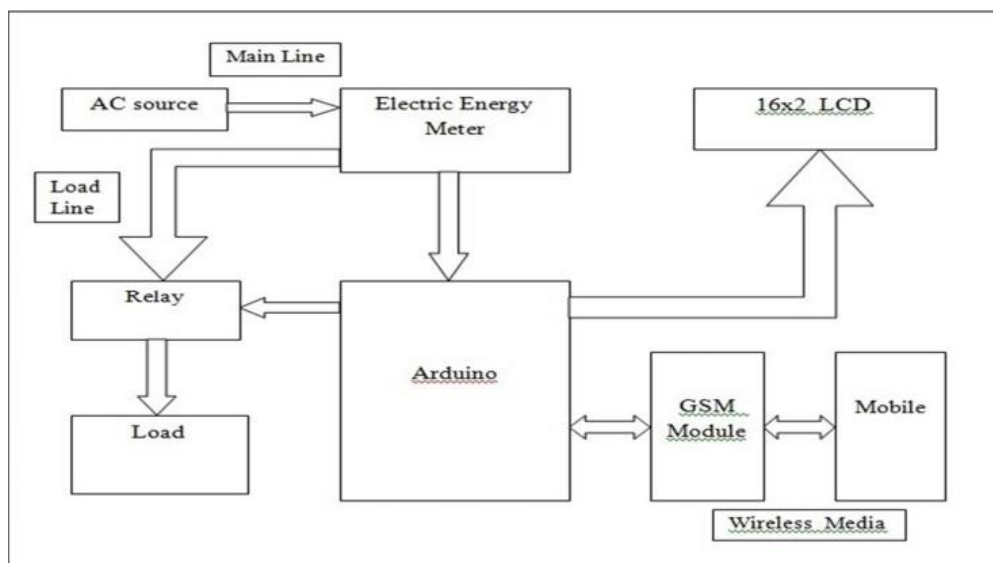


Figure I-1 : Synoptique Compteur Intelligent d'Electricité [3]

Le principe de conception est le suivant ; on relie la carte Arduino à :

- Un compteur qui sert à mesurer la consommation électrique et qui est relié à l'arrivée du courant.
- Un relai dont le rôle sera d'allumer et éteindre le compteur en fonction de la quantité de KWh acheté encore disponible.
- Un écran LCD pour afficher les informations.
- L'ensemble constitué du réseau GSM et d'un téléphone mobile pour la gestion du paiement.

## 2.2. Compteur D'eau Prépayé

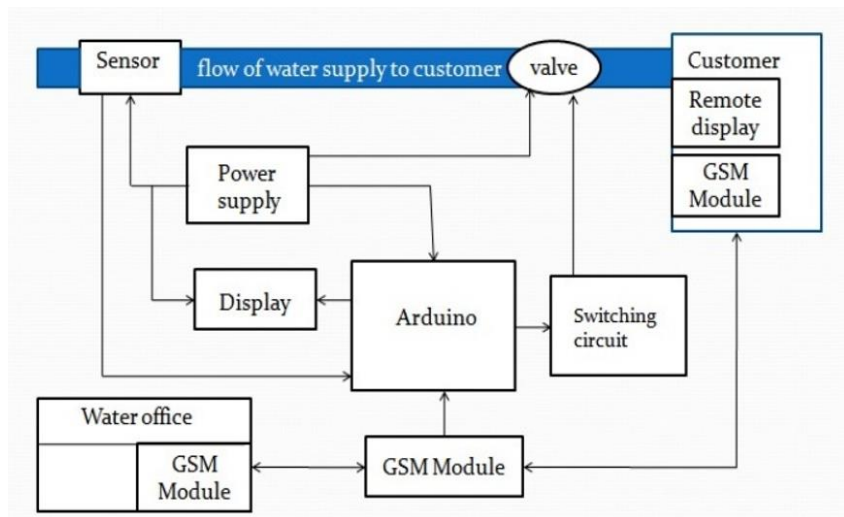


Figure I-2 : Synoptique Compteur Intelligent d'eau [4]

Le principe de conception est le suivant ; on relie la carte Arduino à :

- Un capteur qui aura le rôle de mesurer le débit d'eau et la quantité d'eau consommée.
- Une électrovanne dont le rôle est de couper l'arrivée d'eau dans le domicile ou de laisser passer l'eau en fonction de la somme d'argent et dont le volume d'eau disponible dans le système. L'électrovanne est commandée par un relai relié à une carte Arduino.
- Un écran LCD pour afficher les informations.
- L'ensemble constitué du réseau GSM et d'un téléphone mobile pour la gestion du paiement.



Les méthodes de fonctionnement étant quasiment similaires, la partie du système de notre compteur qui pourra faire la différence par rapport aux autres études existantes serait la méthode de moyen de paiement.

### 3. Les moyens de paiement

Avec les études lues, nous pouvons sortir trois systèmes principaux pour le paiement :

#### 3.1. Compteurs Intelligents Prépayés Avec Technologie GSM

Dans ce type de système, un message est envoyé aux compteurs intelligents via le réseau GSM par les consommateurs après recharge leur compte de téléphone portable. Un certain montant est envoyé au compteur qui est converti en nombre d'unités (qui correspond au KWh ou au volume d'eau équivalent exprimé en M<sup>3</sup>) et ces unités sont stockées dans le système. Lors de la consommation d'eau ou d'électricité, les unités sont déduites par le compteur et lorsque les unités achetées sont épuisées, l'électricité ou l'eau est coupée. Lorsque la prochaine recharge est envoyée au compteur, il reconnaît le numéro de mobile, décode le message et rajoute un nombre d'unités correspondant dans sa mémoire de stockage afin que l'électricité ou l'eau soit restaurée [5] [6] [7] [8] [9].

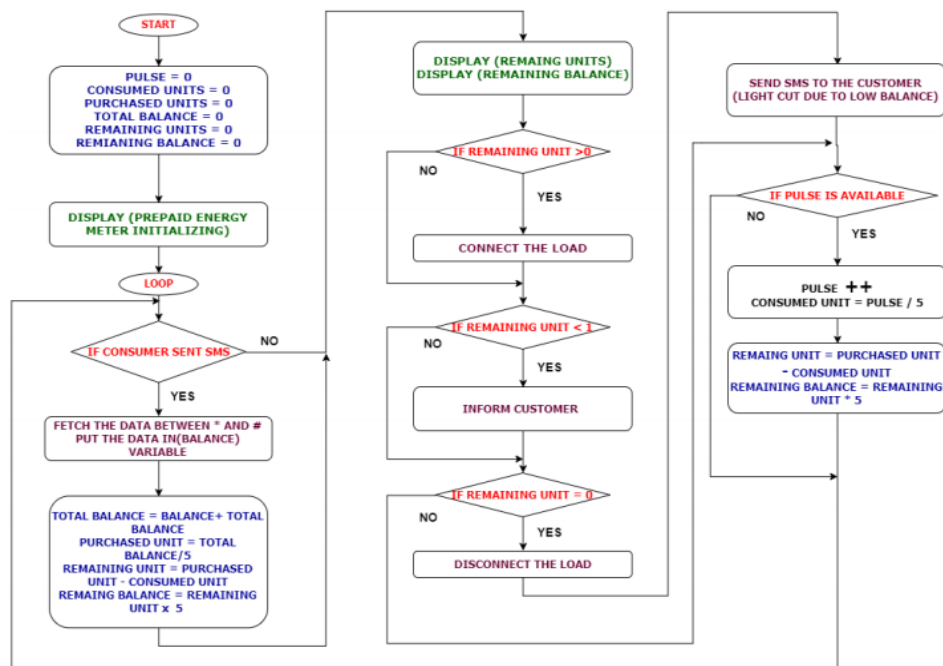


Figure I-3: Exemple de diagramme de fonctionnement de paiement par GSM pour un compteur électrique résumant le fonctionnement [5]

Pour ce système nous trouvons qu'il est vraiment pratique, compte tenu de la disponibilité du réseau mobile GSM sur une grande partie du territoire camerounais. Ce qui rend son exploitation plus aisée et facile par le client qui à partir d'un simple SMS peut activer son compteur.

Contrairement à notre cas, nous constatons que le système présenté dans les divers travaux met seulement en avant la méthode, mais pas la façon de l'appliquer en tenant compte de la sécurité et de la traçabilité auprès du fournisseur. Ainsi on ne propose pas une méthode qui permet au fournisseur d'énergie ou d'eau de savoir exactement quel client a pris une certaine quantité d'unité, et aussi le client, s'il est un peu « bidouilleur », peut trouver le moyen lui-même d'envoyer le code contenant les unités voulues.

### **3.2. Compteurs Intelligents Prépayés Basés Sur Carte A Puce**

Dans ce type de compteur prépayé, il existe deux principaux Composants ; l'un est une carte à puce et un autre est une carte à puce lecteur [5].

La carte à puce est comme une carte de crédit en plastique et se compose de différents composants tels que CPU, ROM, EEPROM, etc. donc le circuit fondamentalement est intégré sur une carte à puce. Il y a tout un système d'exploitation de carte à puce à travers lesquelles données sont contrôlées par une carte à puce. Dans ce genre de système, le consommateur doit recharger sa carte autant de nombre d'unités qu'il souhaite. Plus tard, cette carte est insérée dans le lecteur de carte qui est intégré avec le compteur composant un ensemble.

Ensuite, le lecteur de cartes fait son travail et stocke les unités qui sont disponibles dans la carte à puce, le compteur d'énergie réduit les unités en autant de kilowatts d'électricité consommée. Lorsque l'unité atteint à zéro, il déconnecte l'électricité jusqu'à la prochaine recharge.

Bien qu'étant sécurisé parce que comportant la même technologie que les distributeurs de billets de banque ou encore les modules de paiement en supermarché, ce système aurait pu être un bon axe d'étude, mais a un gros défaut et/ou inconvénient qui est sa rareté et donc le coût élevé de ce type de module sur les systèmes embarqués Arduino.

### **3.3. Compteurs Intelligents Prépayés Basé Sur La Radiofréquence (RFID)**

Le RFID [5] est une méthode pour mémoriser et récupérer des données à distance en utilisant des marqueurs appelés « radio-étiquettes ».

En suivant cette stratégie, les individus (clients) utilisent la carte RFID émises par les fournisseurs d'eau ou d'électricité. Les unités pourront être achetées en rechargeant les cartes RFID tout en utilisant le code dans la carte. Au moment où l'acheteur doit activer sa consommation d'électricité ou d'eau, il doit montrer sa carte au lecteur, à ce stade, le code unique à l'intérieur de la carte est perçu par le lecteur et commence à déduire de la carte RFID le montant qui équivaut à la quantité d'unité d'eau ou d'électricité achetée. Après consommation du montant total, le client doit recharger à nouveau sa carte RFID.

Cette méthode à l'avantage de la sécurité comme celle de la carte à puce et il est facile à obtenir.

La solution RFID est le premier pas vers la technologie NFC qui est une technologie de paiement sans contact, disponible dans de nombreux types de téléphones.

#### **4. Choix de la technologie de paiement**

Nous aurions pu travailler sur cette méthode en utilisant la technologie GSM et en mettant l'accent sur cet aspect sécurité, mais du fait qu'il aurait fallu par exemple trouver un accord avec un des opérateurs téléphoniques pour utiliser leur système de paiement par mobile, ce qui aurait été lourd et contraignant pour nous et du fait aussi que la méthode a été déjà présentée dans de nombreux articles. Ainsi nous avons préféré travailler sur d'autres méthodes de paiement.

Quant à la carte à puce, son non choix est surtout dû au fait de la rareté de ce module avec les cartes Arduino. La méthode par la technologie à l'avantage pour nous de n'avoir pas eu une application dans les études faites sur les compteurs prépayés et commandés par le système Arduino et aussi le matériel est facile à trouver et ne nécessite pas d'énorme moyen financier pour son acquisition.

## **CHAPITRE II : CONCEPTION DU SYSTEME**

Dans cette phase de conception nous allons nous atteler à présenter notre compteur dans son écosystème de fonctionnement, d'expliquer le fonctionnement de ce dernier. Ensuite présenter les outils logiciels et matériels qui vont concourir à la matérialisation de ce compteur.

### **1. Ecosystème du projet**

Le compteur de notre projet, étant intelligent fonctionne dans un écosystème, comprenant le compteur lui-même, le client et le gestionnaire d'eau. Et l'ensemble concoure au bon fonctionnement du projet.

- Le compteur permet de le charger en volume d'eau, de calculer la consommation d'eau, de transmettre ces données au client et de les envoyer au serveur du gestionnaire de l'eau.
- Le client lui, peut interroger le compteur via la fonction SMS pour connaître l'état de sa consommation, peut aussi contrôler l'ouverture et la fermeture du compteur à distance.
- La compagnie des eaux, fournit le compteur, s'occupe de la gestion des recharges des cartes permettant au client de charger son compteur en volume d'eau. Elle tient également une base de données en ligne qui permet de connaître l'état de consommation du client, que ça soit de façon globale ou de façon mensuelle. Cette base de données contient aussi un registre des entrées de chaque recharge effectuée par le client.

### **2. Les capacités du compteur.**

Pour ne pas tomber dans les mêmes tares des compteurs d'eau analogiques conventionnels, le compteur intelligent de notre projet se doit d'être plus évolué et apporter plus de facilités tant au client et qu'à la compagnie d'eau.

Ainsi, pour ce faire le compteur doit être capable de :

- Faire un débitage de l'eau et un arrêt automatique

Même si l'idée est de moderniser le concept de compteur d'eau, il faut bien sûr assurer la fonction principale à savoir mesurer le volume d'eau consommé par le client et dans notre cas le débitmètre sera électronique.

Notre compteur aura la capacité de compter le volume d'eau totale consommée de et surtout de débiter le volume d'eau acheté à chaque consommation. A la fin de la consommation du volume d'eau acheté il s'arrêtera automatiquement et ne recommencera le débitage que lors de l'activation du compte avec un volume d'eau.

- Permettre de charger le compte du client en un volume d'eau quand le client le souhaite :

Pour ce faire, comme cité plus haut nous avons choisi la technologie RFID. Ainsi lors de la livraison du compteur prépayé la compagnie d'eau fournira au client une carte RFID avec un identifiant unique, et qui sera intégré dans le code de fonctionnement du compteur, afin de faire correspondre uniquement cette carte à un unique compteur.

Pour effectuer la recharge de la carte, dans notre modèle, le client devra aller dans une agence de compagnie d'eau ou à un point relais qui permette cette recharge. Cette recharge aura comme valeur 2000 F CFA, 5000 F CFA et 10 000 FCFA. Ceci permettant d'avoir une trace écrite de la recharge de la carte effective du client.

Après la recharge, le client viendra poser sa carte à une distance proche du lecteur RFID et après la lecture de la carte, le système va lire la somme inscrite dans la carte et convertira cette somme en un volume d'eau correspondant à la somme payée. Le volume d'eau généré par cette recharge n'aura pas de limite de temps d'utilisation et ne s'achèvera que lorsque qu'il sera épuisé. Et il y'aura possibilité d'ajouter un volume d'eau avant son épuisement à partir d'un volume seuil.

- Communiquer avec le client et la compagnie des eaux

Le client devra être capable de connaître son état de consommation à tout moment qu'il le souhaite et le compteur devra lui envoyer des informations à des moments précis. Du côté de la compagnie des eaux, le compteur enverra non seulement ces mêmes informations au serveur central lui appartenant mais également d'autres informations via SMS telles que la géolocalisation du compteur.

Pour faire fonctionner le côté communications, nous proposons d'utiliser la technologie GSM dans sa globalité du fait de son faible coût et de la bonne couverture de ce réseau dans tout le pays.

Cette solution impose à la compagnie des eaux à passer un contrat avec un opérateur de téléphonie mobile GSM qui équipera tous les compteurs d'une carte SIM permettant à ces derniers de communiquer avec le téléphone du client et le serveur de la compagnie des eaux.

Pour les clients, nous proposons d'utiliser le SMS, parce que nous faisons le constat que tous les clients n'ont pas forcément accès à un Smartphone, pour pouvoir se connecter en ligne afin d'avoir ses états de consommation. Nous avons aussi choisi cette option pour permettre à un agent de la compagnie de localiser le compteur.

Ainsi nous prévoyons que le client pourra utiliser des mots clés précis envoyées par SMS pour pouvoir avoir son état de consommation et fermer/ouvrir le compteur à distance. Le système sera ainsi programmé à des heures précises pour envoyer l'état de consommation. Par le canal du SMS il recevra des messages de confirmation d'activation du compte d'eau, du compteur et le volume d'eau acheté.

Concernant l'envoi des données sur le serveur en ligne, nous utilisons les fonctions Internet du GSM. Ces données contiendront l'état de consommation du client, la périodicité horaire et mensuelle, ainsi que l'effectivité de l'activation du compte en eau.

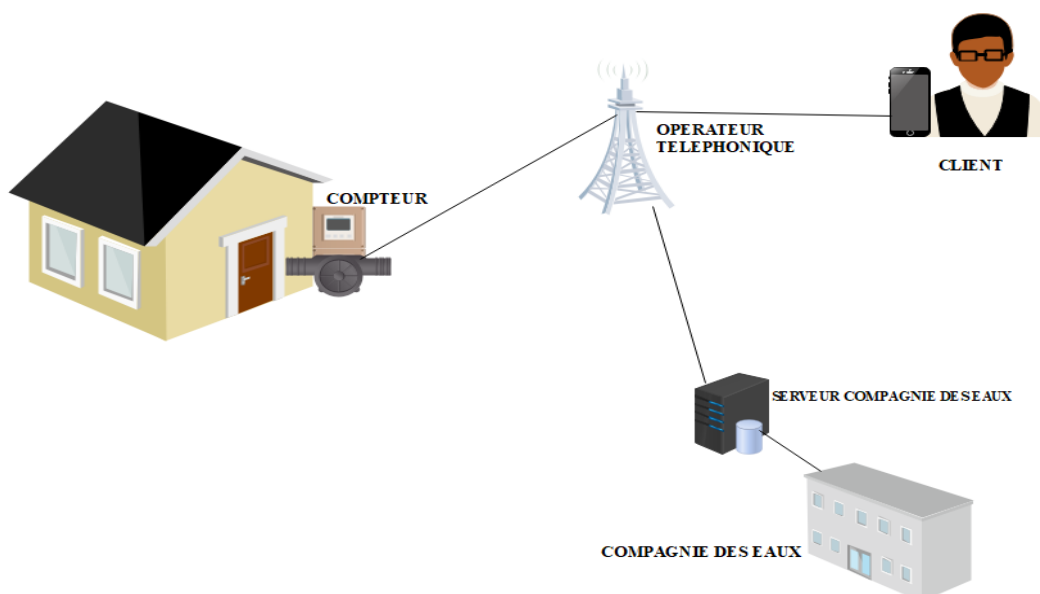


Figure II-1 : Communication entre le compteur/le client/la compagnie des eaux

Après avoir listé les grands axes de notre compteur, nous présentons ici de façon détaillées les actions qu'il est sensée effectuer :

- Recharger son compte en volume d'eau avec une carte à technologie RFID
- Afficher sur un écran l'état de consommation globale, et le volume d'eau restant après activation du compte.
- Envoyer sur une base de données l'état de consommation d'eau de façon périodique.
- Envoyer de manière automatique et périodique l'état de consommation au client via SMS.
- Permettre au client de connaître son état de consommation à tout moment grâce à une simple demande SMS.
- Permettre à la compagnie des eaux de connaître la position exacte de chaque compteur associé à un client précis.
- Permettre de bloquer le compteur à distance par le client s'il constate un souci de fuite par exemple.
- Être informé de la fin proche du cubage acheté et avoir un blocage automatique du compteur en cas de consommation totale du volume acheté.

### **3. Organigramme de fonctionnement du compteur**

Concernant le compteur, nous avons un organigramme qui explique son fonctionnement, dans deux états : Au démarrage du compteur et au moment du fonctionnement de celui-ci.

#### ➤ La phase de Démarrage

Lors de l'allumage du compteur, la première action sera de paramétrer les fonctions liées aux applications telles que le SMS, Internet et GPS.

Ensuite à la fin de cette phase, du fait que le volume d'eau acheté n'est pas encore activé, il est encore à zéro litre, cela implique une fermeture de l'électrovanne du compteur et par la suite l'activation du module RFID pour pouvoir activer son volume d'eau et simultanément l'affichage sur l'écran invitant le client à recharger son compte en eau.

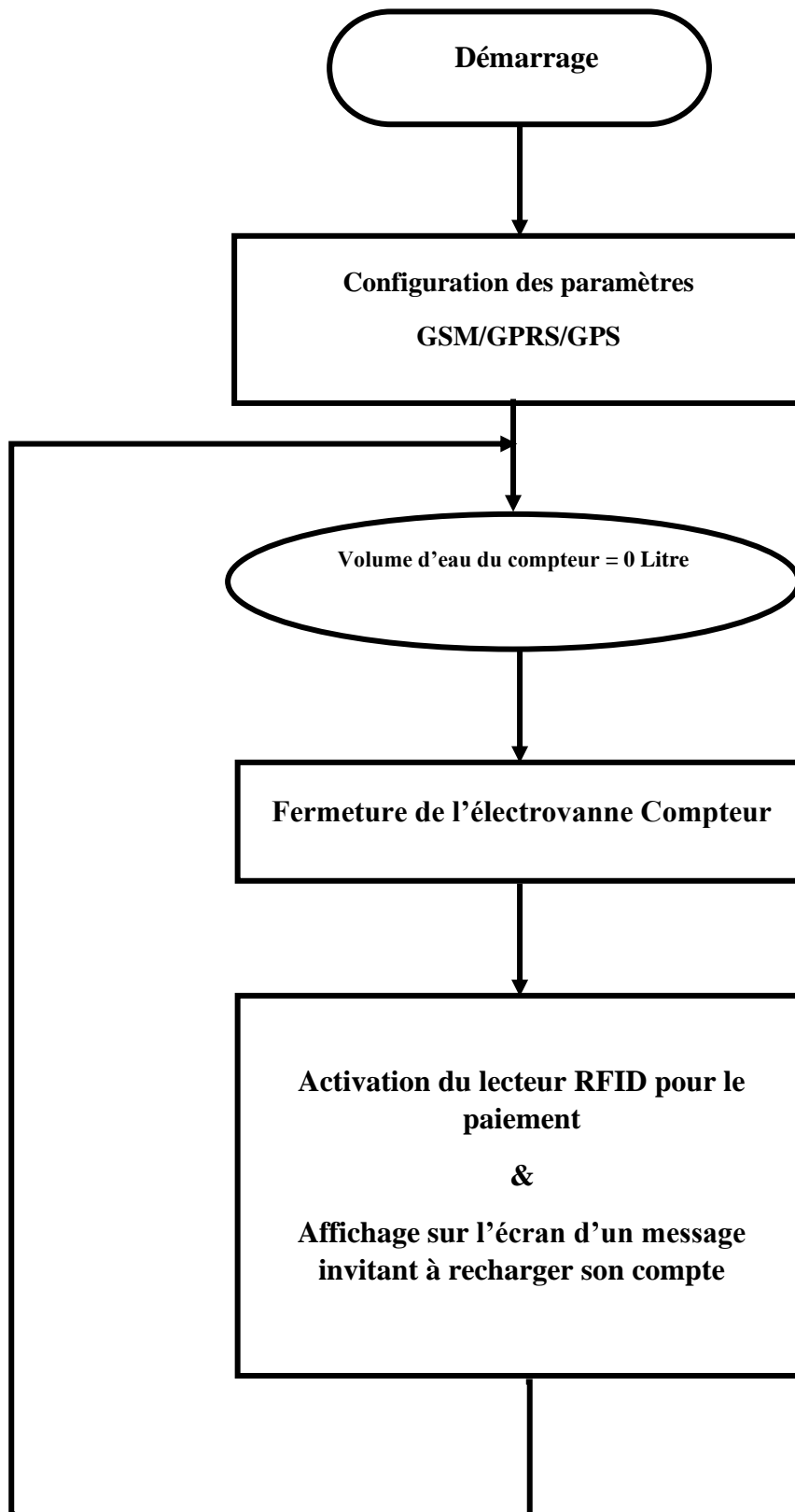
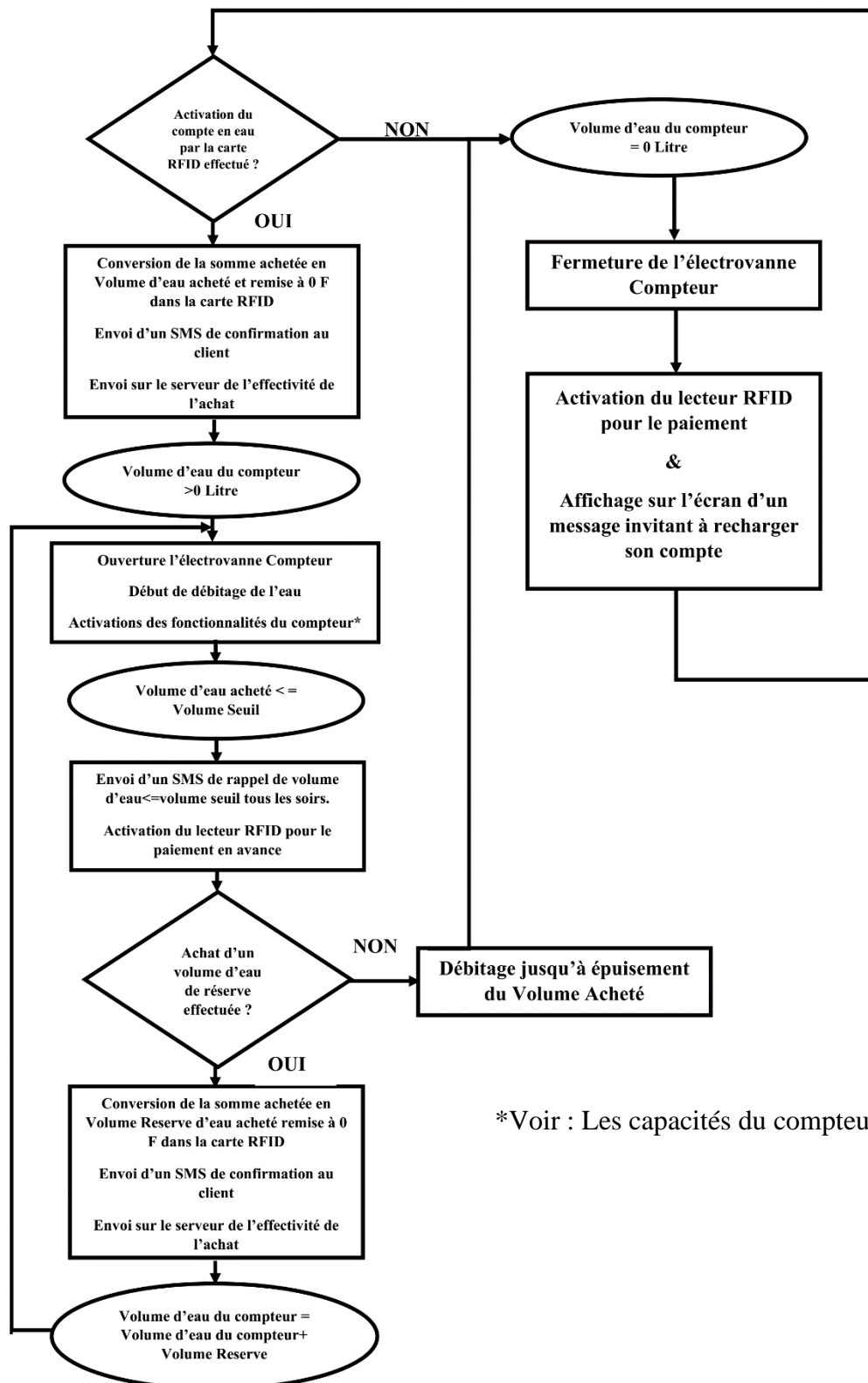


Figure II-2 : Organigramme du compteur au démarrage



➤ La phase de fonctionnement du compteur



\*Voir : Les capacités du compteur.

Figure II-3 : Organigramme du compteur pendant son fonctionnement

Pour le fonctionnement du compteur, on part du choix si le client active son compte en eau ou pas.

- Si le compte n'est pas activé, le volume d'eau du compte reste à zéro, ceci activera le lecteur de carte RFID pour le paiement de son compte et l'affichage d'un message sur l'écran qui va tourner en boucle, invitant le client à recharger son compte en eau.
- Si le compte activé, le système convertit l'argent acheté en volume d'eau, et il y a un envoi de SMS de confirmation au client et d'un envoi au serveur de l'effectivité de l'achat. Ainsi le volume d'eau est supérieur à zéro, entraînant l'ouverture automatique de l'électrovanne et début de débitage. Et toutes les fonctionnalités internes du compteur sont activées.

En continuant le débitage, nous arrivons à un volume seuil qui enclenche l'envoi d'un SMS au client le prévenant de la fin prochaine de son volume d'eau acheté de façon quotidienne et activant la possibilité de payer en avance un volume d'eau de réserve qui s'ajoutera immédiatement au volume d'eau précédent, si le compte est effectivement activé. Et ceci permet de rester dans le cycle normal de fonctionnement. Au cas contraire, le débitage va continuer jusqu'à la consommation totale du volume d'eau acheté, entraînant l'entrée dans le cycle de non activation de volume d'eau.

## **4. Le matériels et les logiciels**

Présenter le fonctionnement de notre système est intéressant, mais l'idéal serait de réellement le matérialiser plus tard. Ainsi dans la conception de notre projet, nous proposons une liste d'utilitaires et de matériels nécessaires pour faire marcher notre système.

### **4.1. Le matériel**

Dans le cadre de la conception et la réalisation de notre compteur et au vu des diverses fonctions implantées à celui-ci, un certain nombre de composants et de module devront être utilisés. Ainsi nous allons présenter le fonctionnement global du compteur en lui-même, faire une étude théorique et au dimensionnement de ces composants et modules. Pour ce faire nous allons présenter leurs principes de fonctionnement, les caractéristiques et critères de choix ainsi que son dimensionnement.

### 4.1.1. Synoptique Global du compteur

Le synoptique ci-dessous présente la disposition globale de notre compteur ainsi que les composants et les modules qui composent notre système.

Ainsi il se compose de :

- Le bloc d'alimentation qui envoie une tension de 12 V.
- L'unité de traitement qui va commander tout le système.
- Le Module GSM/GPS qui va servir à la gestion des notifications par SMS au client et à la compagnie des eaux. Et aussi l'envoi des données de consommations sur un serveur en ligne.
- L'écran LCD qui va servir à afficher des informations sur la consommation, ainsi que la date et l'heure.
- Le lecteur RFID pour la lecture des cartes afin charger le compteur en volume d'eau.
- Une électrovanne pour gérer les coupures d'eau automatiques ou via un ordre du client auprès de la compagnie des eaux.
- Des relais pour pouvoir commander les électrovannes.
- Un module horaire RTC.
- Un débitmètre dont le rôle sera de mesurer le volume d'eau consommé.

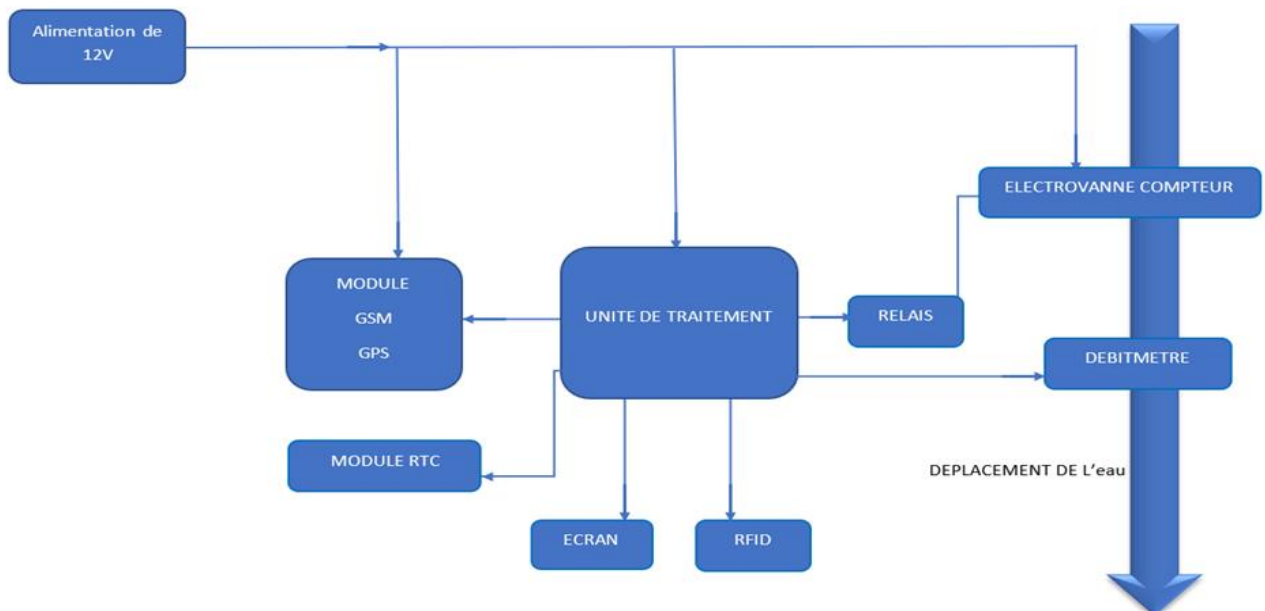


Figure II-4 : Synoptique Global du compteur

## **4.1.2. Le choix des composants matériels du compteur**

En se basant sur le synoptique du compteur et vu le travail de recherche fait au niveau universitaire, un choix de matériel bien précis sera nécessaire pour pouvoir réaliser une maquette composée d'éléments bien précis qui vont concourir au bon fonctionnement de l'ensemble du système pensé lors de la conception.

Ainsi nous avons :

- L'unité de traitement qui est une Carte ARDUINO MEGA 2560.
- Le module GSM/GPS qui est une SIM 808 EVB 3.2.4
- Le débitmètre qui est un débitmètre YF-S201
- La module RFID qui est la RFID RC522.
- L'écran qui est un écran LCD 16×2
- Le relais qui est un 2 Relay Module de 12 V
- Le module RTC DS3231
- Les électrovannes qui sont des Electrovannes 12V /DC
- Les fils de branchements et des LED

### **4.1.2.1. La carte ARDUINO MEGA 2560**

Tout d'abord, nous devons revenir à l'origine d'Arduino. Elle est une carte électronique programmable et utilisée comme microcontrôleur pour la conception et le contrôle de composants électroniques, mécaniques, domotiques ou robotiques. En effet, grâce à Arduino, nous pouvons créer des systèmes amateurs ou professionnels intelligents, tels que des systèmes de contrôle pour divers composants de la maison. Grâce à Arduino, nous pouvons aussi utiliser notre smartphone comme télécommande, par exemple pour allumer ou éteindre les lumières de la maison, ou pour atténuer ou augmenter les volets électriques, régler la température à distance de la maison en contrôlant le chauffage, etc.

Les cartes Arduino ont plusieurs modèles tels que la Arduino UNO R3, la plus connue et utilisée, et aussi la Arduino Nano. Cependant l'Arduino Méga est le modèle le plus perfectionné et puissant de la célèbre carte électronique. Elle permet d'effectuer un maximum d'actions et délivre un potentiel tel, qu'il est possible de se pencher sur les montages les plus lourds et gourmands en code. Cette carte peut être utilisée par les amateurs confirmés, mais est principalement destinée aux experts qui pourront en faire un usage plus professionnel. La carte

dispose d'un brochage spécifique par rapport aux autres cartes. Par rapport à d'autres cartes Arduino, la MEGA a une disposition de broches spécifiques. Cela rend incompatible avec certains composants.

### i. Vue d'ensemble

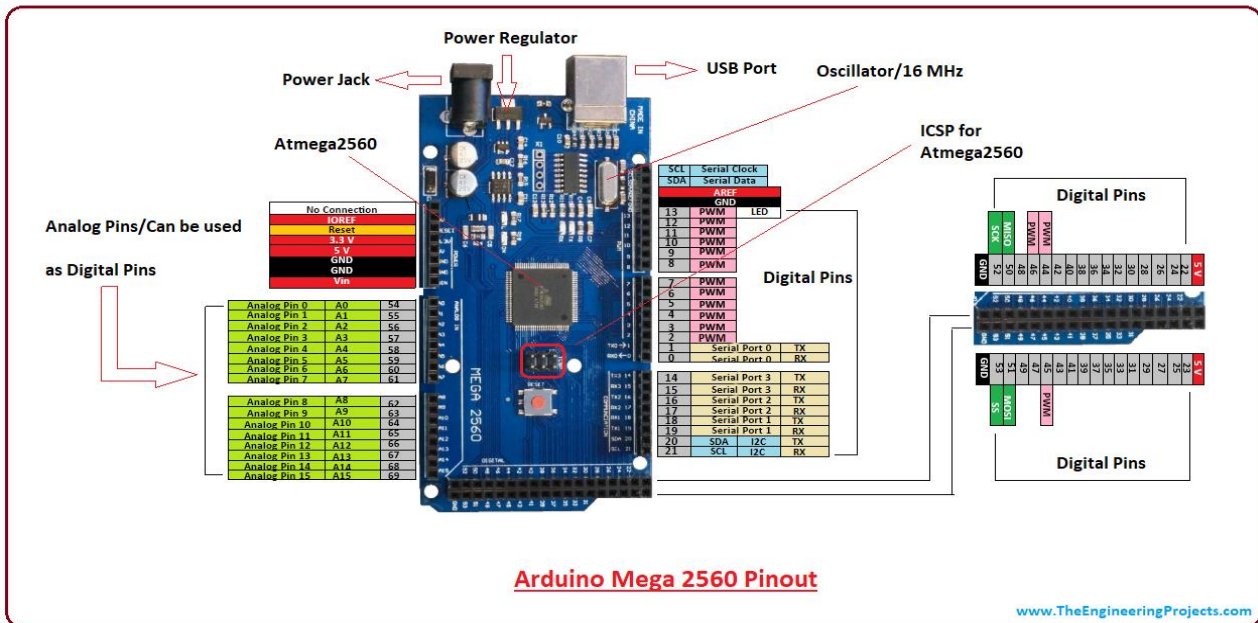


Figure II-5 : Représentation photographique de l'Arduino MEGA [12]

La carte Arduino MEGA 2560 est une carte à microcontrôleur basée sur un ATMEGA2560 et qui dispose de :

- 54 Broches numériques d'entrées/sorties dont 14 peuvent être utilisées en sorties PWM (largeur d'impulsion modulée),
- 16 entrées analogiques (qui peuvent également être utilisées en broches entrées/sorties numériques),
- 4 UART (port série matériel),
- Un quartz 16Mhz,
- Une connexion USB,
- Un connecteur d'alimentation jack,
- Un connecteur ICSP (programmation « in-circuit »),
- Un bouton de réinitialisation (Reset).

### ii. Fiche Technique

<b>Microcontrôleur</b>	ATMEGA2560
<b>Tension de fonctionnement</b>	5V
<b>Tension d'alimentation (Recommandée)</b>	7-12V
<b>Tension d'alimentation (Limites)</b>	6-20V
<b>Broches E/S numériques</b>	54 (dont 15 disposent d'une sortie PWM)
<b>Broches d'entrées analogiques</b>	16 (utilisables en broches E/S numériques)
<b>Intensité max. disponible par broche E/S (5V)</b>	40 mA (ATTENTION : 200mA cumulé pour l'ensemble des broches E/S)
<b>Intensité maxi disponible pour la sortie 3.3V</b>	50 mA
<b>Intensité maxi disponible pour la sortie 5V</b>	Fonction de l'alimentation utilisée - 500 mA max. si port USB utilisé seul
<b>Mémoire Programme Flash</b>	256 KB dont 8 KB sont utilisées par le bootloader
<b>Mémoire SRAM (mémoire volatile)</b>	8 KB
<b>Mémoire EEPROM (mémoire non volatile)</b>	4 KB
<b>Vitesse d'horloge</b>	16 MHz
<b>LED_BUILTIN</b>	13
<b>Longueur</b>	101.52 mm
<b>Largeur</b>	53.3 mm
<b>Poids</b>	37 g

Tableau II-1 : Fiche Technique Arduino MEGA [11]

### iii. L'alimentation

La carte Arduino MEGA 2560 peut être connectée via le port USB pour fournir 5V à 500mA ou utiliser une alimentation externe. La carte sélectionnera automatiquement la source d'alimentation. La source d'alimentation externe (non USB) peut être un adaptateur secteur fournissant généralement une tension de 3V à 12V à 500mA. L'adaptateur secteur peut être connecté en insérant la prise positive de 2,1 mm au centre du connecteur jack de la carte. On peut insérer les fils de la batterie dans le connecteur des broches de la carte. Les noms de ces broches sont GND (la masse ou 0V) et VIN (la tension d'entrée positive) du connecteur d'alimentation.

La carte peut fonctionner avec une alimentation externe de 6 à 20 volts. Cependant, si la tension d'alimentation de la carte est inférieure à 7 V, la tension d'alimentation de la broche 5 V peut être inférieure à 5 V et potentiellement rendre la carte instable. Si la tension utilisée dépasse 12 V, le régulateur de tension de la carte peut chauffer et endommager la carte. De plus, il est recommandé que la plage idéale pour alimenter la carte MEGA soit de 7V à 12V.

Les broches d'alimentation sont les suivantes :

- VIN. : La tension d'entrée positive lorsque la carte Arduino est utilisée avec une source de tension externe (à distinguer du 5V de la connexion USB ou autre source 5V régulée). On peut alimenter la carte à l'aide de cette broche, ou, si l'alimentation est fournie par le jack d'alimentation, accéder à la tension d'alimentation sur cette broche.
- 5V. : La tension régulée utilisée pour faire fonctionner le microcontrôleur et les autres composants de la carte par ce que les circuits électroniques numériques nécessitent une tension d'alimentation parfaitement stable dite « tension régulée » obtenue à l'aide d'un composant appelé un régulateur et qui est intégré à la carte Arduino. Le 5V régulé fourni par cette broche peut donc provenir soit de la tension d'alimentation VIN via le régulateur de la carte, ou bien de la connexion USB ou de tout autre source d'alimentation régulée.
- 3V3. : Une alimentation de 3.3V fournie par le circuit intégré FTDI (circuit intégré faisant l'adaptation du signal entre le port USB de l'ordinateur et le port série de l'ATMEGA) de la carte est disponible, ceci est intéressant pour certains circuits externes nécessitant cette tension au lieu du 5V. L'intensité maximale disponible sur cette broche est de 50mA
- GND. : Broche de masse (ou 0V).

Note : Protection du port USB contre la surcharge en intensité

La carte Arduino MEGA2560 contient un multi-fusible réinitialisable pour protéger le port USB de l'ordinateur contre les surcharges. Bien que la plupart des ordinateurs aient leur propre protection interne, les fusibles de la carte offrent une couche de protection supplémentaire. Si le courant appliqué au port USB dépasse 500 mA, le fusible sur la carte coupera automatiquement la connexion jusqu'à ce que le court-circuit ou la surcharge s'arrête.

#### **iv. La mémoire**

L'ATMEGA 2560 contient 256 Ko de mémoire FLASH pour stocker le programme (dont 8Ko également utilisés par le bootloader). L'ATMEGA 2560 a également 8 Ko de mémoire SRAM (volatile) et 4 Ko d'EEPROM (non volatile - mémoire qui peut être lue à l'aide de la librairie EEPROM).

NB. : Le bootloader est un programme préprogrammé une fois pour toute dans l'ATMEGA et qui permet la communication entre l'ATMEGA et le logiciel Arduino via le port USB, notamment lors de chaque programmation de la carte.

## v. Les Broches Numériques

En utilisant les commandes `pinMode ()`, `digitalWrite ()` et `digitalRead ()` du langage Arduino, chacune des 54 broches numériques de la carte MEGA peut être utilisée comme entrée numérique ou sortie numérique. Ces broches fonctionnent en 5V. Chaque broche peut fournir ou recevoir un maximum de 40mA d'intensité et dispose d'une résistance interne de de 20-50 KOhms dite de « rappel au plus » (en anglais pull-up) et qui est déconnectée par défaut. Cette résistance interne s'active sur une broche en entrée à l'aide de l'instruction `digitalWrite(broche, HIGH)`.

Certaines broches ont des fonctions bien spéciale :

- Communication Série :

Port Série Serial 0 : 0 (RX) & 1 (TX) ;

Port Série Serial 1 : 19 (RX) & 18 (TX) ;

Port Série Serial 2 : 17 (RX) & 16 (TX) ;

Port Série Serial 3 : 15 (RX) & 14 (TX).

Utilisées pour recevoir (RX) et transmettre (TX) les données séries de niveau TTL. Les broches 0 (RX) & 1 (TX) sont connectées aux broches correspondantes du circuit intégré ATMEGA8U2 programmé en convertisseur USB-vers-série de la carte, composant qui assure l'interface entre les niveaux TTL et le port USB de l'ordinateur.

- Interruptions Externes :

Broches : 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), et 21 (interrupt 2).

Ces broches peuvent être configurées pour déclencher une interruption sur une valeur basse, sur un front montant ou descendant, ou sur un changement de valeur. Voir l'instruction `attachInterrupt()` pour plus de détails.

- Impulsion PWM (largeur d'impulsion modulée) :



Broches 2 à 13 et 44 à 46. Fournissent une impulsion PWM 8-bits à l'aide de l'instruction `analogWrite()`.

- SPI (Interface Série Périphérique) :

Broches 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS).

Ces broches supportent la communication SPI (Interface Série Périphérique) disponible avec la librairie pour communication SPI. Les broches SPI sont également connectées sur le connecteur ICSP qui est mécaniquement compatible avec les cartes Uno, Duemilanove et Diecimila.

- I2C :

Broches 20 (SDA) et 21 (SCL).

Supportent les communications de protocole I2C (ou interface TWI (Two Wire Interface - Interface «2 fils »), disponible en utilisant la librairie `Wire/I2C` (ou TWI - Two-Wire interface - interface «2 fils »).

- LED :

Broche 13. Il y a une LED incluse dans la carte connectée à la broche 13. Lorsque la broche est au niveau HAUT, la LED est allumée, lorsque la broche est au niveau BAS, la LED est éteinte.

## **vi. Les Broches Analogiques**

La carte MEGA2560 dispose de 16 entrées analogiques, chacune pouvant fournir une mesure d'une résolution de 10 bits (c'est-à-dire sur 1024 niveaux soit de 0 à 1023) à l'aide de la très utile fonction `analogRead()` du langage Arduino. Par défaut, ces broches mesurent entre le 0V (valeur 0) et le 5V (valeur 1023), mais il est possible de modifier la référence supérieure de la plage de mesure en utilisant la broche AREF et l'instruction `analogReference()` du langage Arduino. Il faut noter que les broches analogiques peuvent être aussi utilisées en tant que broches numériques.

## **vii. Les Autres Broches**

Il y a deux autres broches disponibles sur la carte :

- AREF : Tension de référence pour les entrées analogiques (si différent du 5V). Utilisée avec l'instruction analogReference().
- Reset : Mettre cette broche au niveau BAS entraîne la réinitialisation du microcontrôleur. Typiquement, cette broche est utilisée pour ajouter un bouton de réinitialisation sur le circuit qui bloque celui présent sur la carte.

### **viii. La communication**

La carte Arduino MEGA2560 dispose de toute une série de facilités pour communiquer avec un ordinateur, une autre carte Arduino, ou avec d'autres microcontrôleurs. L'ATMEGA2560 dispose de quatre UARTs (Universal Asynchronous Receiver Transmitter ou émetteur-récepteur asynchrone universel en français) pour une communication série de niveau TTL (5V) et qui est disponible sur les broches 0 (RX) et 1 (TX). Un circuit intégré ATMEGA8U2 sur la carte assure la connexion entre cette communication série de l'un des ports série de l'ATMEGA 2560 vers le port USB de l'ordinateur qui apparaît comme un port COM virtuel pour les logiciels de l'ordinateur. Le code utilisé pour programmer l'ATmega8U2 utilise le driver standard USB COM, et aucun autre driver externe n'est nécessaire. Cependant, sous Windows, un fichier .inf est requis.

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

Une librairie Série Logicielle permet également la communication série (limitée cependant) sur n'importe quelle broche numérique de la carte UNO.

L'ATMEGA2560 supporte également la communication par protocole I2C (ou interface TWI (Two Wire Interface - Interface « 2 fils ») et SPI :

Le logiciel Arduino inclut la librairie Wire qui simplifie l'utilisation du bus I2C.

Pour utiliser la communication SPI (Interface Série Périphérique), la librairie pour communication SPI est disponible.

#### 4.1.2.2. Le module GSM/GPRS/GPS SIM 808 EVB 3.2.4

De nombreux modules compatibles avec les cartes Arduino permettent d'utiliser les services du GSM comme l'appel, les SMS et la connexion sur le réseau internet avec une fonction GPRS, c'est le cas du module SIM900 et d'autres permettent aussi d'utiliser une fonction GPS c'est le cas du module SIM68M.

Mais dans le cadre de notre projet, nous avons souhaité avoir un module qui pourrait gérer toutes ces fonctions une seule fois et c'est ainsi que nous nous sommes tournées vers le module SIM808. Le module SIM808 est un module complet Quad-Band GSM/GPRS qui combine la technologie GPS pour la navigation par satellite.

La conception compacte qui a intégré le GPRS et le GPS dans un package SMT permet d'économiser considérablement le temps et les coûts pour des projets combinant ces deux fonctions. Notre version utilisée est la version EVB 3.2.4 qui a la particularité d'avoir aussi une fonction Bluetooth.

La puce de commande du SIM808 qui donne son nom au module entier est fabriqué par l'entreprise chinoise SIMCom Wireless Solutions qui est un des leaders mondiaux des modules M2M Wireless.M2M qui renvoie à Machine 2 Machine et se traduit par la communication de machine à machine. Le M2M utilise les télécommunications et l'informatique pour permettre des communications entre machines, et ceci sans intervention humaine.

##### i. Vue d'ensemble

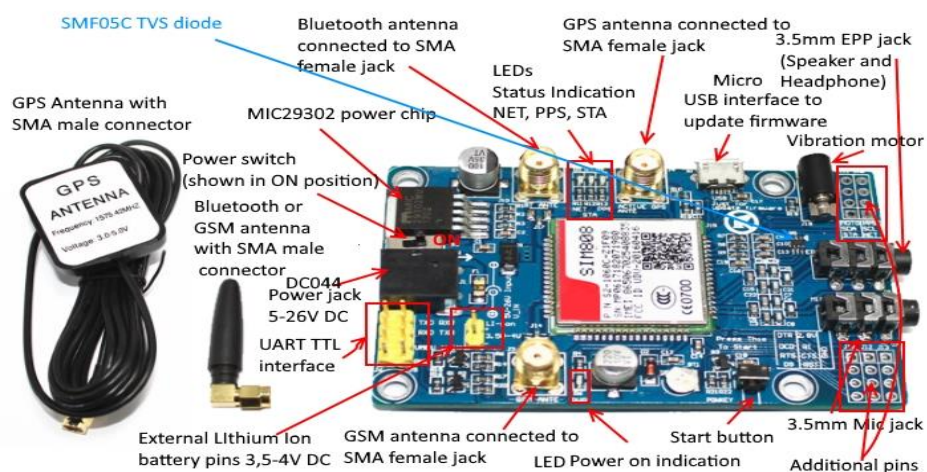


Figure II-6 : Vue de face SIM808 EVB 3.2 [13]

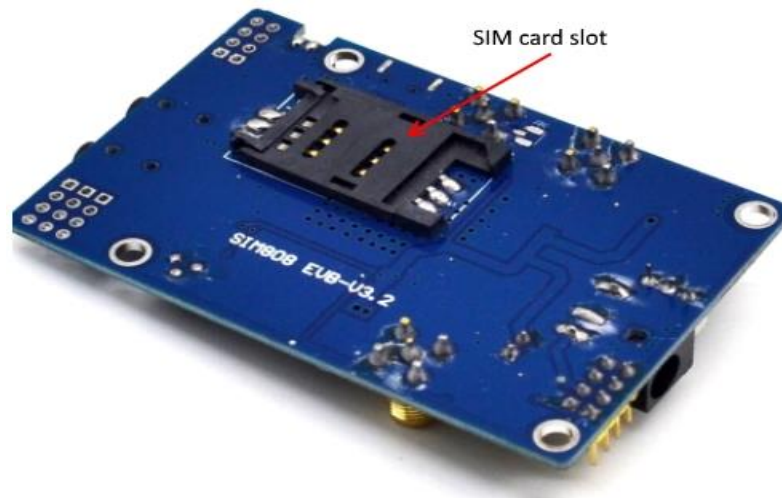


Figure II-7 : Vue arrière SIM808 EVB 3.2 [13]

## ii. Fiche Technique

### ➤ Les fonctions :

- Envoyer et recevoir des données GPRS (TCP / IP, HTTP, etc.)
- Recevoir des données GPS et des données A-GPS
- Envoyer et recevoir des SMS
- Passer et recevoir des appels téléphoniques

### ➤ Caractéristiques générales :

- Quadri-bande 850/900/1800/1900 MHz
- Classe GPRS multi-slots 12/10
- Station mobile GPRS classe B
- Conforme à GSM phase 2/2 + (Classe 4 (2 W @ 850 / 900 MHz ; Classe 1 (1 W @ 1800 / 1900MHz))
- Bluetooth : compatible avec 3.0 + EDR
- FM : Bandes mondiales de 76 ~ 109 MHz avec pas de réglage de 50 kHz
- Dimensions : 24,0 × 24,0 × 2,6 mm
- Poids : 3,30 g
- Contrôle via les commandes AT (3GPP TS 27.007, 27.005 et commandes AT améliorées SIMCOM)
- Alimentation de 7 à 12 V via le port Jack

- Faible consommation d'énergie
- Température de fonctionnement : de -40 °C à 85 °C
- Spécifications pour les données GPRS :
  - Classe GPRS 12 : max. 85,6 kbps (liaison descendante / liaison montante)
- Spécifications pour SMS via GSM / GPRS :
  - Point à point MO et MT
  - Diffusion de la cellule SMS
  - Mode texte et PDU
- Spécifications pour GPS :
  - Type de récepteur (22 voies de suivi / 66 canaux d'acquisition ; code GPS L1 C / A)
  - Sensibilité (suivi : -165 dBm ; démarrages à froid : -147 dBm)
  - Time-To-First-Fix (démarrages à froid : 30s (typ.) ; Démarrages à chaud : 1s (typ.) ; Démarrages à chaud : 28s (typ.))
  - Précision (position horizontale : <2,5 m CEP)
- Connexion : 68 pads SMT comprenant :
  - Port Antenne GSM
  - Port Antenne GPS
  - Port Antenne Bluetooth
- Compatibilité :
  - Interface de commande cellulaire AT
- Module SIM808 :
  - Prend en charge les cartes SIM : 2G /3G /4G
  - Peut être utilisé avec Raspberry Pi et Arduino
  - Comprend des modules d'antenne à savoir SMA : GSM, GPS et Bluetooth.
  - Comprend trois interfaces d'entrée d'alimentation : à savoir DC044, V\_IN et une interface de batterie au lithium.
  - La plage de DC044 et l'entrée de tension de la broche V\_IN est de 5 à 26 V, lorsqu'on utilise le 5 V comme alimentation, assurons-nous que l'alimentation peut fournir un courant de 2 A. La plage de tension de la puissance d'entrée de la batterie au lithium est de 3,5 à 4,2 V.
  - Comprend un port USB, qui est utilisée pour mettre à jour le firmware du module SIM808.

### iii. Le branchement

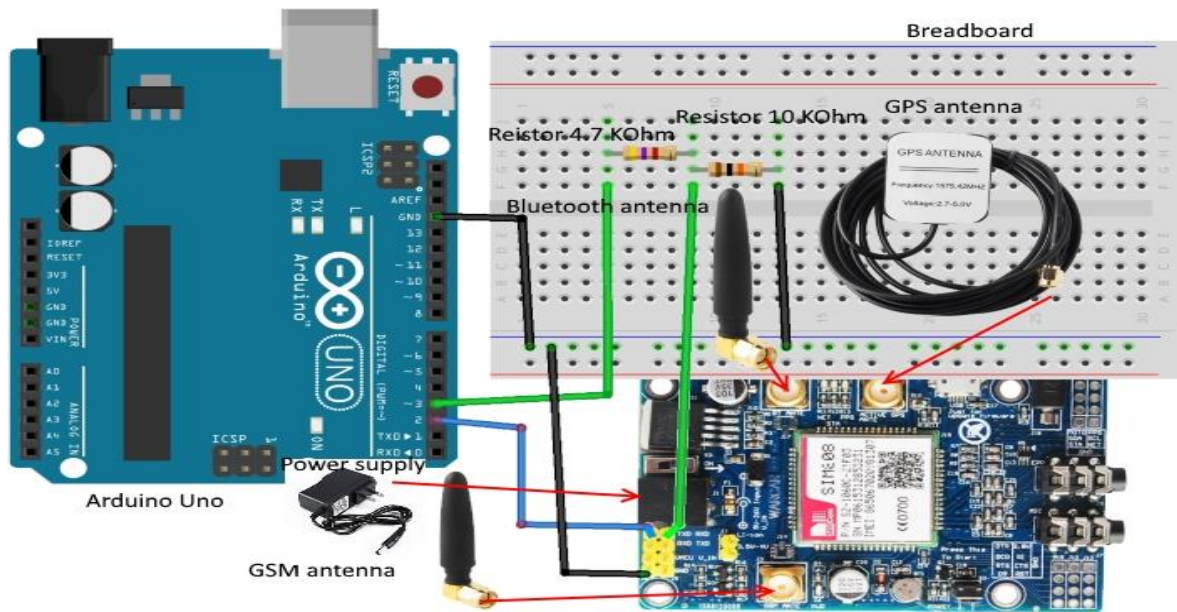
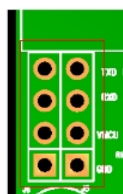


Figure II-8 : Représentation du branchement de la SIM808 [13]

Pour bien faire fonctionner le SIM808 la base est de brancher les antennes GPS et GSM au module, sinon aucune de ses deux fonctions ne fonctionnera pas et aussi il faut connecter la carte à une source d'énergie qui se fait principalement à travers un transformateur envoyant une tension de 12 V de façon idéale.

Concernant le branchement à la carte Arduino de façon standard, on utilise trois broches de la carte à savoir :

- Tx qui se connecte sur le port Rx de la carte Arduino ;
- Rx qui se connecte sur le port Tx de la carte Arduino ;
- GND pour la connexion à la masse.



TXD	RXD
RXD	TXD
VMCU	V_IN
GND	GND

Figure II-9 : Broches essentiels au branchement de la SIM808 [13]

Il faut noter que ce branchement ne permet pas directement de faire fonctionner la carte, pour ce faire il faut appuyer sur le bouton « POWKEY » pendant environ 3 secondes. Pour

remédier à cette situation, le module dispose d'un port appelé « D9 » mais qui n'est pas broché de base, et cette broche se comporte comme un peu comme une LED qu'on peut commander pour allumer via la carte Arduino. Ainsi donc dans le code, on mettra une ligne de code qui activera ce port « D9 » et qui activera automatiquement la carte au démarrage.

#### iv. Les commandes AT

Le module SIM808 offrant beaucoup de fonctionnalités nécessite pour son bon fonctionnement des commandes AT, qui sont le seul moyen d'activer les fonctions liées au GSM, GPRS et GPS.

##### ➤ Généralités

Les commandes AT (AT est l'abréviation d'ATtention) aussi connues sous le nom de Commandes Hayes, constituent un langage de commandes développé à l'origine pour le modem Hayes Smartmodem 300 par Dennis Hayes en 1981 et qui permettent la gestion complète du modem et autres modules à savoir par exemple :

L'ensemble de commande se compose d'une série de courtes chaînes de texte codé en ASCII et commençant obligatoirement par les caractères « AT » qui peuvent être combinées pour produire des commandes pour des opérations telles que passer des appels, les raccrocher ou modifier des paramètres de la connexion.

Ils répondent à la norme GSM 07.07(pour les SMS cf. GSM 07.05).

##### ➤ Fonctionnement

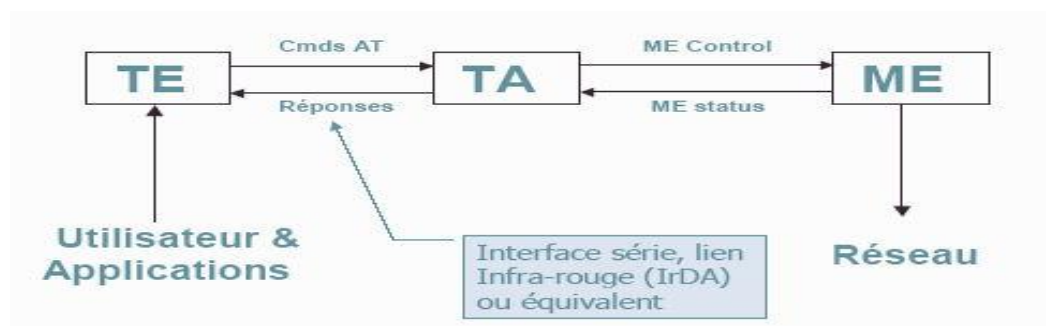


Figure II-10 : Schéma fonctionnement Commande AT [14]

Pour comprendre le fonctionnement des commandes AT, trois entités sont nécessaires :

- Le TE : Terminal Equipment (envoi et affiche les commandes.)
- Le TA : Terminal Adaptator (l'interface entre l'utilisateur et l'équipement mobile).
- ME : Mobile Equipment (le module/modem)

Ainsi l'utilisateur envoie sa commande souhaitée, le TA fait un contrôle et le transmet à l'équipement mobile et l'action est lancée. Que l'action soit concluante ou pas, l'équipement mobile transfère l'état de l'action (positive ou pas) et le TA renvoie à l'utilisateur la réponse appropriée suite à la commande souhaitée.

### ➤ Structure d'une commande AT

Elle s'écrit de la forme habituelle : AT+CMD1 <CR>. Avec plusieurs spécificités, nous avons :

- AT : Début d'une commande

L'utilisation de la commande AT comme commande unique devra fournir comme réponse « OK ». Elle est utilisée pour vérifier si le module est bien connecté à l'unité de traitement.

- CMD1 : La commande AT Souhaité
- CMD2=3 ; : Correspond au paramètre souhaité dans la commande.
- AT+CMD1 ; : Correspond à la commande étendue (elle commence toujours avec le + et fini avec ;
- AT+CMD2=, ,14 ; : dans ce cas de figure on est face à des commandes avec plusieurs paramètres dont on n'est pas obligé de tout entrer
- AT+CMD2 ? : Correspond à la commande pour connaître l'état de commande souhaité
- AT+CMD2=? : Correspond à une commande qui affiche les valeurs possibles qui peuvent être générées par la commande

Note : CR qui est le Carriage Return est un caractère de contrôle indiquant un retour à la ligne en ASCII, en langage C il sera représenté par /r.

### ➤ Structure d'une Réponse AT



Quand la commande a été exécutée on aura à la fin « OK » ou « ERROR » par exemple selon si l'action a été bien faite ou pas et sous cette forme on parlera de forme étendue. Ou bien à la fin on peut avoir des réponses chiffrées : 1 ou 4 et là on parlera de forme courte. Et les réponses s'affichent différemment en fonction du type d'action faite.

Ainsi si nous prenons les modèles de commandes de la section précédente, on aurait :

- Pour +CMD2?

```
« CR » « LF » +CMD2 :3 ,0 ,15 "GSM" « CR » « LF »
```

- Pour +CMD2=?

```
« CR » « LF » +CMD2 :(0-3) ,(0,1) ,(0-12,15),("GSM","IRA" « CR » « LF »
```

Et le OK à la fin qui va sera de la forme : « CR » « LF » OK « CR » « LF »

Note : LF qui est le Line Feed est un caractère de contrôle indiquant le passage à la ligne de texte suivante en ASCII par contre en langage C il sera représenté par /n.

## ➤ Les commandes AT dans Arduino

Connectée aux cartes Arduino, les modules mobiles de communication fonctionnant avec les commandes AT ont des séries de commandes qui varient en fonction des divers modules, même si la base est standard.

Ainsi pour tester les commandes on peut passer le Moniteur Série, dans laquelle on va taper notre commande et voir directement le résultat. Ou bien écrire la commande dans le code et il s'exécutera après téléversement dans la carte Arduino.

Et il faut noter que l'utilisation des commandes /r ou /n, n'est pas obligatoire au moment d'écrire le code.

NB : Un manuel de l'ensemble des commandes AT compatibles avec le SIM808 est disponible sur le site internet du fabricant permettant de pouvoir exécuter toutes les actions disponibles offertes par ce module ou Shield.

AT	Check module status.
AT + CPIN = "XXXX"	Enter the PIN of the SIM. Change XXXX by the PIN.
AT + CREG?	Check the connection to the network.
ATDXXXXXXXX;	Make a call Replace XXXXXXXXXXX with the number you want to call.
ATA	Pick up a call.
ATH	End the call.
AT + CMGF = 1	Set the text mode to send or receive messages. Returns ">" as inductor.
AT + CMGS = "XXXXXXXXXX"	No. to which we are going to send the message.
AT + CLIP = 1	We activate the call identification.
AT + CNMI = 2,2,0,0,0	We configure the module to show the SMS through the serial port.
AT + CGATT = 1	We connect to the GPRS network.
AT + CSTT = "APN", "user", "password"	We define APN. user and password
AT + CIICR	We activate the wireless data profile
AT + CIFSR	We get our IP
AT + CIPSTART = "TCP", "IP address", "port"	We indicate the type of connection, IP address and port to which we made the connection
AT + CIPSEND	We prepare the sending of data. Returns ">" as inductor.
AT + CIPCLOSE	We closed the connection.
AT + CIPSHUT	Close the PDP context of the GPRS
AT + CGPSPWR = 1	Activate the GPS.
AT + CGPSSTATUS?	Check that the GPS has found the network.
AT + CGPSINF = 0	Get the GPS data: Mode, Latitude, Longitude, Altitude, Timetable, Response time, Number of satellites, Speed, Course
AT + CGPSOUT = 32	to obtain GPS data using the NMEA specification: UTC Time, State, Latitude, Longitude, Speed in knots, Defeat angle in degrees, Date, Magnetic variation, Cheksum data
AT + CGPSPWR = 0	Close the GPS.

Tableau II-2 : Quelques Commandes AT et leurs actions sur le SIM808 [13]

### 4.1.2.3. Le débitmètre YF-S201

Pour la mesure de la consommation en eau de notre compteur, nous allons nous servir d'un débitmètre qui est un appareil destiné à mesurer le débit d'un fluide, liquide ou gazeux.

En fonction du débit et de la nature du fluide, la nature du débitmètre varie beaucoup et nous pouvons citer quelques exemples tels que :

- Des débitmètres à ultrasons, des anémomètres, des débitmètres ioniques qui sont basés sur la mesure de la vitesse du fluide ;
- Des débitmètres massiques ;
- Des débitmètres électromagnétiques pour des liquides conducteurs ;
- Des débitmètres à pistons, à rotors, à turbines qui vont se baser sur le fait d'utiliser le fluide pour entraîner quantitativement des systèmes mécaniques mobiles.

Et c'est dans cette dernière catégorie que se retrouve le débitmètre, principalement à rotors, que nous allons utiliser pour notre compteur, et qui permet de récupérer les données de consommations de façon électronique. Notre choix s'est porté sur le YF-S201.

#### i. Vue d'ensemble



Figure II-11 : Vue Extérieure du YF-S201 [15]



Figure II-12 : Vue Intérieure du YF-S201 [16]

## ii. Fiche Technique

- Type de capteur : Capteur à effet Hall
- Tension de fonctionnement : 5 à 18 V CC
- Consommation de courant max. : 15mA @ 5V
- Type de sortie : 5V TTL
- Débit de travail : 1 à 30 litres / minute
- Plage de température de travail : -25 à + 80 °C
- Plage d'humidité de travail : 35% -80% HR
- Précision :  $\pm 10\%$
- Pression d'eau maximale : 2,0 Mpa = 20 Bars
- Cycle de sortie : 50% + -10%
- Temps de montée en sortie : 0,04us
- Temps de chute de sortie : 0,18us
- Caractéristiques d'impulsion de débit : Fréquence (Hz) =  $7,5 \times \text{Débit (L / min)}$
- Impulsions par litre : 450
- Durabilité : minimum 300 000 cycles
- Longueur de câble : 15 cm
- Raccords de tuyaux nominaux 1/2 ", diamètre extérieur 0,78", filetage 1/2 "
- Taille : 2,5 "x 1,4" x 1,4 "

### iii. Le Fonctionnement

Le YF-S201 est sur un capteur qui fonctionne grâce à l'effet Hall. Effet hall qui est énoncé comme suit : « un courant électrique traversant un matériau baignant dans un champ magnétique, engendre une tension perpendiculaire à ce dernier ».

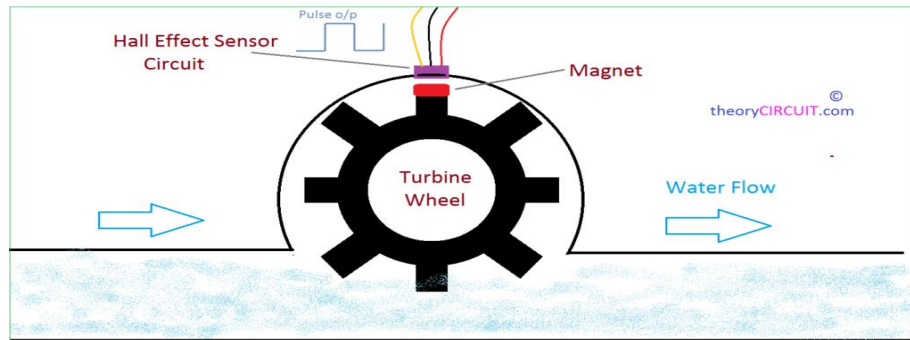


Figure II-13 : Principe de fonctionnement du YF-S201 [17]

Nous avons une turbine à roue encastree avec un aimant qui est placée sur une enveloppe en plastique fermée et un capteur à effet Hall placé de façon fixe. Lorsque l'eau s'écoule à travers le débitmètre, cela fait tourner la roue de la turbine et donc le flux magnétique interfère avec le capteur à effet Hall, le taux d'interférence dépend de la vitesse du flux de l'eau, de sorte que le capteur à effet Hall produit en sortie un signal d'impulsion. Et cette impulsion émise permet de calculer le débit de l'eau ainsi que le volume d'eau par la suite. Le signal d'impulsion est une simple onde carrée, il est donc assez facile d'enregistrer et de convertir en litres par heure en utilisant la formule suivante :

$$\text{Fréquence d'impulsion (Hz)} = 7.5Q$$

Avec Q qui est le débit d'eau en litre/ minutes

De cette formule fournie par le constructeur, nous pouvons en tirer à la fin le volume d'eau consommé dès que le rotor cesse de tourner.

Et ce volume d'eau est obtenu à partir de la formule

$$\text{Volume d'eau (L)} = Q/60$$

#### iv. Le Branchement

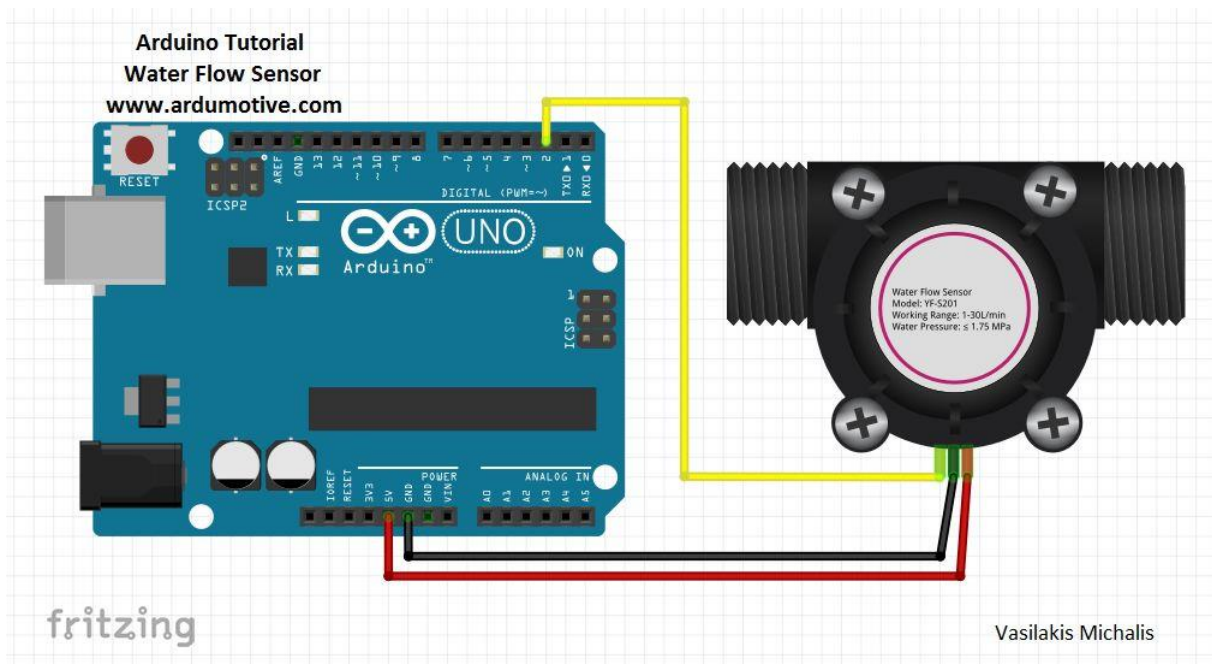


Figure II-14 : Branchement Standard du YF-S201 sur une carte Arduino [15]

Comme vous le voyez les Fils de branchement suivent une logique bien précise :

- Le fils Rouge : se connecte au 5V de l'Arduino.
- Le fil Noir : se connecte au GND de l'Arduino.
- Le fils Jaune : se connecte à un port PWM ou Analogique selon le choix.

#### 4.1.2.4. Le RFID RC522

Pour le paiement de notre compteur, comme précisé plus haut, nous avons fait le choix de le faire avec une carte RFID, technologie RFID que nous avons déjà présentée. Les modules RFID qui permettent une identification et un suivi automatique des cartes contenant des informations et des données stockées numériquement.

Ainsi pour pouvoir utiliser cette fonctionnalité avec notre carte Arduino, nous avons opté pour le module RFID RC522, du fait de son accessibilité rapide et de la carte et du bracelet 1k qui l'accompagnent pour son fonctionnement.

## i. Vue d'ensemble



Figure II-13 : Module RFID RC522 avec Carte et Bracelet Mifare Classic 1k [18]

## ii. Fiche Technique

- Basée sur une puce MFRC522 de NXP
- Tension d'alimentation : 3,3 V
- Courant : 13-26mA
- Fréquence d'utilisation : 13,56 MHz
- Distance opérationnelle : 0 ~ 60 mm
- Interface : SPI
- Dimensions : 40 mm × 60 mm
- Courant de travail : 13—26mA / DC 3.3V
- Courant de veille : 10-13mA / DC 3.3V
- Vitesse de communication des données : Maximum 10Mbit / s
- Cartes Supportées : Mifare1 S50, Mifare1 S70, Mifare UltraLight, Mifare Pro, Mifare Desfire
- Température de fonctionnement : -20 à 80 °C
- Humidité : humidité pertinente de 5% à 95%

### **iii. Le Fonctionnement**

Le module RC522, qui est un module qui permet la lecture et des moyens d'identification sans contact à savoir des cartes (badges) ou des clés RFID. Il est basé sur le circuit intégré Philips RC522 et communique avec Arduino via l'interface SPI. Il utilise la bande de fréquence ISM 13,56 MHz, la distance de communication peut atteindre 6 cm.

Il a aussi la possibilité de fonctionner avec les cartes de types NFC, vu qu'ils utilisent la même fréquence, mais ces derniers fonctionnent de façon optimale à une distance de 1 cm.

Le mécanisme du RC522 est basé sur des communications radio à courte portée utilisant les normes ECM-A340 et ISO / IEC 18092.

### **iv. La Carte MIFARE 1K**

Comme dit plus haut, la carte MIFARE 1K est celle qui a été utilisé pour notre projet, et bien entendu en plus des fonctions identifications de la carte, nous avons intégré une écriture de données qui permet de convertir cette donnée en valeur financière. Et cela nécessite de comprendre comment est structuré la mémoire de cette carte.

Comme son nom l'indique la MIFARE 1K, contient une mémoire de 1Ko qui est compartimenté de la façon suivante :

- 16 Secteurs (Numérotés de 0 à 15)
- Chaque Secteur est divisé en 4 Blocs (Numérotés de 0 à 3)
- Chaque Bloc qui contient 16 octets de données (Numérotés de 0 à 15)

Nous avons ainsi de façon résumée :

$$16 \text{ Secteurs} \times 4 \text{ Blocs} \times 16 \text{ octets de données} = 1024 \text{ o} = 1 \text{ Ko}$$



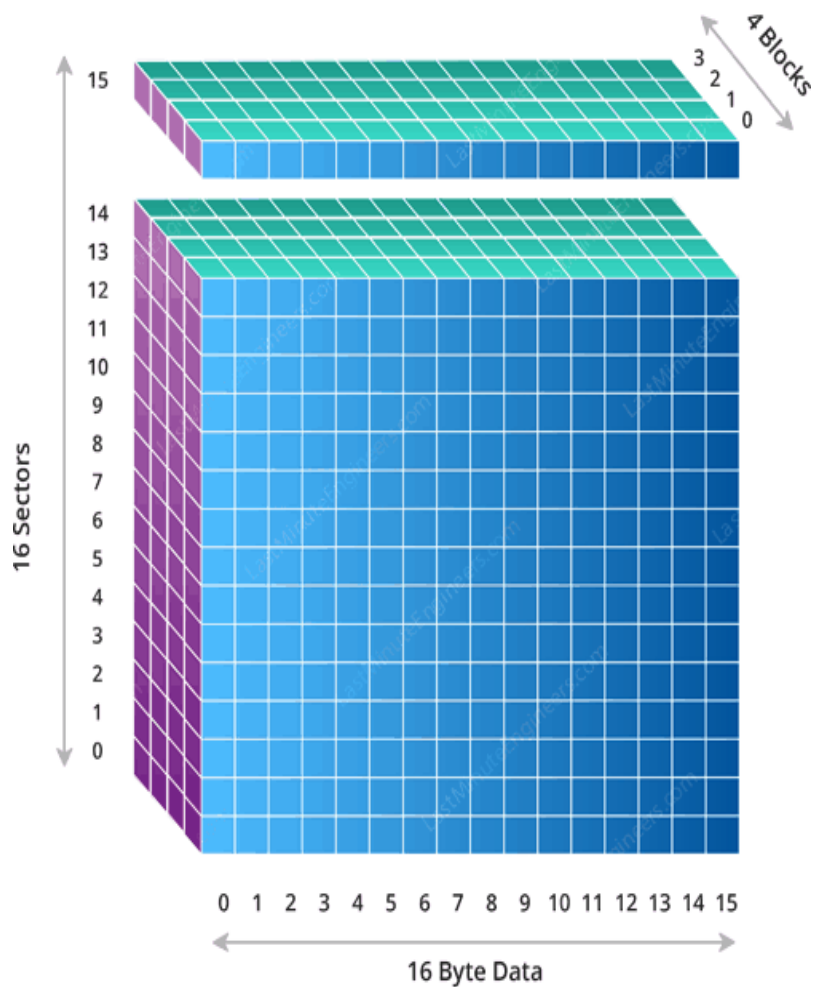


Figure II-14 : Représentation 3D de la mémoire de la MIFARE 1K [19]

- Le premier bloc du secteur 0 est appelé Bloc du Fabricant. Il contient des informations mises par le fabricant et contient le numéro d'identification unique de la carte (UID). Cet UID prend 4 octets de données.
- Le dernier bloc de chaque secteur est appelé « Sector Trailer », contient des informations appelées « Acces Bits » permettant d'accorder un accès en lecture et en écriture aux blocs restants du secteur. Cela implique que seuls les 3 derniers blocs (blocs 0, 1 et 2) de chaque secteur peuvent réellement être utilisés pour le stockage de données, ce qui signifie que 48 octets dans chaque secteur de 64 octets peuvent être réellement utilisés à nos propres fins.

## v. Le Branchement

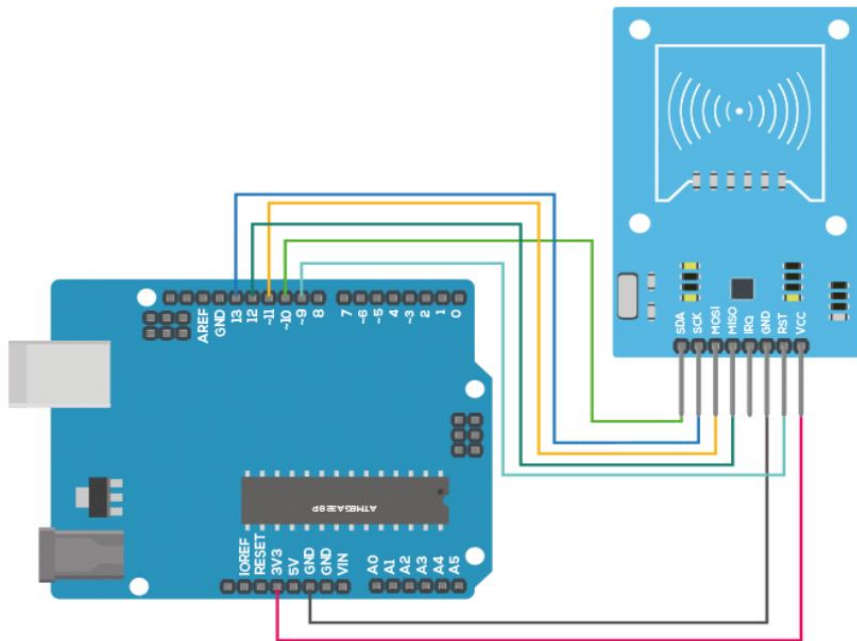


Figure II-14 : Branchement de la RC522 avec une carte Arduino Uno [20]

Le RC522 comporte diverses broches qui ne se connectent pas n'importe comment. Ainsi nous avons :

- VCC : qui se connecte au port 3.3 V de la carte Arduino et non la 5V pouvant causer une surtension de la carte et la griller.
- GND : qui se connecte au port de la masse de la carte Arduino GND.
- RST est une entrée pour la réinitialisation et la mise hors tension.
- IRQ : est une broche d'interruption qui alerte la carte Arduino lorsque la carte RFID arrive à proximité et il a la particularité de ne pas se connecter physiquement à la carte Arduino.
- MISO (Master In Slave Out) : agit comme un maître en sortie esclave lorsque l'interface SPI est activée.
- MOSI (Master Out Slave In) : est l'entrée SPI du module RC522.
- SCK (Serial Clock) : accepte les impulsions d'horloge provenant du bus SPI de la carte Arduino.
- SDA : agit comme signal d'entrée et aussi comme données série lorsque l'interface SPI est activée.

Mais à part les pins d'entrées GND ,3.3 V et RST (le RST qui se met dans n'importe quel port numérique) qui sont communes à toutes les cartes Arduino, le branchement des autres broches du RC522 diffère selon le type de cartes et suivant le plan fournit dans le tableau ci-après :

	MOSI	MISO	SCK	SDA
Arduino Uno	11	12	13	10
Arduino Nano	11	12	13	10
Arduino Mega	51	50	52	53

Tableau II-3 : Branchement des broches MOSI, MISO, SCK, SDA selon les cartes Arduino

#### 4.1.2.5. L'écran LCD 16×2

Pour afficher les informations de consommation du client, nous avons opté pour l'utilisation d'un écran LCD 16×2. Et notre écran pour des facilités de branchements utilisera un bus I2C.

##### i. Vue d'ensemble



Figure II-15 : Afficheur LCD 16×2 [21]

LCD (Liquid Cristal Display) ou Ecran à Cristaux Liquides en français est un afficheur qui utilise la propriété de modulation de lumière des cristaux liquides.

Ainsi les écrans à cristaux liquides sont composés de deux couches de polariseurs, avec des directions de polarisation perpendiculaire, prenant en sandwich deux plaques de verres entre lesquelles sont placés les cristaux liquides. Sur les plaques de verre se trouve une matrice d'électrodes pour chaque pixel. Une tension appliquée entre les électrodes d'un pixel entraîne

un changement d'orientation des molécules et donc la transparence du pixel qui peut alors laisser, ou non, passer la lumière du rétroéclairage.

Il est appelé LCD 16×2 du fait qu'il permet un affichage de 16 caractères sur 2 lignes.

## ii. Fiche Technique

- Alimentation : 5 V
- Interface I2C (adresse 0x27)
- Caractères blancs sur fond bleu
- Contraste ajustable via potentiomètre
- Dimensions : 80 x 38 x 18 mm

## iii. Le Branchement

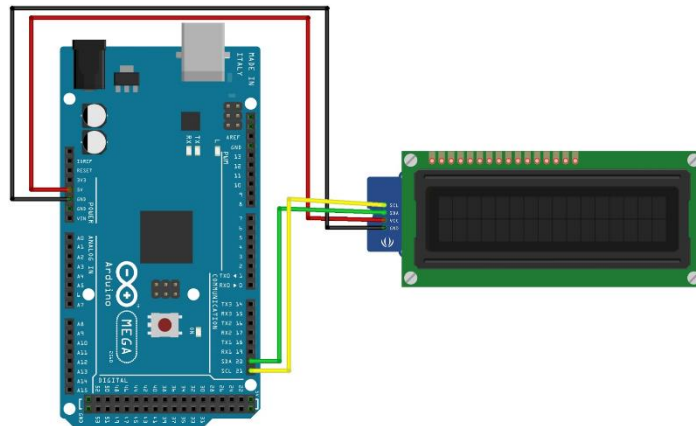


Figure II-16 : Branchement LCD 16×2 avec I2C sur Arduino Méga [22]

Pour brancher notre écran LCD à la carte Arduino, nous avons opté de le coupler à un bus I2C (Inter-Integrated Circuit) qui est un bus informatique réalisé par Philips pour les applications de domotique et d'électronique domestique permettant de relier facilement un microprocesseur et différents circuits tout en réduisant le nombre de lignes nécessaires à seulement deux lignes : SDA (Serial DATA) et SCL (Serial CLock). Il intègre un potentiomètre pour régler l'éclairage de l'écran.

Pour le branchement à la carte Arduino, nous avons :

- GND vers le pin GND.
- VCC vers le pin 5V.
- SDA vers le pin A4 de la Uno / le pin 20 sur la Mega
- SCL vers le pin A5 de la Uno /le pin 21 sur Mega

#### 4.1.2.6. L'électrovanne 12V/DC

Pour le système de coupure automatique des eaux, nous avons opté pour l'utilisation de deux électrovannes. Une qui sert à la fin de notre abonnement en eau et l'autre qui sert à la coupure à distance de la part du client en cas de souci par exemple.

Ainsi l'électrovanne encore appelée électrovalve est une vanne commandée de façon électronique et qui agira sur le débit de l'eau en sortie.

##### i. Vue d'ensemble



Figure II-17 : Modèle électrovanne 12V/DC [23]

##### ii. Fiche Technique

- Température de travail : 90 °C
- Diamètre nominal : 8mm
- Matériau d'étanchéité : caoutchouc

- Milieu de travail : eau du robinet, eau pure
- Pression minimale : 0.8 Mpa = 0.8 Bar
- Taille : 9 \* 6 \* 6 cm
- Poids : 110 g
- Tension de fonctionnement : 12 V

### iii. Fonctionnement

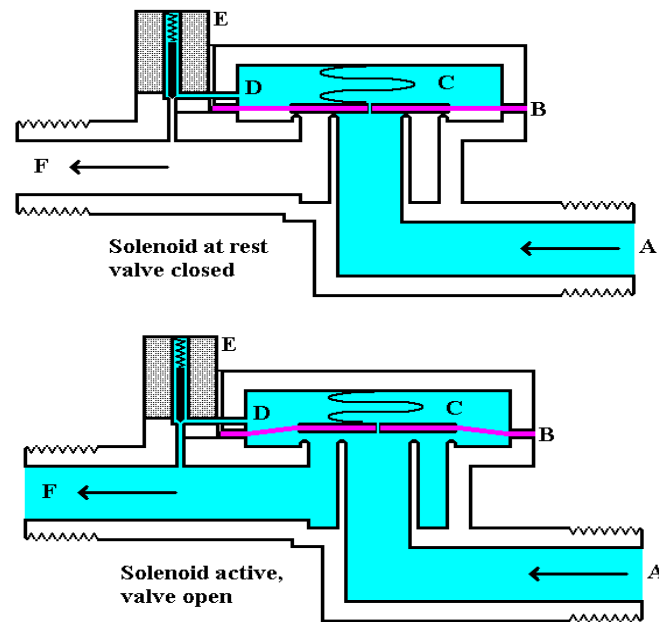


Figure II-18 : Fonctionnement d'une électrovanne [24]

Avec

- A- Arrivée de l'eau
- B- Diaphragme
- C- Chambre de pression
- D- Conduit
- E- Électro-aimant
- F- Sortie de l'eau

Ainsi pour fonctionner, l'électroaimant est excité par le courant électrique, créant un champ magnétique agissant sur la bobine solénoïde qui se trouve au-dessus du diaphragme. Cette bobine va agir sur le diaphragme qui va se soulever, permettant ainsi le passage de l'eau.

Et dès que le courant envoyé est arrêté, on se trouve en opération inverse, faisant baisser le diaphragme et ainsi empêcher l'eau de passer.

#### iv. Branchement

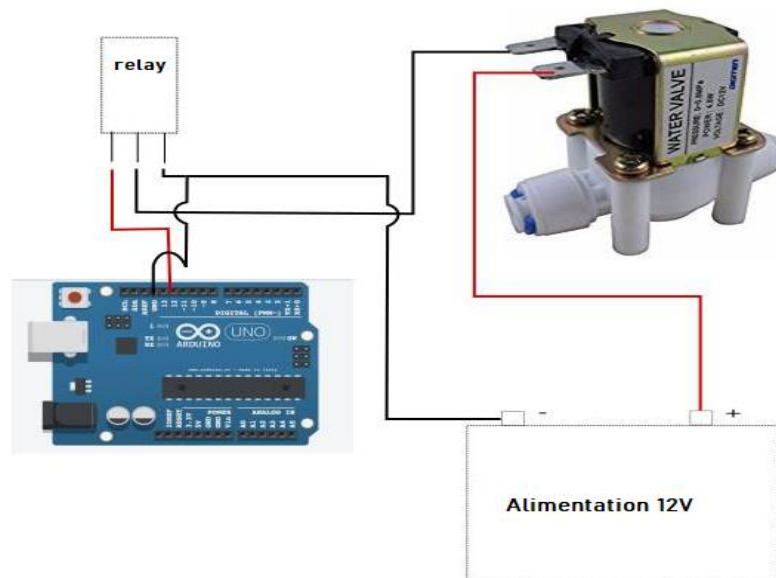


Figure II-19 : Branchement de l'électrovanne vers une carte Arduino

Du fait que les cartes Arduino quels que soit les modèles ne permettent que de commander que des éléments de 5V au maximum, il est nécessaire de passer par un relai. Ce relai permet de commander des composants qui demandent une tension supérieure à 5V.

Ainsi le relai agit comme un interrupteur qui va éteindre et allumer notre électrovanne selon nos envies. S'agissant du fonctionnement concret du relai nous le verrons dans la partie suivante qui lui est consacrée.

#### 4.1.2.7. Le Relay 2 Module

Comme dit plus haut la commande des modules demandant une tension d'alimentation de plus de 5V nécessite un relai, qui comme son nom l'indique sert de relai entre la carte Arduino et le module à commander. Ainsi elle permet de contrôler l'allumage d'une télé, d'une ampoule, d'un micro-onde... etc.

Mais pour notre système du fait de la présence de 2 électrovannes, nous avons opté pour un relai 2 Modules, qui permet de commander deux composants externes grâce à un seul relai.

## i. Vue d'ensemble



Figure II-20 : Relai 2 Modules [25]

## ii. Fiche Technique

- Sortie maximale du relai : DC 30V / 10A, AC 250V / 10A
- Dimensions : 50,6 x 38,8 x 19,3 mm
- Poids : 30g

## iii. Fonctionnement

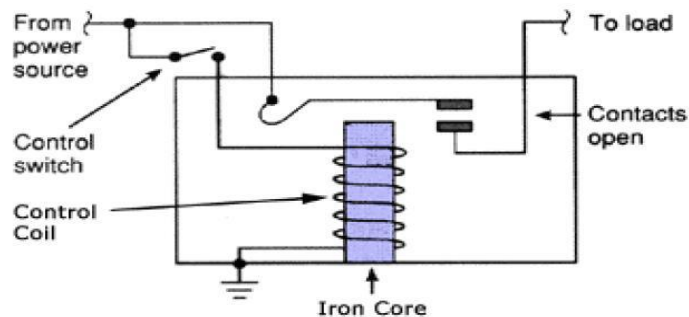


Figure II-21 : Coupe transversale interne d'un relai [25]

Déjà il faut savoir qu'un relai est divisé en 5 parties :

- Un Électroaimant : il se compose d'un noyau de fer enroulé par une bobine de fils. Lorsque l'électricité passe à travers, elle devient magnétique.
- Une Armature : la bande magnétique mobile est connue sous le nom d'armature. Lorsque le courant les traverse, la bobine est- énergisée produisant ainsi un champ magnétique qui est utilisé pour activer ou désactiver les points normalement ouverts (N/O) ou normalement fermés (N/C).



- Un ressort : lorsqu'aucun courant ne passe à travers la bobine de l'électroaimant, le ressort tire l'armature de sorte que le circuit ne peut pas être fermé.
- L'ensemble des contacts électriques, à savoir :

Normalement ouvert – connecté lorsque le relais est activé et déconnecté lorsqu'il est inactif.

Normalement fermer – pas connecté lorsque le relais est activé, et connecté quand il est inactif.

- Le cadre moulé : qui est le plastique de protection du relai.

Ainsi en se basant sur l'image de la coupe interne d'un relai. Le noyau de fer est entouré par la bobine de commande. Comme le montre la figure, l'alimentation est fournie à l'électroaimant via l'interrupteur de commande et les contacts de charge. Lorsque le courant commence à circuler dans la bobine de commande, l'électroaimant commence à s'activer, améliorant ainsi le champ magnétique. Par conséquent, le bras de contact supérieur commence à être attiré par le bras fixe inférieur, fermant ainsi les contacts, entraînant un court-circuit de la charge. En revanche, si le relais a été mis hors tension lors de la fermeture du contact, le contact se déplace en sens inverse et s'ouvre. Une fois le courant de bobine coupé, l'armature mobile reviendra à sa position initiale par la force.

#### **iv. Branchement et Commande**

Sur le relai, nous avons les broches suivantes qui se connectent comme suit :

- GND vers le pin GND de la carte Arduino
- VCC vers le pin 5V pour alimenter le relai
- IN1 vers un pin numérique, il sert à commander le premier objet qui est branché sur le relai.
- IN2 vers un pin numérique différent du premier, il sert à commander un second objet qui est branché sur le relai.

En ce qui concerne l'objet à commander nous aurons trois contacts électriques à savoir :

- Normalement fermé (NF)
- Normalement ouvert (NO)
- La Borne commune. (C)

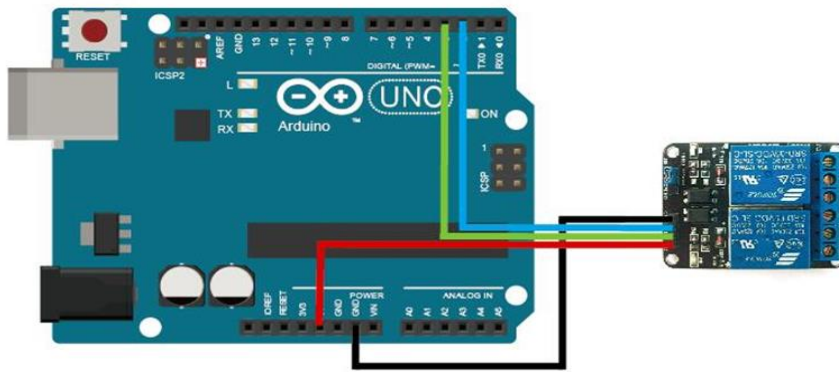


Figure II-22 : Branchement relai 2 modules vers carte Arduino UNO [25]

Selon le branchement nous aurons des résultats différents :

- Contact normalement ouvert :

Dans la configuration normalement ouverte, lorsque le relai reçoit la commande HIGH, le commutateur se ferme et permet au courant de circuler de la borne NO vers la borne C et là d'assurer la continuité du courant de NO à C. La commande LOW désactive le relai et empêche le courant de passer. Donc, si nous souhaitons allumer le relai avec la commande HIGH, il faut utiliser cette configuration.

- Contact normalement fermé :

Dans la configuration normalement fermée, la commande HIGH ouvre le commutateur et interrompt la sortie du courant vers l'objet. Par contre la commande LOW ferme l'interrupteur et permet au courant de circuler de la borne B vers la borne NF. Par conséquent, si nous souhaitons que la commande HIGH éteigne le courant de sortie, il faut utiliser cette configuration.



Figure II-22 : Relai - Contact Normalement Ouvert



Figure II-23 : Relai - Contact Normalement Fermé

#### 4.1.2.8. Le RTC DS3231

Pour toutes opérations nécessitant de se dérouler en temps réel, nous avons besoin d'un module qui va se charger de cette opération. Et ce module se nomme RTC (RTC pour Real Time Clock), ce qui donne en français Horloge Temps Réel. Une fois configuré il compte le temps qui passe et de façon autonome et précise grâce à sa pile s'il n'est pas branché.

Ainsi nous avons utilisé, le module RTC DS3231 qui est certes économique, mais qui fournit de très bons résultats.

##### i. Vue d'ensemble



Figure II-24 : RTC DS3231 de Keystudio [26]

##### ii. Fiche technique

- Plage de température : -40 à 85 °C ;
- Précision de synchronisation :  $\pm 5$  ppm ( $\pm 0,432$  seconde / jour)

- La fonction calendrier et l'horloge contient les secondes et les minutes, les heures, les semaines, la date, le mois et l'année et fournit une compensation pour les années bissextiles jusqu'en 2100.
- Sortie : 1 Hz et 32,768 kHz
- Réinitialiser la sortie et le rebond d'entrée du bouton-poussoir
- Haute vitesse (400 kHz), bus série I2C
- Tension d'alimentation : + 3,3 V à + 5,5 V

### iii. Branchement

Ce module fonctionnant avec un bus I2C le branchement se fera comme celui de notre écran LCD vers la carte Arduino. Donc nous avons :

- GND vers le pin GND.
- VCC vers le pin 5V.
- SDA vers le pin A4 de la Uno / le pin 20 sur la Mega
- SCL vers le pin A5 de la Uno /le pin 21 sur la Mega

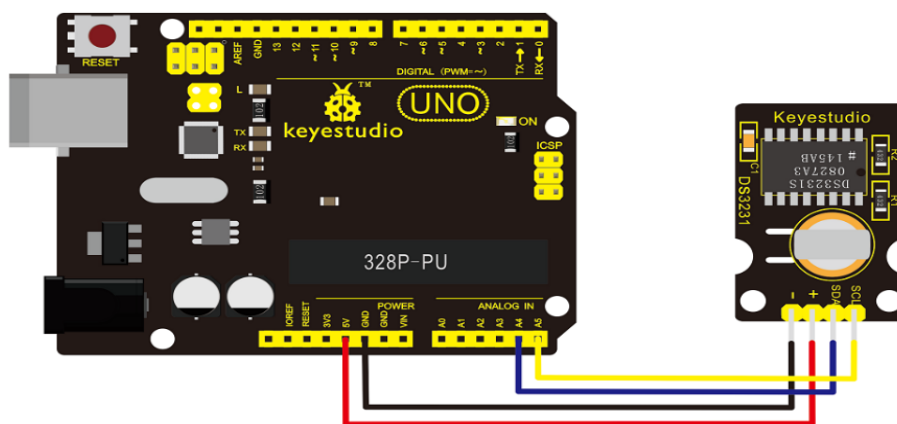


Figure II-25 : Branchement DS3231 vers carte Arduino [26]

#### 4.1.3. Etat prix du matériel

Après avoir présenté le choix du matériel, il est nécessaire de présenter un tableau récapitulatif du coût total des composants. Les prix affichés plus bas en Franc CFA, sont ceux obtenues à partir de notre fournisseur et donc peut varier selon le lieu où on souhaite l'acheter.

Et bien sûr ce bilan financier, ne tient pas compte des diverses dépenses liées à la main d'œuvre et autres dépenses.

MATERIELS	QUANTITES	PRIX UNITAIRE	PRIX TOTAL
CARTE ARDUINO MEGA	1	8000	8000
SIM808	1	25000	25000
DEBITMETRE YF-S201	1	5000	5000
RFID RC522+CARTE	1	3500	3500
LCD 16×2	1	4500	4500
RELAY	1	1000	1000
RTC DS3231	1	2000	2000
ELCTROVANNE 12V	1	7000	7000
LED	2	25	50
RESISTANCES	2	25	50
BOITIER	1	7000	7000
CHARGEUR 12V	1	6500	6500
CONNECTEURS PLOMBERIE	2	1000	2000
JUMPERS	2	1000	2000
		<b>TOTAL</b>	<b>73600</b>

Tableau II-4 : Prix du matériel

## 4.2. La partie Logicielle

Ce paragraphe décrit l'ensemble de la partie non physique de notre projet et comprend :

- Le Logiciel Arduino.
- La Base de Données.

### 4.2.1. Le logiciel Arduino

Pour faire fonctionner l'unité de traitement à savoir la carte Arduino, nous avons besoin de l'IDE qui permet de programmer la carte et donc faire fonctionner l'ensemble des composants du compteur. Et cet IDE est fourni par la société Arduino éditrice des cartes qui portent le même nom.

Ce logiciel permet d'écrire des programmes donc le langage de programmation est le C et le C++. Pour faire fonctionner le programme et donc par ricochet la carte et les composants, il est nécessaire de télécharger les bibliothèques de chaque composant et si possible d'avoir les dernières versions.

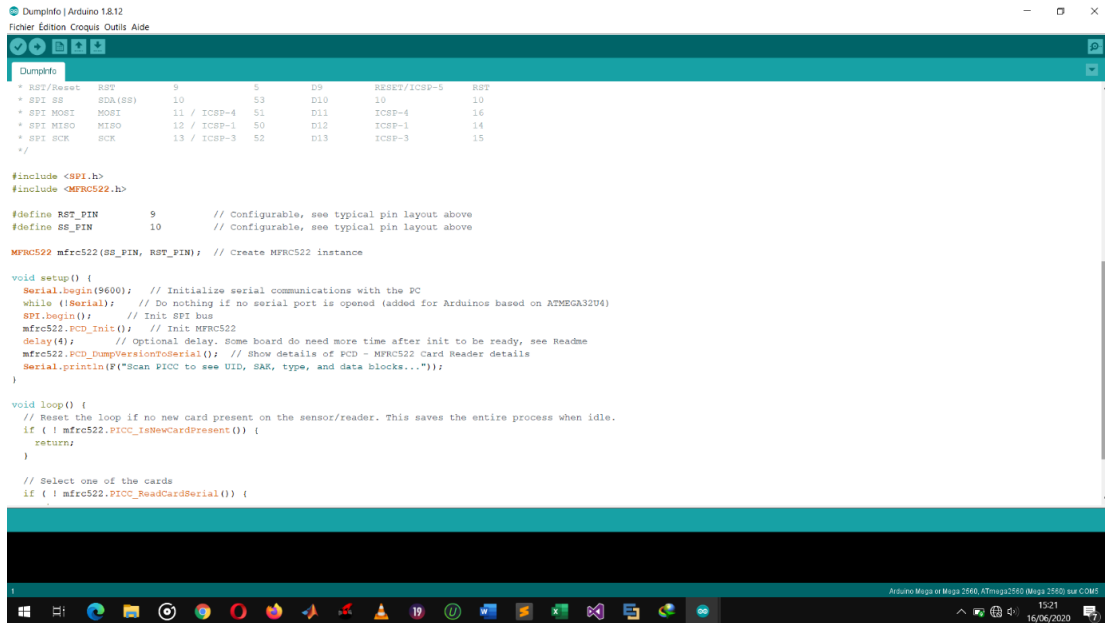


Figure II-26 : Interface du logiciel Arduino en version 1.8.12

## 4.2.2. La base de données

Pour la sauvegarde des données sur une base de données, il sera nécessaire d'utiliser un service de gestion de base de données. La base de données permettra de sauvegarder certaines données qui sont :

- L'état de consommation totale du client.
- L'état de consommation restante du volume d'eau acheté par le client
- L'état de consommation pour une période de 30 jours tous les mois à l'exception du mois du février qui compile les données sur 28 ou 29 jours.
- Le date et l'heure de l'activation effective de son compte d'eau au niveau du compteur.

Compte tenu des possibilités que nous offres le module SIM808, choisi pour notre réalisation d'une maquette, surtout avec sa fonction GPRS, nous avons opté pour le Service THINGSPEAK.

Concernant THINGSPEAK, il faut savoir que c'est une API et une application open source pour l'IOT (Internet Of Things) ou Internet des Objets en français, permettant de stocker et de collecter les données des objets connectés en passant par le protocole HTTP via Internet. Il est conçu par The MathWorks la structure derrière MATLAB.

Il offre de nombreux avantages à savoir entre autres :

- Une interface graphique permettant de visualiser les données envoyées
- Une intégration aux plates-formes Arduino, Raspberry Pi, aux applications mobiles/Web et aux réseaux sociaux.
- Analyser les données obtenues avec le logiciel MATLAB
- Exporter les données en format CSV, JSON pour une étude à travers un logiciel tableur comme Excel.

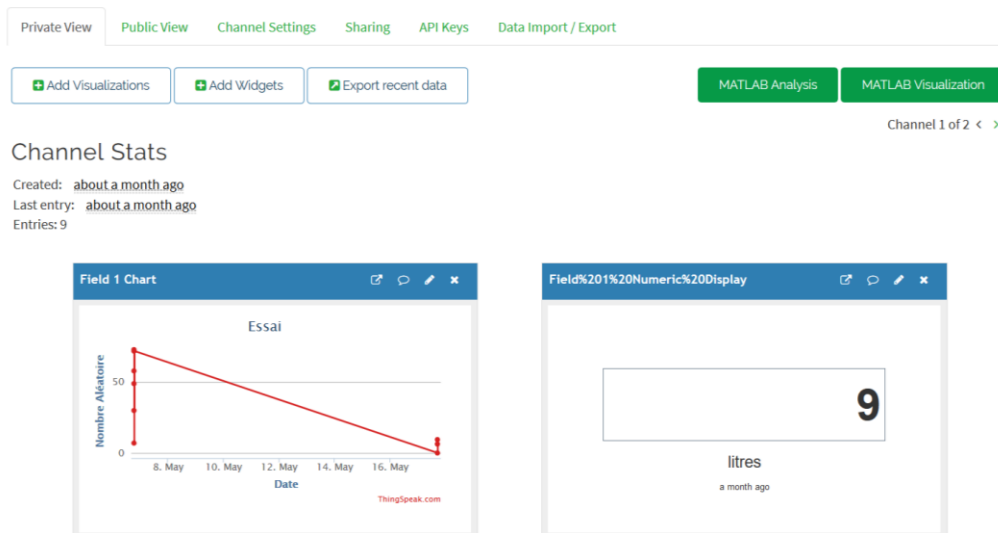


Figure II-27 : Interface Contrôle des données sur ThingSpeak

## **CHAPITRE III : IMPLEMENTATION**

Après la phase de conception qui a permis de présenter le fonctionnement de notre compteur et ses fonctionnalités, et du matériel nécessaire pour le réaliser, il est question dans ce chapitre de présenter la maquette physique et son fonctionnement. Ainsi d'expliquer une partie du code liée à certaines actions de ce compteur.

### **1. Le compteur**

#### **1.1. Le compteur en action**

Nous allons présenter, le comportement du compteur lors de certaines phases de son fonctionnement.

##### **1.1.1. La phase de démarrage.**

Lors de la phase de configuration des paramètres, le compteur affiche un message sur l'écran : « CONFIGURATION DU COMPTEUR », jusqu'à la fin de cette phase.



Figure III-1 : Phase configuration Compteur

##### **1.1.2. Compte en eau vide.**

Les électrovannes sont fermées et sur l'écran un message l'informant de ce vide et l'invitant à recharger est affiché sur l'écran et une LED rouge signalant que le compte est vide est allumé.





Figure III-2 : Compteur lorsque le compte est vide en volume d'eau

### 1.1.3. La phase de paiement

Nous avons deux comportements du système, en premier lorsqu'une carte non identifiée à celle du client du compteur est utilisée pour payer et le second cas celle de la carte du client.

Dans le premier cas, nous avons un message « IDENTIFIANT INCONNU » qui est affiché sur l'écran et dans le second cas, les messages relatifs à la recharge du compte sont affichés.



Figure III-3 : Mauvaise Carte Utilisée



Figure III-4 : Bonne Carte Utilisée

### 1.1.4. La phase de fonctionnement

Lorsque le compte est chargé en eau, nous avons sur l'écran deux messages affichant la consommation globale de tout le compteur et la quantité d'eau restant dans le compte. Et la LED verte est allumée pour signaler que le compte est bien actif.



Figure III-5 : Le compteur en fonctionnement normal

## 1.2. Les captures d'écran SMS

Lors du fonctionnement du compteur, le client reçoit certaines informations par le canal du SMS. Nous allons présenter certains messages.

### ➤ La phase de recharge du compte

Lors de la recharge du compte le client reçoit un message l'informant du volume d'eau qu'il vient de mettre dans son compte.

### ➤ Connaitre son état de consommation

Après avoir envoyé le message « Consommation » au numéro de la SIM de son compteur, il reçoit un message lui indiquant la quantité d'eau restant dans son compte.

### ➤ La phase de recharge du compte avant la fin de sa précédente recharge.

Après avoir rechargé son compte d'un volume d'eau de réserve avant la fin de sa précédente recharge le client reçoit un message l'informant du volume d'eau ajouté et du volume d'eau total maintenant présent dans son compte.

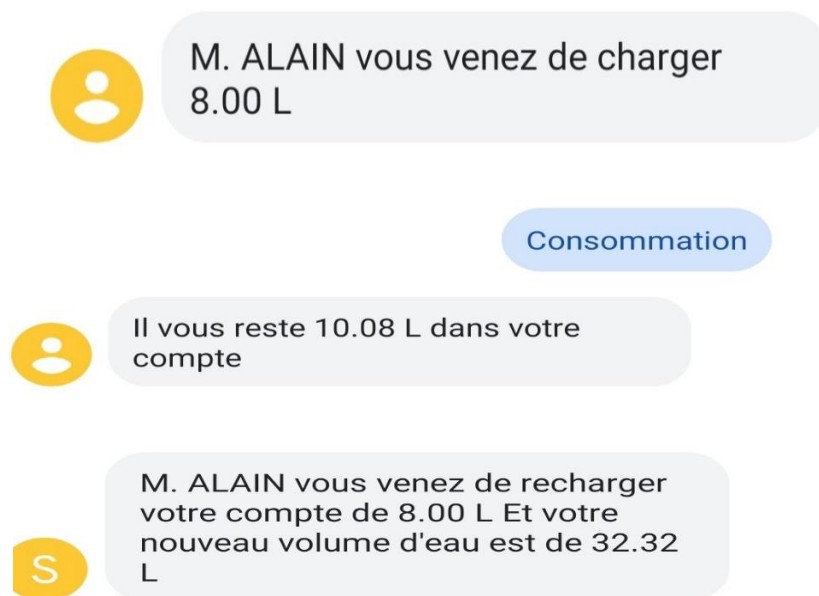


Figure III-6 : Captures d'écran des messages reçus par le client

## 2. Les codes de fonctionnement

Dans le cadre de la phase de réalisation de notre projet, il nous est apparu adéquat de présenter la programmation de certaines fonctionnalités phares de notre maquette, en présentant les parties du code qui permettent la réalisation de ces actions. C'est ainsi que notre choix s'est porté sur :

- La conversion de la somme d'argent de la carte RFID en volume d'eau.
- La réception par SMS du volume d'eau restant dans le compte du client.
- Le calcul du volume d'eau restant après consommation.
- L'affichage de l'état de consommations d'eau sur l'écran du compteur.
- La fermeture automatique de l'électrovanne à la fin de la consommation du volume d'eau acheté.
- L'envoi de l'état de consommation sur le serveur à une heure précise.

### 2.1. Conversion de la somme d'argent de la carte RFID en volume d'eau

Après la lecture du secteur de la carte RFID, contenant la somme payée par le client pour remplir son compteur d'un volume d'eau et qui est en byte, nous avons une conversion en

une variable myString de type String. Cette dernière variable subit une conversion en valeur numérique grâce à la fonction toFloat () et on en récupère la partie entière qu'on affecte à la variable somme en Int qui correspond à la variable qui va nous permettre d'obtenir le volume d'eau correspondant à la somme payée en agence.

```
//Recuperation de la somme et conversion en volume d'eau
String myString = String((char*)readbackblock); //conversion en string de la valeur ecrite sur la carte
int somme = myString.toFloat(); //conversion en valeur chifffé de la somme

-
VOLT = somme*5; //volume total acheté
restant = VOLT; //Recupération du volume d'eau restant dans le compteur
```

Figure III-7 : Extrait du code Relatif à la conversion de la somme d'argent de la carte RFID en Volume d'eau

Par la suite la variable somme subit une opération afin d'avoir le volume d'eau correspondant à l'argent, que l'on appelle VOLT. Et ce volume acheté est affecté à une dernière variable restante, qui va correspondre dans l'ensemble du système au volume d'eau restant à consommer par le client dans le compteur.

## 2.2. Calcul du volume d'eau restant après consommation

```
DEBIT = FREQUENCY/7.5;
LS = DEBIT/60;

TOTAL = TOTAL + LS;
VOL=VOL+LS;
restant = VOLT-TOTAL;
if (restantR!=0)
{
    restant = VOLT-TOTAL+restantR;
}
```

Figure III-8 : Extrait du code Relatif au Calcul du volume d'eau restant après consommation

Comme présenté dans le fonctionnement du débitmètre, on obtient le débit d'eau en divisant la fréquence d'impulsion par 7.5 et pour obtenir le volume d'eau on le divise par 60, correspondant respectivement aux variables DEBIT et LS dans le code.

Par la suite on crée deux variables VOL et TOTAL initialisées à des valeurs nulles, qui au moment de la première utilisation du compteur jusqu'à l'épuisement de l'eau restante consommées ont les mêmes valeurs et qui correspondent à la consommation totale du compteur depuis ses débuts et à la consommation du compteur pendant la présence d'un volume d'eau acheté restant et qui se réinitialise à la fin du volume d'eau restant.

Pour que le système connaisse la volume d'eau restant dans le compteur, on applique à la variable restant l'opération VOLT-TOTAL qui est le volume d'eau acheté moins le volume d'eau compté par le compteur. Cette opération s'effectue jusqu'à ce que la variable restante devienne nulle.

En cas d'un achat d'un volume d'eau de réserve à partir du volume d'eau seuil et noté restantR le volume d'eau restant devient VOLT-TOTAL+restantR, faisant que le volume d'eau restant dans le compte augmente directement.

### **2.3. Réception par SMS du volume d'eau restant dans le compte du client**

Pour connaître le volume d'eau restant dans le compteur du client, il faut que le client envoie le mot clé Consommation par SMS au numéro de la puce placé dans le module SIM808, et pour valider la réponse du système il faut qu'il y ait encore de l'eau disponible dans le système. Ainsi le système génère un message qui est du modèle : « Il vous reste (restant) L » et comme le message est de type String il faut convertir le volume d'eau en une variable de type String.

Par la suite ce message est envoyé au travers d'une fonction SendSMS dont le rôle est d'activer la commande AT+CMGS = \Numéro Téléphone Client relative à l'envoi d'un SMS au numéro de téléphone du client, par la suite le module SIM808 amorce l'envoi du message généré plus haut. Pour la suite, on applique le code ASCII 26, qui permet de mettre fin à la commande AT.

Et l'ensemble est bloqué dans un délai de temps pour envoyer le message au client.

```

if (textMessage.indexOf("Consommation") >= 0 && restant >0) { //message du sms pour activer le pin

textMessage = "";
String message = "Il vous reste " + String(restant) + (" L"); //le message renvoyé à la personne
sendSMS(message);

void sendSMS(String message) // Fonction pour envoyer les sms
{

GPSmodule.println("AT+CMGS=\""+237697540053\"");
delay(100);
// Send the SMS
GPSmodule.print(message);
delay(100);
Serial.println(message);

GPSmodule.println((char)26); // End AT command with a ^Z, ASCII code 26
delay(50);
GPSmodule.println();
// Give module time to send SMS
delay(5000);
}

```

Figure III-9 : Extrait du code relatif à la réception d'un SMS contenant le volume d'eau restant

## 2.4. Affichage de l'état de consommations d'eau sur l'écran du compteur

```

lcd.clear();
lcd.setCursor(0,0);
lcd.print("TOTAL:");
lcd.print( VOL);
lcd.print(" L");
lcd.setCursor(0,1);
lcd.print("RESTANT:");
lcd.print( restant);
lcd.print(" L");

```

Figure III-10 : Extrait du code relatif à l'affichage sur l'écran LCD des états de consommation

Pour afficher les états de consommation de l'eau à savoir le volume d'eau total consommé par le compteur et le volume d'eau restant dans le compte, on réinitialise l'écran LCD, on place sur la première ligne le texte fixe « Volume Global » suivi de la valeur correspondante. Et la seconde ligne commence avec le texte fixe « Volume Restant » suivi de la valeur correspondante.

## **2.5. Fermeture automatique de l'électrovanne à la fin de la consommation du volume d'eau acheté**

Dès que le volume d'eau restant dans le système est nul, le système entre temps donne la possibilité d'activer un nouveau volume d'eau par l'activation du lecteur RFID, et nous avons le relai qui est mis en position « LOW » enclenchant la fermeture de l'électrovanne.

```
if (restant<=0)
{
    rfid ();
    delay (1000);|
    pinMode (eVCompteur=46, LOW);
}
```

Figure III-11 : Extrait du code portant sur la fermeture automatique de l'électrovanne à la fin de la consommation du volume d'eau acheté

## **2.6. Envoi de l'état de consommation sur le serveur à une heure précise**

Pour envoyer les états de consommation en ligne, il faut d'abord l'heure à laquelle on doit envoyer les données et nous avons choisis de la faire toutes les heures à la minute zéro et cinquante-quatrième seconde. Ce choix s'est fait pour ne pas entrer en concurrence avec les autres fonctions qui utilisent l'heure pour fonctionner.

Par la suite on active la fonction sendserver qui comme son nom l'indique permet l'envoi des données sur le serveur, avec l'activation de la connexion internet entre le compteur et le serveur préalablement faite lors du paramétrage du compteur.

Ainsi donc la commande « AT+CIPSEND » est lancée pour préparer l'envoi des données par le protocole TCP, par la suite en fonction de la configuration de notre serveur à savoir Thingspeak, on met les variables VOLT et restant dans les champs 1 et 2 qui correspondent à

leurs positions respectives dans la table. Après l'envoi s'effectue. Ensuite s'applique la commande « AT+CIPSHUT » pour fermer la connexion.

```
if ((dt.Minute() == 00) && (dt.Second() == 54)) |
{
    sendserver ();
}

...
GPSmodule.println("AT+CIPSEND");//begin send data to remote server
delay(4000);
ShowSerialData();
String str="GET https://api.thingspeak.com/update?api_key=J7ZTXK96OUGL259J&field1=" + String(VOL)+"&field2=" +String(restant);
Serial.println(str);
GPSmodule.println(str);//begin send data to remote server
delay(4000);
ShowSerialData();
GPSmodule.println((char)26);//sending
delay(5000);//waitting for reply, important! the time is base on the condition of internet
GPSmodule.println();

ShowSerialData();

GPSmodule.println("AT+CIPSHUT");//close the connection
delay(100);
ShowSerialData();
```

Figure III-12 : Envoi de l'état de consommation sur le serveur à une heure précise

### 3. Le câblage

Certes le compteur fonctionne, mais il est nécessaire de présenter l'intérieur de notre compteur afin de voir comment tous les éléments mis bout à bout sont associés.

C'est ainsi que nous présentons le câblage à l'intérieur du boîtier en premier et en second, le schéma électronique.



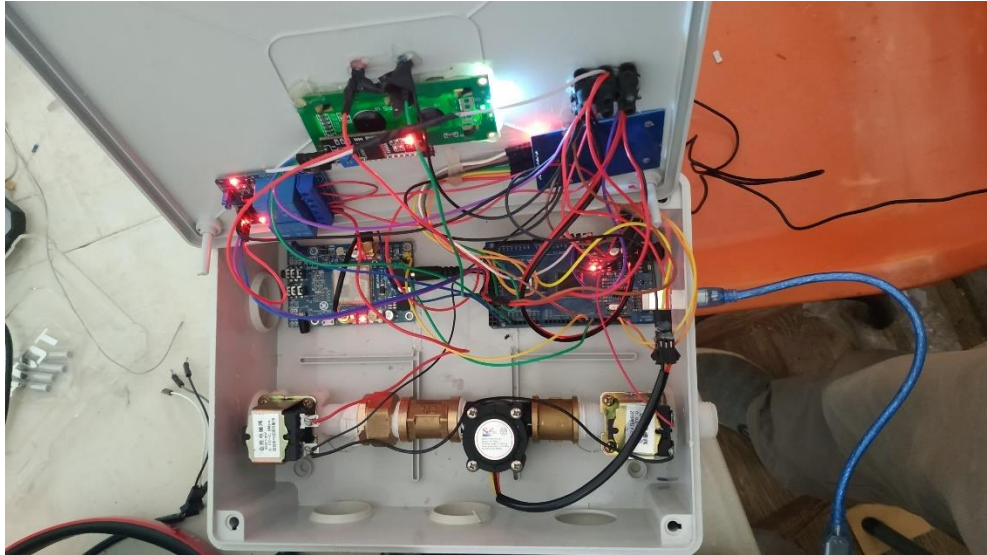


Figure III-13 : Câblage Interne

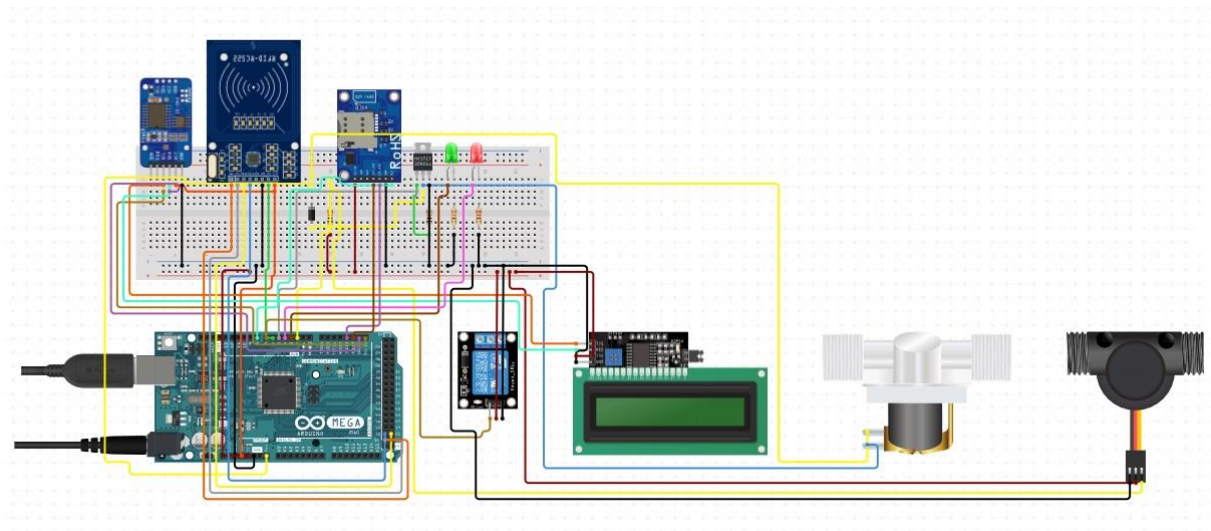


Figure III-14 : Schéma de câblage

## **CHAPITRE IV : SIMULATION ET RESULTATS**

Après avoir conçu et réaliser une maquette, il est question d'éprouver le fruit de notre recherche dans certaines conditions qui se rapprocherait d'une implémentation concrète de ce compteur. Les fonctions liées au compteur à savoir l'affichage des informations sur l'écran, le paiement via carte RFID, l'envoi des données de consommation sur un serveur ou au client via SMS, ne nécessitants pas un passage d'eau dans le compteur pour fonctionner. Nos tests se dérouleront dans des conditions qui vont simuler le raccordement du compteur dans un réseau de distribution des eaux. On simulera aussi le passage d'un fluide autre que l'eau à savoir l'air.

### **1. Passage de l'air**

Ce cas permet de voir comment se comporte le compteur, en cas de passage de l'air au lieu de l'eau comme cela arrive souvent avant le retour de l'eau dans les canalisations après une coupure d'eau.

#### **1.1. Condition du test**

Pour simuler l'envoi de l'air nous avons utilisé deux moyens à savoir : un souffleur portable envoyant un flux d'air de moyen de  $3 \text{ m}^3/\text{min}$  et un souffle venant de notre bouche avec lequel on enverra de l'air pendant cinq secondes. Les deux électrovannes activées nécessitent une pression minimale chacune de 0.8 Bar pour laisser passer un fluide. Et nous activons dans notre compte un volume d'eau de 30 Litres.

#### **1.2. Test et résultats**

##### **1.2.1. Souffleur portable**

Le souffleur utilisé à son maximum, nous observons que l'air envoyé à haute pression fait bien tourner notre compteur et au bout de 5 tests, nous observons que les 30 Litres d'eau achetés dans le compteur finissent entre 15 et 20 secondes.

### 1.2.2. Le souffle humain

Pour cette partie il faut noter que le débitage de notre air par le compteur, n'était pas parfait à tout moment, du fait que l'air envoyé ne traversait pas le compteur. Nous avons effectué cinq tests de cinq secondes en moyenne.

Test	Durée moyenne (Seconde)	Volume d'air mesuré (Litres)
1	5	0,08
2	5	1,05
3	5	0,09
4	5	0,72
5	5	0,54

Tableau IV – 1 : Volume d'air du souffle humain mesuré par le compteur

### 1.3. Analyses des résultats

Au vu de ses tests, nous constatons qu'un fluide comme l'air peut être bien comptabilisé par notre compteur et nous observons des valeurs un peu disparates, du fait de la non maîtrise de notre part du flux d'air envoyé et de la mesure du débitmètre qui a une précision de  $\pm 10\%$ .

Nous pouvons conclure, que notre système peut être bien utiliser dans la mesure de fluides comme le gaz et l'air. Et ceci augmentant bien les champs d'application de notre compteur.

Au vu du débitage du compteur en cas de passage de l'air dans certaines conditions, nous préconisons au l'ajout en amont de notre compteur, d'un capteur de niveau d'eau dont le rôle sera de s'assurer de la présence effective de l'eau dans les tuyaux d'eau. Ceci pour ne pas se retrouver à dans une situation dans laquelle, le compte en eau sera débité inutilement.

Ainsi en cas de présence d'eau il activera l'électrovanne du compteur pour laisser passer l'eau et en cas de passage d'air donc absence d'eau, il fermera automatiquement l'électrovanne.

Pour représenter, l'ajout de notre capteur en niveau d'eau nous proposons un synoptique plus bas présentant notre compteur et ce capteur ajouté.

#### ➤ Choix du capteur du niveau d'eau

Pour le choix de ce capteur, nous préconisons le Module capteur de niveau d'eau ST045. Ce choix peut se justifier par son coût pas onéreux et de sa petite taille.

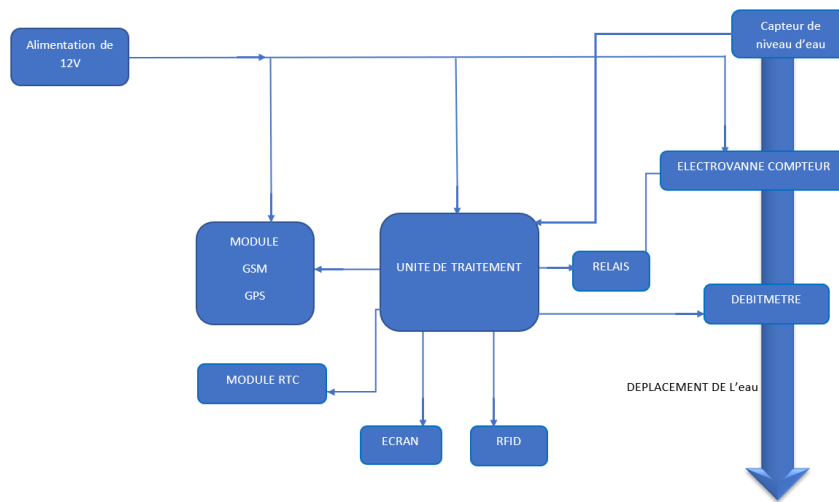


Figure IV-1 : Synoptique compteur avec un capteur d'eau en amont du système

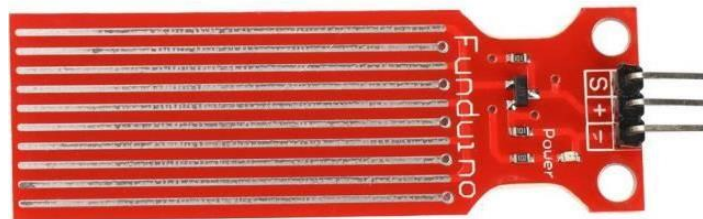


Figure IV-2 : Capteur de niveau d'eau ST045

## 2. Passage de l'eau

Ce cas permet de voir comment se comporte le compteur au moment du passage de l'eau dans le compteur

### 2.1. Condition du test

Du fait que notre maquette nécessite une source d'alimentation venant du courant électrique, nous n'avons pas pu raccorder notre compteur à la place d'un compteur CAMWATER. L'autre solution aurait été donc se raccorder à un robinet d'une maison, mais malheureusement à cause de la contrainte de notre maquette dû à son alimentation électrique n'étant pas autonome, cette solution n'est pas pratique vu que les positions des robinets d'eau dans les maisons ne sont pas proches des sources d'alimentation électriques. C'est ainsi que pour le passage de l'eau nous avons optés pour l'utilisation d'un seau d'eau avec un robinet intégré. Le seau que nous avons utilisé à un débit de 0,043 Litre/Seconde à son maximum, cette

mesure a été obtenue par une mesure préalable de notre part. Et le tuyau de raccordement mesure 16 mm de diamètre et 0,5 Mètre de longueur.

## 2.2. Test et résultats

Avec toutes les conditions citées plus haut, lors de l'alimentation en eau de notre compteur nous remarquons une non sortie de l'eau du système, malgré notre remplissage. Cette situation ne trouvant son explication par un manque de pression suffisante, nous nous proposons d'injecter une pression externe provenant de notre souffle. Et nous observons cette fois-ci un débitage effectif du compteur.

Du fait des conditions précédentes ne permettant pas d'avoir une très bonne mesure de l'eau au vu de la non maîtrise de la pression envoyée par nos soins, nous optons pour une autre approche à savoir raccorder une bouteille d'eau au tuyau d'entrée. Et cette approche nous permet d'avoir un débitage normal du compteur.

Nous avons effectué cinq tests en injectant 0,5 Litres d'eau à chaque fois et nous avons obtenu les résultats suivants :

Test	Volume d'eau mesuré (Litres)
1	0,45
2	0,49
3	0,46
4	0,42
5	0,41

Tableau IV – 2 : Volume d'eau mesuré

## 2.3. Analyses des résultats

Malgré les conditions pas idéales pour mettre à l'épreuve notre compteur, nous observons que le débitage de l'eau est bien faisable, mais nécessite une pression minimale de l'eau qui devra correspondre à celle fourni par les constructeurs du débitmètre et des électrovannes.

Et pour les mesures du volume d'eau, nous remarquons que la précision du débitmètre qui est de  $\pm 10\%$ , pourrait causer de légères pertes ou gains, au client ou à la compagnie des eaux.

Ainsi donc nous préconisons, un étalonnage strict du débitmètre au début de l'installation et aussi après la période de fonctionnement optimal du débitmètre.

Concernant le souci de pression, nous pouvons déduire que le choix de l'électrovanne est en ait la source principale, vu qu'il demande en entrée une pression minimale de 0,8 bar, ce qui est assez conséquent, donc l'idéal serait de se tourner vers une électrovanne dont la pression minimale de fonctionnement serait quasiment nulle comme pour le débitmètre YF-S201.

Pour le souci, observé pour le placement de notre compteur, nous préconisons le placement de notre compteur sous une toiture pour le protéger de l'environnement hostile de la pluie et du soleil. Si le branchement se fait effectivement à partir d'une alimentation électrique provenant de la compagnie d'électricité, il sera nécessaire de placer dans le compteur une batterie de secours, dont la durée sera fournie au client afin de lui permettre de faire des réserves d'eau par exemple et dans une meilleure situation permettre de faire le compteur pour une durée de 24 heures, selon les capacités de la batterie.

Ainsi de façon optimale, nous pouvons avoir le synoptique du compteur qui pourrait être le suivant :

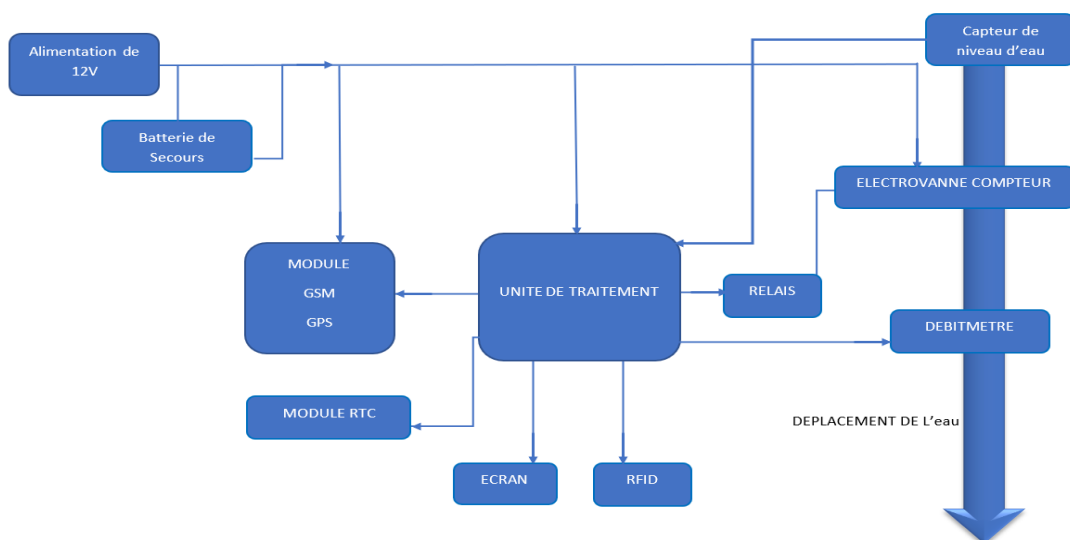


Figure IV-3 : Synoptique idéal du compteur actuel

## **CONCLUSION GENERALE**

Notre travail portait sur la conception et la réalisation d'un compteur d'eau intelligent prépayé. Ce mode prépayé est basé sur la technologie RFID. Ce projet venait répondre sur la problématique de rendre les compteurs d'eau automatiques et plus intuitifs.

C'est ainsi que nous avons commencé à faire une étude sur les divers travaux concernant les compteurs intelligents en ressortant certaines limites en tenant compte du contexte camerounais.

Par la suite, nous avons conceptualisé le fonctionnement et les outils qui allaient permettre de mener à bien notre projet. Il s'est avéré que notre projet tournait autour de la technologie Arduino et des différents composants compatibles avec cette technologie. Ce choix s'est quasiment imposé à nous du fait de la taille de notre projet qui ne devait produire qu'un exemplaire, ainsi que le coût et la facilité d'accès aux composants du compteur et la programmation qui va avec, afin de bien faire fonctionner l'ensemble. S'agissant l'opération de paiement nous avons choisi le module RFID RC522 et pour le débitage de l'eau le débitmètre YF-S201.

Après la réalisation, nous avons eu à faire des essais de simulation en condition réelle. Malgré des conditions d'expérience ne permettant de pas de juger de tout le potentiel optimal de notre compteur, nous pouvons en tirer qu'avec le choix de notre matériel pour bien fonctionner notre compte, nous aurons a besoin d'une pression de l'eau assez suffisante. Et ces essais ont permis de confirmer une application de ce compteur pour une autre industrie que celle de la distribution de l'eau tel que la distribution du gaz.

Après ces résultats, qui sont certes intéressants, il faut bien penser à des améliorations dans les travaux de recherches sur la conception et la réalisation de ce compteur. Ainsi les axes futurs pourraient être :

- L'ajout d'un système de mémoire interne qui pourrait combler une absence de connexion Internet sur le réseau ;
- Penser à un système d'alimentation qui permettrait de faire fonctionner le compteur même en absence de l'électricité, soit par la connexion de ce dernier à une source d'énergie renouvelable du type solaire ;

- Du fait de l'évolution du RFID vers le NFC, penser à développer une application capable de fonctionner avec les fonctionnalités NFC des smartphones, en profitant du développement du « Mobile Money » au Cameroun.



## **BIBLIOGRAPHIE**

- [2] Grégoire Wallenborn. (2013). Avis sur les compteurs communicants. Parlement Wallon.
- [3] Patil, S. (2019). Prepaid Energy Meter Using Arduino and Gsm Module. <https://doi.org/10.22214/ijraset.2019.3175>
- [4] padma gade. AN INTELLIGENT SMS BASED PREPAID WATER METERING SYSTEM <https://fr.slideshare.net/padmagade1/an-intelligent-sms-based-prepaid-water-metering-system>
- [5] Uzair Ahmed Rajput, & Khalid Rafique. (2018). Modeling of Arduino-based Prepaid Energy Meter using GSM Technology. (IJACSA) International Journal of Advanced Computer Science and Applications, 9(5), 445 449.
- [6] Annapurna Mishra, Avinash Kumar, Chetan Chaturvedi, Gautam Kamran, & Ravi Kumar. (2018). Prepaid Electricity Meter using GSM Module. International Journal Of Advance Research And development, 3(2), 240 244.
- [7] D. Harshitha Reddy, & P. Shilpa. (2012). Smart Prepaid Energy Meter using GSM and Arduino. International Journal of Management, Technology And Engineering, 8(12), 4400 4006.
- [8] B. O. Omijeh, G. I. Ighalo, & F. I. Anyasi. (2011). SMS- based Recharge Protocol for Prepaid Energy Billing System. International Journal of Engineering Innovation & Research, 1(6), 553 558.
- [9] NACK, NSALI, WAFO. (2018). CONCEPTION ET REALISATION D'UN COMPTEUR INTELLIGENT DE GESTION DE L'ENERGIE EN MODE PREPAYE [Memoire Master]. ENSET DOUALA.

## **WEBOGRAPHIE**

- [1] WILD, J. (2018, mai 31). Compteur communicant. Encyclopédie de l'énergie. <https://www.encyclopedie-energie.org/compteur-communicant/>
- [10] Arduino Mega : Utilisation et fonctionnement. (2019, février 18). Arduino France. <https://www.arduino-france.com/review/arduino-mega-utilisation-et-fonctionnement/>
- [11] Arduino Mega 2560 Rev3 | Arduino Official Store. (s. d.). Consulté 5 juin 2020, à l'adresse <https://store.arduino.cc/arduino-mega-2560-rev3>
- [12] Pinterest. (s. d.). Pinterest. Consulté 5 juin 2020, à l'adresse <https://www.pinterest.it/carlocraveri/schema-ramps-2560/>
- [13] Basics : Project 053d SIM808 GSM GPRS GPS Bluetooth evolution board (EVB-V3.2) at Acoptex.com / ACOPTEX.COM. (s. d.). Consulté 7 juin 2020, à l'adresse <http://acoptex.com/project/264/basics-project-053d-sim808-gsm-gprs-gps-bluetooth-evolution-board-evb-v32-at-acoptexcom/#sthash.bQe0reDh.uJz11DDU.dpbs>
- [14] Les commandes AT. (s. d.). Consulté 7 juin 2020, à l'adresse [https://www.technologuepro.com/gsm/commande\\_at.htm](https://www.technologuepro.com/gsm/commande_at.htm)
- [15] codebender\_ccFollow. (s. d.). How to Use Water Flow Sensor—Arduino Tutorial. Instructables. Consulté 8 juin 2020, à l'adresse <https://www.instructables.com/id/How-to-Use-Water-Flow-Sensor-Arduino-Tutorial/>
- [16] Water Flow Sensor -1/2" – ktechnics. (s. d.). Consulté 8 juin 2020, à l'adresse <https://ktechnics.com/shop/sensors/gas/water-flow-sensor-12/>
- [17] admin. (2017, novembre 21). Water Flow Sensor YF-S201 Arduino Interface. TheoryCIRCUIT - Do It Yourself Electronics Projects. <http://www.theorycircuit.com/water-flow-sensor-yf-s201-arduino-interface/>
- [18] SunFounder RFID Kit Mifare RC522 RFID Reader Module : Amazon.in : Electronics. (s. d.). Consulté 24 juin 2020, à l'adresse <https://www.amazon.in/SunFounder-Mifare-Reader-Arduino-Raspberry/dp/B07KGBJ9VG>
- [19] RC522 RFID Module Pinout, Features, Specs & How to Use It. (s. d.). Consulté 14 juin 2020, à l'adresse <https://components101.com/wireless/rc522-rfid-module>
- [20] Fabacks. (2017, septembre 25). Arduino : RFID-RC522. D@hoo. <https://dahoo.fr/wordpress/arduino-rfid-rc522/>
- [21] 16X2 LCD Display. (s. d.). PotentialLabs. Consulté 24 juin 2020, à l'adresse [https://potentiallabs.com/cart/lcd-16\\*2-india](https://potentiallabs.com/cart/lcd-16*2-india)
- [22] Mega\_2560\_lcd\_i2c\_bb.jpg (1600×1033). (s. d.). Consulté 24 juin 2020, à l'adresse [https://1.bp.blogspot.com/-WWuWQTjhiYs/WlnTNZ7XFal/AAAAAAAAA-NM/3t24q\\_uqrzYujKaA7H71NW-uoKT4qd8ogCLcBGAs/s1600/mega\\_2560\\_lcd\\_i2c\\_bb.jpg](https://1.bp.blogspot.com/-WWuWQTjhiYs/WlnTNZ7XFal/AAAAAAAAA-NM/3t24q_uqrzYujKaA7H71NW-uoKT4qd8ogCLcBGAs/s1600/mega_2560_lcd_i2c_bb.jpg)
- [23] US \$1.86 31% OFF|DC 12V Pressure Parallel Solenoid Valve Thread Normally Closed Solenoid Valve Connection Switch Normally Closed Solenoid Valve|Valve|—AliExpress.

(s. d.). aliexpress.com. Consulté 15 juin 2020, à l'adresse [//www.aliexpress.com/item/4000983942110.html?src=ibdm\\_d03p0558e02r02&sk=&aff\\_platform=&aff\\_trace\\_key=&af=&cv=&cn=&dp=](https://www.aliexpress.com/item/4000983942110.html?src=ibdm_d03p0558e02r02&sk=&aff_platform=&aff_trace_key=&af=&cv=&cn=&dp=)

[24] Électrovanne. (2020). In Wikipédia. <https://fr.wikipedia.org/w/index.php?title=%C3%89lectrovanne&oldid=168390139>

[25] Arduino lesson – 2-Channel Relay Module « osoyoo.com. (s. d.). Consulté 15 juin 2020, à l'adresse <https://osoyoo.com/2017/07/31/arduino-lesson-2-channel-relay-module/>

[26] Ks0039 keyestudio DS3231 Clock Module—Keyestudio Wiki. (s. d.). Consulté 15 juin 2020, à l'adresse [https://wiki.keyestudio.com/Ks0039\\_keyestudio\\_DS3231\\_Clock\\_Module](https://wiki.keyestudio.com/Ks0039_keyestudio_DS3231_Clock_Module)

# ANNEXES

## 1. Fiche Technique capteur de niveau d'eau ST045

- Nom du produit : capteur de niveau d'eau
- Tension de fonctionnement : DC3-5V
- Courant de fonctionnement : moins de 20mA
- Type de capteur : Analogique
- Zone de détection : 40mm x 16mm
- Production processus : FR4 double-face HASL
- Température de fonctionnement : 10 °-30 °
- Humidité : 10%-90% sans condensation
- Dimensions du produit : 62mm x 20mm x 8mm
- Erreur de mesure permise :  $\pm 1-3$  cm.
- Prix  $\approx 1000$  FCFA

## 2. CODE RFID

```
void rfid ()
{
  SPI.begin(); // Initiate SPI bus
  mfrc522.PCD_Init(); // Initiate MFRC522
  for (byte i = 0; i < 6; i++)
  {
    key.keyByte[i] = 0xFF;
  }
  lcd.clear();
  lcd.setCursor(3,0);
  lcd.print("COMPTE VIDE");
  lcd.setCursor(2,1);
  lcd.print("RECHARGER SVP");
}
```

```

if ( ! mfr522.PICC_IsNewCardPresent())
{
return;
}
// Select one of the cards
if ( ! mfr522.PICC_ReadCardSerial())
{
return;
}
String content= "";
byte letter;
for (byte i = 0; i < mfr522.uid.size; i++)
{
//Serial.print(mfr522.uid.uidByte[i] < 0x10 ? " 0" : " ");
//Serial.print(mfr522.uid.uidByte[i], HEX);
content.concat(String(mfr522.uid.uidByte[i] < 0x10 ? " 0" : " "));
content.concat(String(mfr522.uid.uidByte[i], HEX));
}
content.toUpperCase();
if (content.substring(1) == "FC DD 4E 43") //l'UID de la carte du client
{
readBlock(block, readbackblock);
lcd.clear();
lcd.setCursor(0,0);
lcd.print(" CREDIT CARTE ");
//Serial.print("Votre Compteur a été crédité de : ");
for (int j=0 ; j<16 ; j++)
{
Serial.write (readbackblock[j]);
}
}

```

```

//Recuperation de la somme et conversion en volume d'eau

String myString = String((char*)readbackblock); //conversion en string de la valeur ecrite sur
la carte

int somme = myString.toFloat(); //conversion en valeur chifffé de la somme

// writeBlock(block, blockcontent);

if (somme==0)
{
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("ATTENTION VOTRE"); //valeur de notre carte
  lcd.setCursor(0,1);
  lcd.print("CARTE EST VIDE"); //valeur de notre carte
  delay (2500);
}
else
{
  lcd.setCursor(4,1);
  lcd.print(somme); //valeur de notre carte
  lcd.print (" FCFA");
  delay (2500);

  /* if (somme == 2000){
    volT = somme*10; //volume total acheté
  }

  if (somme == 5000){
    volT = somme*15; //volume total acheté
  }

```

```

if (somme == 10000){
    volT = somme*25; //volume total acheté
}*/
lcd.clear();
lcd.setCursor(0,0);
lcd.print(" VOLUME CHARGE");
lcd.setCursor(6,1);
lcd.print(VOLT );
lcd.print(" L");

messagecharge ();
delay (2500);
sendserverRecharge () ;
}
}
else
{
lcd.clear();
lcd.setCursor(2,0);
lcd.print("IDENTIFIANT");
lcd.setCursor(5,1);
lcd.print("INCONNU");
delay(3000);
}
}

```

### 3. CODE DEBITAGE

```
void debitmetre ()
{
X = pulseIn(inputdebit, HIGH);
Y = pulseIn(inputdebit, LOW);
TIME = X + Y;
FREQUENCY = 1000000/TIME;
DEBIT = FREQUENCY/7.5;
LS = DEBIT/60;
if(FREQUENCY >= 0)
{
if(isinf(FREQUENCY))
{
lcd.clear();
lcd.setCursor(0,0);
lcd.print("VOL GBLE:");
lcd.print( VOL);
lcd.print(" L");
lcd.setCursor(0,1);
lcd.print("VOL RST :");
lcd.print( restant);
lcd.print(" L");

}
else
{
TOTAL = TOTAL + LS;
VOL=VOL+LS;
restant = VOLT-TOTAL;
if (restantR!=0)
```



```

{
    restant = VOLT-TOTAL+restantR;
}
lcd.clear();
lcd.setCursor(0,0);
lcd.print("VOL GBLE:");
lcd.print( VOL);
lcd.print(" L");
lcd.setCursor(0,1);
lcd.print("VOL RST :");
lcd.print( restant);
lcd.print(" L");
}
}
delay(1000);
if (restant<0)
{
    restant = (restant/restant)-1;
    TOTAL=(TOTAL/TOTAL)-1;
    restantR = (restantR/restantR)-1;
    VolReserve=(VolReserve/VolReserve)-1;
}
}

```