

République Algérienne Démocratique et Populaire
Ministère de L'Enseignement Supérieur et de la A Recherche Scientifique

UNIVERSITE MOULOUD MAMMERI DE TIZI-OUZOU



FACULTE DE GENIE ELECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'ELECTRONIQUE

**Mémoire de Fin d'Etude
de MASTER ACADEMIQUE**
Filière : **Génie électrique**
Spécialité : **Electronique Biomédicale**

Présenté par
Keltouma IZZA

Mémoire dirigé par **Mr H. ACHOUR**

Thème

**CONCEPTION ET REALISATION D'UN
DATA-LOGGER A 8 VOIES A BASE
D'UNE CARTE ARDUINO YUN**

Mémoire soutenu publiquement le 11/07/2016 devant le jury composé de :

Mr BENAMMANE KAMEL, MCA , Président
M^{elle} SIDI SAID AMEL , MCA, Examineur
Mr LAGHROUCHE MOURAD, Professeur, Examineur

Remerciements

La réalisation de ce mémoire a été possible grâce
au soutien de plusieurs personnes à qui je voudrais témoigner
toute ma reconnaissance.

Je voudrais tout d'abord adresser ma profonde et respectueuse
gratitude à mon promoteur, **Mr Achour**, pour sa patience, sa disponibilité
et surtout ses judicieux conseils durant
l'élaboration de mon projet.

Je désire aussi remercier tous mes anciens professeurs, qui m'ont fourni les
outils nécessaires à la réussite de mes études universitaires.

Je tiens à remercier Mr Nachef.M
pour son aide et le temps qu'il nous accordé.

Je voudrais exprimer ma reconnaissance envers les ami(e)s et ma
famille qui m'ont apporté leur support moral et intellectuel
tout au long de mon travail.

Enfin, je tiens à remercier les membres de jury qui ont
acceptés de juger et examiner mon travail.

Dédicace

Je dédie ce modeste travail consécration de
Mes études, et fruit d'un long travail
Aux êtres les plus chers de ma vie,
Mes très chers parents, qui ont tout fait pour
M'encourager durant toutes mes études, et c'est
Grâce à eux si je suis arrivé à ce que je suis aujourd'hui, qu'ils
Trouvent ici l'expression de mes sentiments de
Reconnaissance et de gratitude.
A mes très chers frères et beaux frères
A mes chères sœurs et belle sœur
A mes nièces et mes neveux
A toute personne qui était derrière ma réussite et
ma soutenait par tous les moyens

Sommaire

Liste des figures	
Liste des tableaux	
Liste des abréviations	
Introduction générale.....	1
Problématique.....	2
Objectif	3

Chapitre I : Acquisition des données

I. Système d'acquisition de données	4
I.1. Introduction	4
I.2. Définition	4
I.2.1. La chaîne d'acquisition de données	4
I.2.2. Schéma bloc d'une chaîne d'acquisition	5
I.2.3. Carte d'acquisition de donnée	6
I.2.3.1. Schéma bloc d'une carte d'acquisition	6
II. La conversion analogique numérique.....	6
II.1. Identification de la fonction C.A.N	6
II.2. Symbole d'un C.A.N.....	6
II.3. Principe.....	7
II.3.1. Echantillonnage.....	7
II.3.2. Bloqueur.....	8
II.3.3. Numérisation	9
II.3.3.1. Principe	9
II.3.3.2. Résolution	9
III. Le filtrage analogique	9
III.1. 1. Filtre actif et filtre passif.....	10
IV. Data-logger.....	11
IV.1. Introduction.....	11
IV.2. C'est quoi Data-logger.....	11
IV.3. Généralité sur les data-logger.....	12
IV.4. Différent type de data-logger	13
IV.6. Caractéristiques du data-logger	14

Chapitre II : Conception du data-logger

I. Introduction	16
II. Description générale du système réalisé.....	16
III. Description du système	17
III.1. Partie analogique.....	17
III.1.1. Electrodes.....	17
III.1.2. Filtrage du signal	20
III.1.3. Amplification.....	21
III.1.4. Alimentation de circuit	22
III.1.5. Circuit imprimé	23
III.2. Partie numérique.....	24
III.2.1. Présentation de la carte Arduino	24
III.2.2. Applications.....	24
III.2.3. Description de la carte Arduino Yún.....	25
III.2.4. Caractéristiques des deux processeurs	28
III.2.5. Le pont « Bridge »	29
III.2.5.1. Bibliothèques de bridge	30
III.2.6. Alimentation de la carte Arduino Yún.....	31
III.2.7. Réseau et configuration du WiFi	31
III.2.7. 1. Connexion Arduino Yun sur Wifi.....	31
III.2.7. 2.Connexion au réseau Wifi.....	34
III.2.8. La carte SD Arduino.....	35
III.2.8.1. Formatage et préparation de la carte.....	35
III.9. L'environnement de programmation Arduino.....	36
III.9.1. Description du logiciel Arduino.....	37
III.9.2. Description de la barre des boutons.....	38
III.9.3. Description de la barre des menus.....	38
IV- Organigramme	40

Chapitre III : Réalisation et résultats expérimentaux

I. Introduction.....	42
II. simulation et test de la partie analogique.....	42
II.1. Test avec le logiciel de simulation proteus et lab d'essai.....	42
III. Circuit imprimé	44
IV. Nomenclature de la carte électrique.....	46
V. Mise en boîtier du data-logger.....	47
VI. Logiciel de gestion.....	48
VII. Test du système et résultats obtenus.....	49

Liste des figures

Figure I.1 : Structure d'une chaine d'acquisition.....	5
Figure I.2 : Structure d'une carte d'acquisition.....	6
Figure I.3 : Symbole de la fonction Conversion Analogique / Numérique.....	6
Figure I.4 : Principe du chaîne de conversion analogique-numérique.....	7
Figure I.5 : Exemple de signal échantillonné.....	8
Figure I.6 : Structure de fonction de filtre.....	9
Figure I.7 : 2a (filtre passif) et 2b (filtre actif)	10
Figure I.8 : Filtre passif.....	10
Figure I.9 : data-logger.....	12
Figure I.10 : Différent type de data-logger.....	13
Figure II.1 : schéma synoptique de la chaine d'acquisition.....	16
Figure II.2 : Cellule électrochimique.....	17
Figure II.3 : Electrode de travail.....	18
Figure II.4 : Electrode de référence	19
Figure II.5 : Filtre passe bas.....	20
Figure II.6 : Amplificateur avec offset.....	21
Figure II.7 : Tension de référence.....	22
Figure II.8 : Schéma électrique de l'alimentation.....	23
Figure II.9 : le circuit analogique du système.....	23
Figure II.10 : Carte Arduino Yún.....	25
Figure II.11 : Carte Arduino vue avant et arrière	26
Figure II.12 : les différentes leds de la carte Arduino.....	26
Figure II.13 : les trois boutons de réinitialisation de la carte Arduino.....	27
Figure II.14 : Pont de communication entre les deux microprocesseurs	30
Figure II.15 : Arduino Yún sur Wifi.....	32
Figure II.16 : Plateforme d'accès a Arduino Yún	32
Figure II.17 : Interface Arduino après connexion au Wifi arduino Yún.....	33
Figure II.18 : Icône de configuration de Arduino Yún.....	33
Figure II.19 : Connexion de la carte au réseau Wifi.....	34
Figure II.20 : Fenêtre principale l'environnement de développement Arduino.....	36
Figure II.21 : Les différentes parties de la fenêtre principale du logiciel Arduino.....	37
Figure II.22 : accès au moniteur série.....	38
Figure II.23 : Fenêtre associée au moniteur série.....	38
Figure II.24 : Structure de la barre des boutons.....	39
Figure II.25 : Structure de la barre des menus.....	39
Figure II.26 : Organigramme de fonctionnement du système d'acquisition.....	40
Figure III.1 : Résultat obtenu avec la simulation sous proteus.....	42
Figure III.2 : Circuit électronique sur lab d'essai	43
Figure III.3 : Signal obtenu à la sortie du l'amplificateur	43
Figure III.4 : Circuit imprimé du système électronique.....	44

Figure III.5: Schéma d'implantation des composants de la carte électronique.....	44
Figure III.6 : Schéma du circuit imprimé avec l'implantation des composants sur la carte électronique.....	45
Figure III.7 : Vue 3D de la carte électronique	45
Figure III.8 : Carte électronique du data-logger.....	47
Figure III.9 : Face avant du boîtier.....	47
Figure III.10 : Face arrière du boîtier.....	48
Figure III.11 : Test du système d'acquisition réalisé.....	49
Figure III.12 : Résultats obtenus.....	50

Liste des tableaux

Tableau II.1 : Les caractéristiques du microcontrôleur ATmega32U4.....	28
Tableau II.2 : Les caractéristiques Atheros AR9331.....	29
Tableau III.3 : Nomenclature de la carte électrique	46

Liste des abréviations

SAD : système d'acquisition des données

CNA : convertisseur analogique numérique

ADC : analogic digital conversion

SD: secure digital

AP: point d'accès

SPI: serial peripheral interface

CDC: communication device class

PoE: power over Ethernet

Introduction générale

Depuis que l'électronique existe, sa croissance est fulgurante et continue encore aujourd'hui. Si bien que faire de l'électronique est devenu accessible à toutes personnes le désirant.

Le savant D.Mendeleev a écrit « La science commence là où commence la mesure » ; cela signifie que nous n'avons pas de science sans mesure. La mesure est un processus de connaissance qui grâce à l'expérience physique nous donne une information quantitative du rapport entre la grandeur mesurable et une grandeur de même nature prise comme référence.

Il existe plusieurs appareillages dédiés à la mesure et le stockage de données, ils diffèrent selon le domaine d'utilisation. Parmi eux, on retrouve les data-loggers, qui permettent d'acquérir et d'enregistrer des paramètres physiques ou électriques sur des phénomènes à analyser, que l'on souhaite vérifier ou dont on veut connaître les lois de variation.

Dans le cadre de ce travail de fin d'étude il nous été demandé de concevoir un datalogger qui est un mélange d'électronique et d'informatique. L'électronique pour l'instrumentation et l'informatique pour le stockage numérique des mesures effectuées et la communication avec d'autres systèmes informatiques.

Aussi, avons nous organisé notre travail en trois parties :

Dans le premier chapitre on va énoncer quelque généralité, sur l'acquisition de données et les data-loggers.

Puis, le second chapitre, abordera la conception du système proposé. Une étude des différents étages et une présentation de la carte Arduino Yun autour de laquelle s'architecture notre système y seront détaillées.

Le dernier chapitre quant à lui, portera sur la réalisation du système, les différents tests ainsi que les résultats obtenus.

Problématique

La plupart des métaux ne sont thermodynamiquement pas stables aussi bien dans l'atmosphère terrestre qu'enfouis dans le sol. Ils subissent une oxydation naturelle, plus ou moins rapide, leur surface étant alors en constante évolution. Cependant, des couches d'oxyde sont formées à la surface des objets pour les protéger d'une dégradation rapide en favorisant la croissance d'une couche protectrice.

La mesure de la corrosion est un processus important qui peut aider un électro-chimiste à construire des matériaux bien enrobés qui peuvent résister à l'environnement corrosif, comme elle peut l'aider à déterminer les caractéristiques et tirer des conclusions qui aideront par conséquent le fabricant à améliorer la qualité du revêtement et réduire le risque de corrosion.

L'objectif du Laboratoire d'Electrochimie et Matériaux de Tizi-Ouzou est de faire une Synthèse chimique et électrochimique et analyse des propriétés physico chimiques de nouveaux matériaux moléculaires et composites obtenus à partir de polymère organique conducteur conjugué et d'oxyde comme le bioxyde de manganèse, ou de semi conducteurs ou photoconducteurs, ou à partir de matériaux moléculaires dans lesquelles sont incorporés des métaux de transition.

Le laboratoire s'intéresse aussi aux problèmes de corrosion et de la protection des métaux contre la corrosion comme la protection cathodique ou l'utilisation des inhibiteurs. Cette mesure de corrosion peut être effectuée par des chaines ou appareillages de mesures électroniques assez onéreux.

Vu le manque d'appareillage au niveau du laboratoire qui puisse faire le suivi du potentiel d'un ensemble d'échantillons immergées dans un électrolyte (milieu corrosif) pour en étudier le comportement sur une longue durée (jusqu'à plusieurs mois), il nous a été proposé de concevoir un tel système de mesure autonome multi voies.

Objectifs

L'objectif de la réalisation du data-logger est de permettre aux utilisateurs du laboratoire d'électrochimie de l'université Mouloud Mammeri de Tizi-Ouzou de suivre l'évolution du potentiel en fonction du temps et par rapport à une référence d'un ensemble d'échantillons à analyser. Ces mesures permettent d'évaluer la capacité anti-corrosive du matériau d'électrode utilisé. L'idéal est de réaliser une mesure simultanée (minimum 8 voies à la fois) du potentiel pour plusieurs électrodes immergées dans un électrolyte (une solution quelconque, généralement une solution corrosive (solution de chlorure de sodium (NaCl) par exemple).

Pour cela, un système d'acquisition de données est nécessaire afin de permettre de traiter les résultats obtenus pour chacune de ces électrodes (différentes voies). Concernant les critères de mesures, elles peuvent être imposées par l'utilisateur, c.à.d. un critère de temps (mesure chaque intervalle de temps donné (10s) par exemple), ou bien un critère de potentiel (mesure pour chaque variation de potentiel de 5 mV par exemple).

I. Système d'acquisition de donnée [1]

I.1. Introduction

Nous présenterons dans ce chapitre quelques principes généraux qui permettraient de mettre en œuvre des phénomènes intervenant dans l'acquisition de données,

I.2. Définition

Le système d'acquisition des données (S.A.D) est un ensemble d'éléments matériel et logiciel destiné à recueillir des données physiques par l'intermédiaire de capteurs.

I.2.1. La chaîne d'acquisition de données

La chaîne d'acquisition de données est l'ensemble des éléments nécessaires à la "capture" des données (analogiques ou numériques) à leur transmission jusqu'au récepteur et à l'utilisateur (homme ou machine) des données capturées. Cet utilisateur peut vouloir utiliser ces données immédiatement ou les stocker pour les utiliser ultérieurement.

La chaîne d'acquisition généralement, elle est constituée de :

-) Une source d'information
-) Les capteurs capables de transformer l'information physiologique en une grandeur électrique.
-) Les circuits de mise en forme réalisant les fonctions d'amplification, conditionneurs, de filtrage, de calibration, multiplexage
-) La carte d'acquisition des données proprement dite permettant l'interfaçage homme machine.
-) transformation des données CAN
-) Un support logiciel réalisant l'acquisition, l'affichage, le traitement et la transmission des données conformément à un protocole de communication.

I.2.2. Schéma bloc d'une chaîne d'acquisition

Une chaîne d'acquisition peut se représenter selon le Schéma bloc suivant :

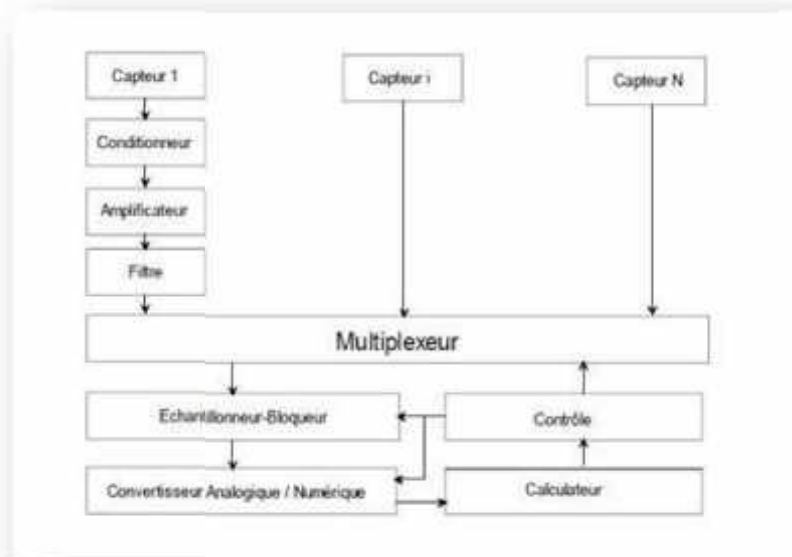


Figure I.1 : Structure d'une chaîne d'acquisition

I.2.3. Carte d'acquisition de donnée

Ensemble de circuits électroniques ou interface d'entrée/sortie, destiné à traiter l'information venant des capteurs: échantillonnage, digitalisation, etc.

I.2.3.1. Schéma bloc d'une carte d'acquisition

Une carte d'acquisition numérique peut se représenter selon la figure suivante :

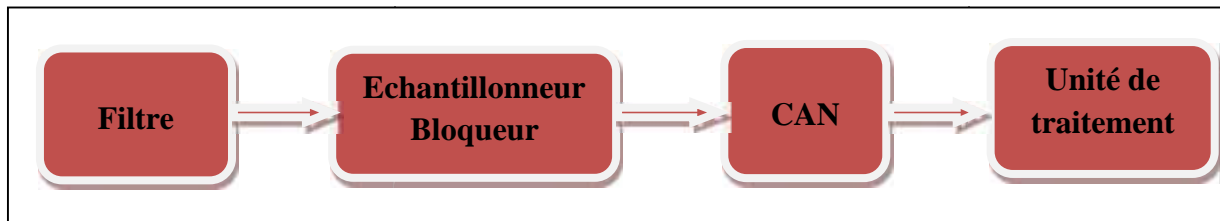


Figure I.2 : Structure d'une carte d'acquisition

I. La conversion analogique numérique

II.1. Identification de la fonction C.A.N

On appelle Convertisseur Analogique Numérique (C.A.N.) tout dispositif électronique qui transforme une grandeur analogique d'entrée U_e en un nombre binaire de sortie N proportionnel à cette grandeur u_e . En anglais, le Convertisseur Analogique Numérique est appelé Analogic Digital Conversion (A.D.C.)

II.2. Symbole d'un C.A.N

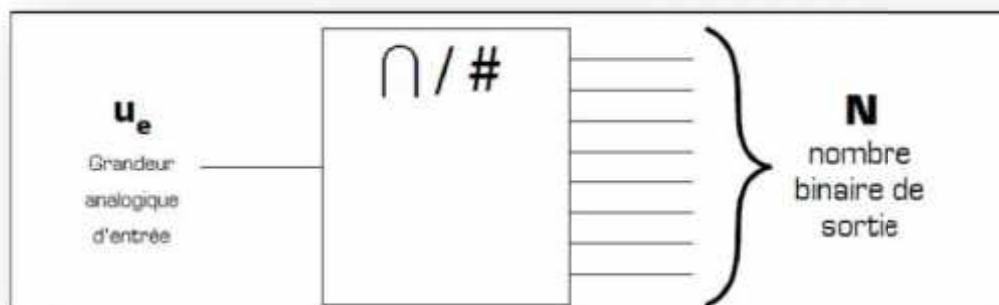


Figure I.3 : Symbole de la fonction Conversion Analogique / Numérique

Le signe n indique que la grandeur est de type analogique (il s'agit ici de l'entrée).
Le signe $\#$ indique que la grandeur est de type numérique (il s'agit ici de la sortie).

II.3. Principe

L'obtention d'un signal numérique à partir d'un signal analogique (exemple : microphone) nécessite de mettre en œuvre une chaîne de conversion analogique-numérique. Le convertisseur analogique-numérique a pour fonction de faire correspondre à un signal analogique un signal numérique de sortie.

Cette chaîne comporte typiquement la structure fonctionnelle suivante :

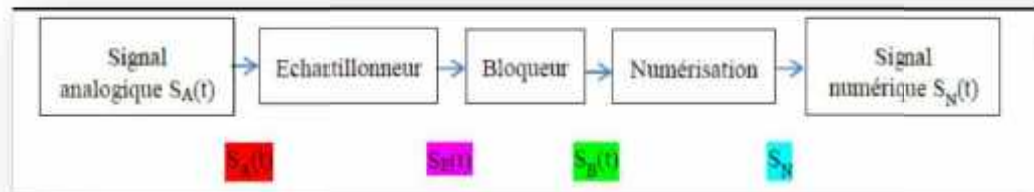


Figure I.4 : Principe du chaîne de conversion analogique-numérique

Les fonctions réalisées par chaque élément sont les suivantes :

- ❖ Echantillonneur : Acquérir la grandeur analogique à un instant t ;
- ❖ Bloqueur : Maintenir cette grandeur pendant la durée de la numérisation ;
- ❖ Numérisation : Convertir le signal analogique bloqué en un nombre (sur N bits).

II.3.1. Echantillonnage

L'échantillonneur a pour rôle de prélever périodiquement la valeur du signal analogique. Il réalise une « photographie » du signal qui « fige » son image jusqu'à la « prise » suivante. L'intervalle de temps entre 2 instants consécutifs d'échantillonnage constitue la période d'échantillonnage, notée T_E , du système de conversion analogique numérique.

Le choix de cette période est imposée à la fois par la nature du signal que l'on veut numériser mais aussi par le temps de met le convertisseur analogique numérique pour réaliser sa conversion. Usuellement, on utilise la notion de fréquence d'échantillonnage telle que

$$F_E = \frac{1}{T_E}$$

Un choix judicieux de cette fréquence est imposé par l'application du théorème de Shannon. Celui-ci précise qu'elle doit être au moins égale au double de la plus grande des fréquences du signal analogique.

Le signal échantillonné, à l'instant t (qui est un multiple entier de la période d'échantillonnage), est noté de la façon suivante que : $S_E(t) = S_E(nT_E) = S_A(t)$

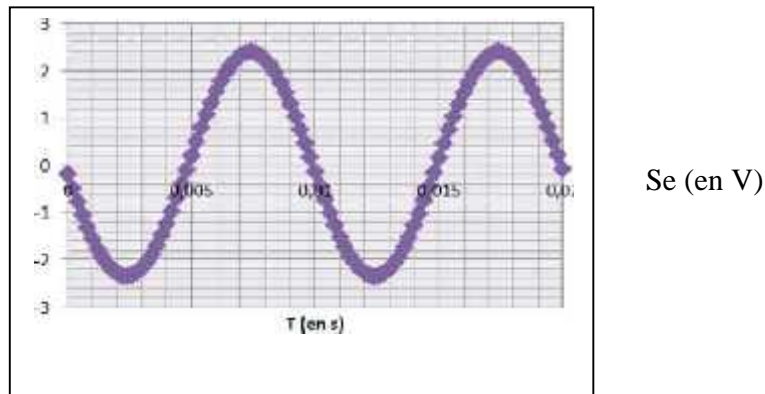


Figure I.5 : Exemple de signal échantillonné

La figure ci-dessus représente l'allure du signal échantillonné avec une période d'échantillonnage de 200 μ s. A chaque instant (multiple de 200 μ s), une valeur du signal analogique est acquise (comme une photo).

D'une manière générale, un signal véhiculant une information contient un certain nombre de fréquences correspondant à son spectre. Par nature, pour chaque fréquence, l'amplitude est aléatoire. Il est donc nécessaire de considérer une bande de fréquence dans laquelle se trouvent de façon pertinente les informations relatives au signal. On fixe ainsi une fréquence maximale du signal analogique et par conséquent on accepte de « perdre » une partie de l'information.

Ce choix doit être pertinent en trouvant un compromis entre rapidité de conversion et qualité de numérisation.

II.3.2. Bloqueur

Le bloqueur a pour fonction de maintenir constant le signal échantillonné $SE(t)$ afin de permettre au convertisseur analogique numérique de le numériser. Ce signal est maintenu constant à la valeur $SB(t)$ jusqu'à l'échantillonnage de la valeur suivante.

Actuellement les bloqueurs agissent comme une mémoire qui garde constante la valeur échantillonnée ($SB(t)=\text{constante}$), on parle alors de bloqueur d'ordre 0. On trouve aussi des bloqueurs d'ordre supérieur (1, 2..), lorsque l'évolution du signal $SB(t)$ est assimilable à des décroissances exponentielles ou paraboliques.

II.3.3. Numérisation

II.3.3.1. Principe

La numérisation est la transformation d'un signal analogique issu du bloqueur en un signal numérique. Elle est réalisée par un convertisseur analogie numérique (en anglais : ADC) qui convertit le signal bloqué en un nombre. La sortie du convertisseur est raccordée ensuite à un système numérique (ordinateur, analyseur de spectre numérique.)

Cette numérisation réalise la quantification du signal analogique bloqué en lui associant un nombre. Ce nombre est compris entre 0 et 2^N-1 valeurs possibles où N est le nombre de bits en sortie du convertisseur. Ainsi pour 8 bits, on a 2^8 valeurs différentes possibles en sortie soit 256 valeurs différentes. Les nombres associés vont alors de la valeur 0 jusqu'à 255.

II.3.3.2. Résolution

La résolution du convertisseur, notée r, est l'intervalle de tension analogique permettant de passer du nombre n à (n+1). Pour une tension analogique d'entrée allant de 0 V à une amplitude V_{MAX} . Si cette tension d'entrée varie de 0 à 10 V et que l'on utilise un convertisseur analogique numérique de 8 bits, on a une résolution de 39,2 mV. En d'autres termes, pour que la valeur numérique en sortie du convertisseur varie, il faut la tension d'entrée ait varié de 39,2 mV.

I. Le filtrage analogique

D'une manière générale, un filtre est un système linéaire dont le rôle est de modifier la composition spectrale d'un signal sans y ajouter de nouvelles composantes. Il permet le renforcement ou l'atténuation d'une ou plusieurs bandes de fréquences. Ainsi il élimine les parasites.

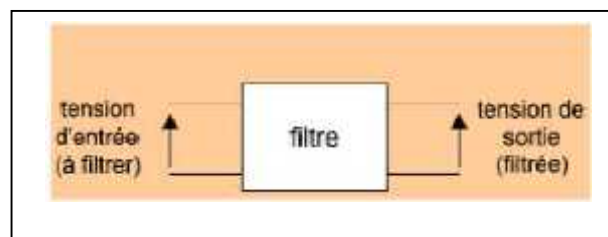


Figure I.6 : Structure de fonction de filtre

Un filtre est un circuit dont le comportement dépend de la fréquence.

Si la tension d'entrée est sinusoïdale alors la tension de sortie est sinusoïdale et de même fréquence.

Les différents filtres sont :

- a. Filtre passe-bas
- b. Filtre passe-haut
- c. Filtre coupe-bande
- d. Filtre passe-bande

III.1. Filtre actif et filtre passif

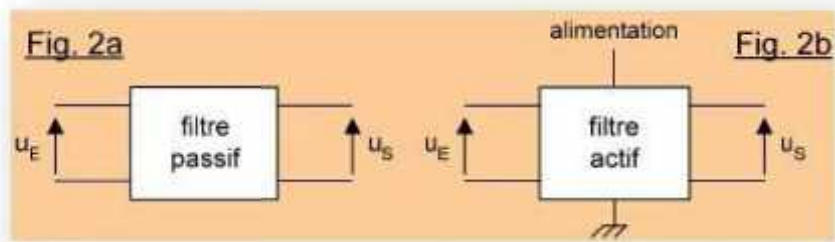


Figure I.7 : 2a (filtre passif) et 2b (filtre actif)

- **filtre passif** : on y trouve résistances, bobines et condensateurs.



Figure I.8 : Filtre passif

- **filtre actif** : alimentation externe, transistors, A.O.

I. Les Data-loggers

IV.1. Introduction

Trois types d'instruments sont couramment utilisés pour la collecte et le stockage de données, ce sont :

- Les Real-Time Data Acquisition Systems,
- Les Enregistreurs Graphiques
- Et les data-loggers.

Parmi ces trois types d'appareillages c'est les data-loggers qui offrent plus de souplesse et sont disponibles avec une plus grande variété de types d'entrée. La plupart des data-loggers recueillent des données qui peuvent être transférées directement à un ordinateur. Bien que cette option soit disponible avec certains enregistreurs.

Les data-loggers s'utilisent dans tous les domaines dans lesquelles il faut réaliser des prises de longue durée d'un paramètre de mesure sans avoir besoin d'une présence continue de personnel de contrôle sur le lieu de la mesure .

Le data-logger est placé, la plupart du temps, dans un environnement où l'homme a besoin de suivre l'évolution dans le temps de grandeurs physiques et n'a pas la possibilité d'être présent sur place, comme un lieu difficile d'accès de type : pont, ouvrage d'art, objet en mouvement, sites dangereux ... ou à cause de temps de mesure important : plusieurs heures, jours voir mois.

Ainsi les data-loggers sont utilisés surtout pour des tests sur le terrain, le contrôle de transports (par exemple transports en camions), les tests dans l'industrie alimentaire, l'analyse d'erreurs de systèmes (par exemple), aussi dans le domaine biomédicale à titre d'exemple data-logger de température permet la surveillance de la température des produits pharmaceutiques dans les réfrigérateurs, ECG data-loggers (holter) sont portables utilisés par les cardiologues, pour surveiller en continu l'activité du système cardio - vasculaire

IV.2. C'est quoi un Data-Logger ?

Le data-logger est composé des mots "logger" qui est un enregistreur et "data" qui correspond à donnée. C'est un ensemble électronique et informatique, d'éléments permettant d'acquérir et d'enregistrer des paramètres physiques ou électriques sur des phénomènes à analyser, que l'on souhaite vérifier ou dont on veut connaître les lois de variation, pour pouvoir les améliorer, les comprendre etc... et ce à des intervalles déterminés sur une période de temps donnée.

Le data-logger décrit est en réalité composé d'une ou plusieurs entrées capteurs et d'une unité de traitement et de mémorisation. Pour assurer une grande autonomie, les data-loggers font appel à des circuits intégrés consommant très peu d'énergie. Ils permettent la mesure de différentes grandeurs physiques (Tension, température, courant, pression, force, accélération..).



Figure I.9: data-logger [3]

IV.3. Généralité sur les data-loggers

Le data-logger est un instrument électronique qui sert à enregistrer des mesures à des intervalles déterminés sur une période de temps donnée.

Les data-loggers se composent d'un processeur programmable, de ports et d'un ou plusieurs canaux d'entrée pour la connexion de différents capteurs et d'une mémoire pour le stockage de données.

Ces dispositifs peuvent être soit des unités portables autonomes avec des capteurs internes ou des stations de collecte de données multi-canal avec plusieurs capteurs externes. Les data-loggers peuvent être utilisés pour diverses applications, que ce soit en extérieur, intérieur ou sous l'eau. En d'autres termes, les data-loggers sont utilisées partout où des données sont susceptible d'être mesurées.

Cependant, la plupart des fabricants d'instruments considèrent un data-logger comme un dispositif autonome qui peut lire différents types de signaux électriques et les stocker dans la mémoire interne pour les télécharger vers un ordinateur. [4]

L'avantage des enregistreurs de données (data-logger) est qu'ils peuvent fonctionner indépendamment d'un ordinateur. Lors de l'activation, les enregistreurs de données sont généralement déployés et laissés sans surveillance pour mesurer et enregistrer des informations pour la durée de la période de surveillance. Cela permet une image complète et précise des conditions d'environnement à surveiller, comme la température de l'air et de l'humidité relative, contrairement à de nombreux autres types de dispositifs d'acquisition de données. Enregistreurs de données sont disponibles en différentes formes et tailles. La gamme comprend des simples enregistreurs de fonction économique fixe à un seul canal à des dispositifs programmables plus puissants capables de traiter des centaines d'entrées. [4]

IV.4. Différent type de data-logger [2]

Il existe plusieurs types de data-loggers sur le marché, mais les trois principaux types sont:

- Les enregistreurs de données Bluetooth.
 - Les systèmes d'exploitation basés sur le Web.
 - Enregistreurs de données autonomes.
- **Bluetooth Data Logger** : Le data-logger Bluetooth est un enregistreur simple qui a l'avantage de la connectivité sans fil. Cet enregistreur est capable de mesurer et transmettre la température et d'autres données sans fil à des ordinateurs ou des appareils mobiles dans un rayon de 30 mètres.
 - **Web-Based System Logging** : les systèmes d'exploitation basés sur le Web permettent de donner l'accès à distance en utilisant l'Internet, le Wi-Fi ou le GSM.
 - **Stand-Alone Data-Logger** : ce type de data-logger est portable, simple, petit et réutilisable. Les data-logger qui utilisent des capteurs internes surveillent directement sur le lieu enregistreur, tandis que ceux qui utilisent des capteurs externes surveillent à distance de l'enregistreur de données. Les données d'enregistrement sont alors transmises à un ordinateur par l'intermédiaire d'une interface USB.



Figure I.10 : Différent type de data-logger [3]

IV.6. Caractéristiques du data-logger [4]

Lors du choix d'un data-logger les paramètres suivants doivent être pris en considération.

) **Signal d'entrée**

Certains data-loggers sont dédiés à un certain type d'entrée tandis que d'autres sont programmables pour des différents types d'entrées.

) **Nombre d'entrées**

Les data-loggers sont disponibles en simples ou multi-canaux (jusqu'à 32 canaux). Certains data-loggers sont capables de traiter des centaines d'entrées, bien que ceux-ci aient tendance à être volumineux et coûteux. Les systèmes d'acquisition de données qui enregistrent les données dans la mémoire du PC hôte en générale sont plus populaires que les data-logger autonomes dans le cas des systèmes à nombre de canaux élevé.

) **Taille**

Avec beaucoup d'appareils, l'espace est une limitation. Dans ces cas, la taille du data logger peut être un paramètre de sélection essentiel.

) **Vitesse / Mémoire**

En comparaison avec les systèmes d'acquisition de données en temps réel, les enregistreurs de données ont généralement des taux d'échantillonnage faibles, ceci est normal parce qu'ils stockent les données dans des mémoires internes limitées et aussi parce que les phénomènes physiques à mesurer ont des fréquences de variation faible.

Lors du dimensionnement d'un data-logger, il est important de déterminer la fréquence d'échantillonnage et la durée d'acquisition de l'échantillon qui peut être utilisé pour le calcul de la mémoire requise.

Par exemple, si une application nécessite des taux d'échantillonnage de 1 échantillon par seconde et que le test a une durée d'une heure, le data-logger doit être capable de stocker 3600 échantillons (1 échantillon / sec x 1 heure x 3600 secondes / heure).

) **Fonctionnement en temps réel**

Certains data-loggers ont la fonctionnalité d'afficher les données collectées en temps réel sur un ordinateur.

) **Période d'enregistrement**

La durée d'enregistrement dépend de la capacité de mémoire du data-logger et le taux d'échantillonnage souhaité.

Pour déterminer la durée maximum d'enregistrement, il faut diviser la capacité mémoire (nombre d'échantillons) par le taux d'échantillonnage. A titre d'exemple, supposons qu'un data-logger donné peut stocker 10.000 échantillons, si l'on souhaite enregistrer 2 échantillons à chaque minute, l'enregistreur de données peut fonctionner pour $10.000 / 2$ ou 5.000 minutes

(environ 3,5 jours). Si le taux d'échantillonnage a été réduit de moitié (1 échantillon par minute), la période d'enregistrement doublerait à 7 jours.

I. Introduction

L'objectif du data-logger à réaliser est de permettre de suivre l'évolution du potentiel de plusieurs échantillons par rapport à une référence en fonction du temps. L'idéal est de réaliser une mesure simultanée (minimum 8 voies à la fois) du potentiel pour plusieurs électrodes immergées dans un électrolyte, généralement corrosive (solution de chlorure de sodium (NaCl) par exemple).

Un système d'acquisition de données est nécessaire afin de permettre le traitement des résultats (données) obtenus pour chacune de ces électrodes (différentes voies). Concernant les critères de mesures, elles peuvent être imposées selon les besoins de l'utilisateur, c.à.d. un critère de temps (mesure à chaque intervalle de temps donné (toutes les secondes par exemple), ou bien un critère de potentiel (mesure à chaque variation de potentiel de 5 mV par exemple).

Ce travail est centré autour d'une petite carte appelé Arduino Yun, une plate-forme de prototypage open source largement distribué, peu couteuse et facile à utiliser avec tout un environnement de développement gratuit.

Ce chapitre est consacré à la description des différents blocs composant le système avec leurs schémas électriques que nous allons réaliser.

II. Description générale du système

Le système que nous proposons de réaliser est constitué de trois parties principales (acquisition du signal, conditionnement du signal, et récupération des données via une mémoire SD, le réseau Ethernet ou Wifi), comme illustré dans la figure suivante.

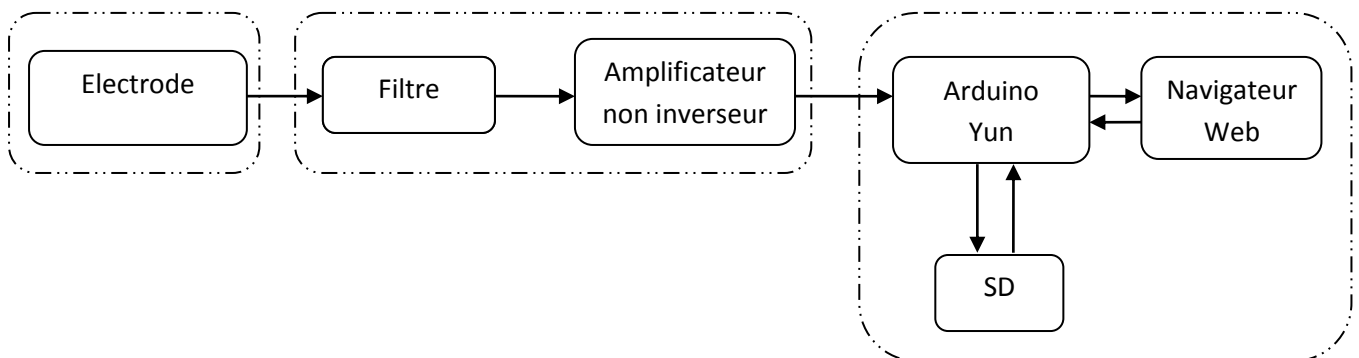


Figure II.1 : Schéma synoptique du data-logger

III. Description du système

Pour l'unité de traitement et de communication, nous avons opté pour une carte Arduino Yun, offrant des entrées analogiques et des moyens de communication tels l'Ethernet et le Wifi en plus d'un lecteur de carte mémoire micro SD pouvant servir pour le stockage des données.

III.1. Partie analogique

Le signal à acquérir entre un échantillon et la référence est une tension susceptible de varier entre +/- 2.5 V. Comme les entrées analogiques de la carte Arduino permettent l'acquisition de signaux entre 0 et 5 volts. On introduit donc un offset de 2.5 V afin d'obtenir un signal de sortie de 0-5V.

III.1.1. Electrodes

Dans notre réalisation nous allons exploiter des électrodes utilisées dans la plupart des méthodes d'analyse dans le domaine de l'électrochimie. Les électrodes sont des conducteurs électroniques, ou ioniques dont la fonction est de capter ou de libérer des électrons.

Les composants de base d'une cellule électrochimique sont : une électrode de travail, et une électrode de référence. Ces électrodes sont enfermées dans l'enveloppe de la cellule en contact avec un électrolyte liquide.

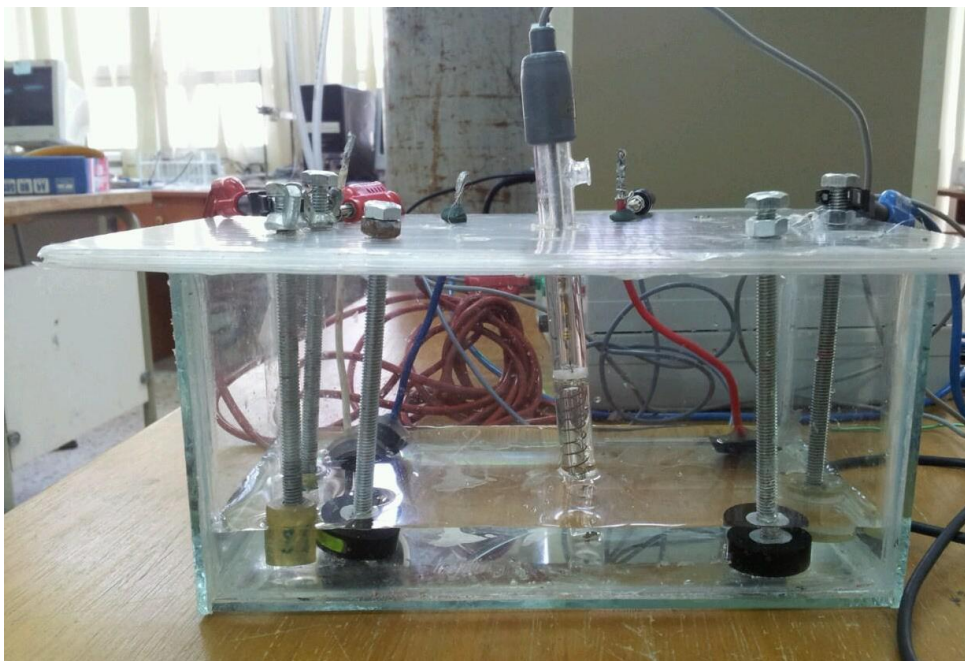


Figure II.2 : Cellule électrochimique

- Les électrodes de travail sont constituées des matériaux à étudier et dont le potentiel varie en fonction de la solution utilisée.



Figure II.3 : Electrode de travail

- Une électrode de référence est un élément dont le potentiel est fixe, ne variant pas pendant la mesure. Ainsi les mesures effectuées se résument à des mesures de tension entre les électrodes de travail et l'électrode de référence.



Figure II.4 : Electrode de référence

III.1.2. Filtrage du signal

Le signal pouvant être perturbé par du bruit, la fonction principale de notre filtre est d'éliminer totalement ou partiellement un certain nombre d'harmonique véhiculées par le signal mais ne faisant pas partie du signal utile. Les signaux à acquérir étant à faible variations dans le temps, on opte pour un filtre passe bas (**Figure II.4**) de fréquence de coupure de 10 environs.

➤ Calcule de la fréquence de coupure
$$F_c = \frac{1}{2\pi R_1 C_1}$$

$$\Rightarrow F_c = \frac{1}{2 * 3.14 * 1.5k * 10nf}$$

$$\Rightarrow F_c = 10.61 \text{ Hz}$$

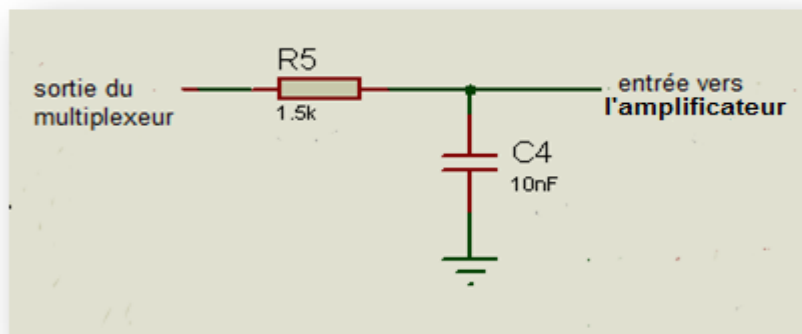


Figure II.5 : Filtre passe bas

III.1.3. Amplification

Une fois le signal filtré, il passe par un amplificateur à AO (**Figure II.6**) de gain 1 qui introduit un offset de 2.5V, et on aura en sortie un signal variant de 0 à 5 volts.

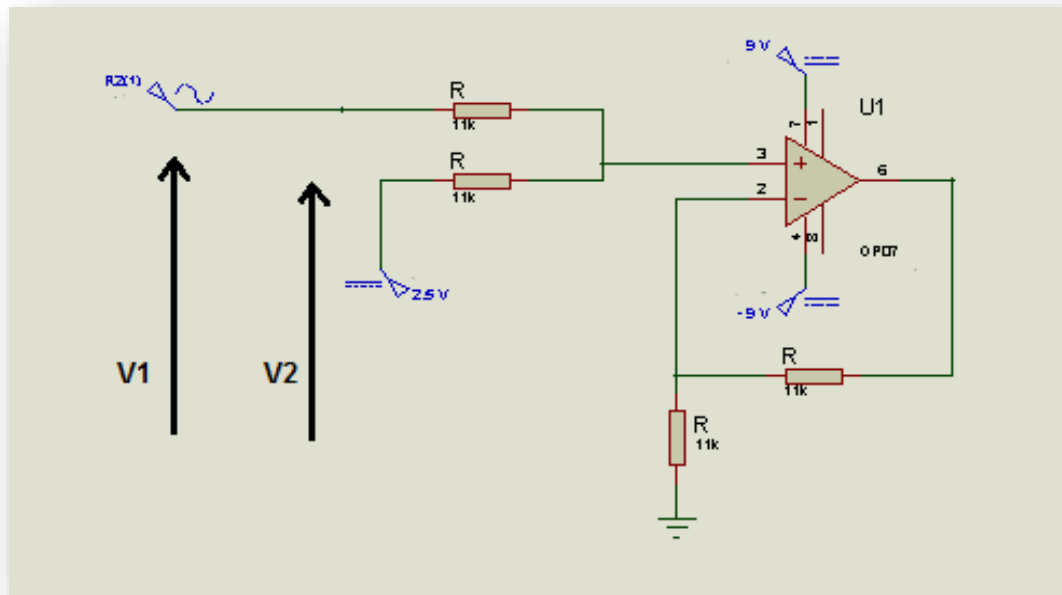


Figure II.6 : Amplificateur avec offset

➤ **Calculer de V_s en fonction de V_1 et V_2**

On suppose qu'avec ce montage on a:

- Un fonctionnement linéaire
- Un AOP supposé parfait, donc $E=0V$ et ainsi $V_+=V_-$

On aura donc :

$$V_s = \left(\frac{R_1 + R_2}{2 \cdot R_1} \right) * (V_1 + V_2)$$

Dans notre réalisation $R_1=R_2=R_3=R_4=R$ ce qui donne un gain de 1

Donc $V_s=V_1+V_2$

Pour améliorer la précision de la mesure, on pourrait introduire un amplificateur à gain programmable qui permettrait des gains supérieurs à 1 dans le cas de faibles signaux.

❖ Tension de référence

Pour avoir l'offset au niveau de l'amplification, on utilisera une diode zener programmable de type TL431 qui permet de délivrer une tension de référence stable de 2,5 V. L'avantage de ce composant c'est qu'il délivre une tension très stable même pour une plage de courant d'utilisation assez étendue, ce qu'on ne peut guère obtenir avec une diode zener classique. Pour ce faire, elle doit avoir un courant de polarisation minimum de 1 mA, afin de fournir sa tension de référence indiquée de 2,5 V. D'où l'utilisation d'une résistance de 2.2k comme indiqué sur la (Figure II.7)

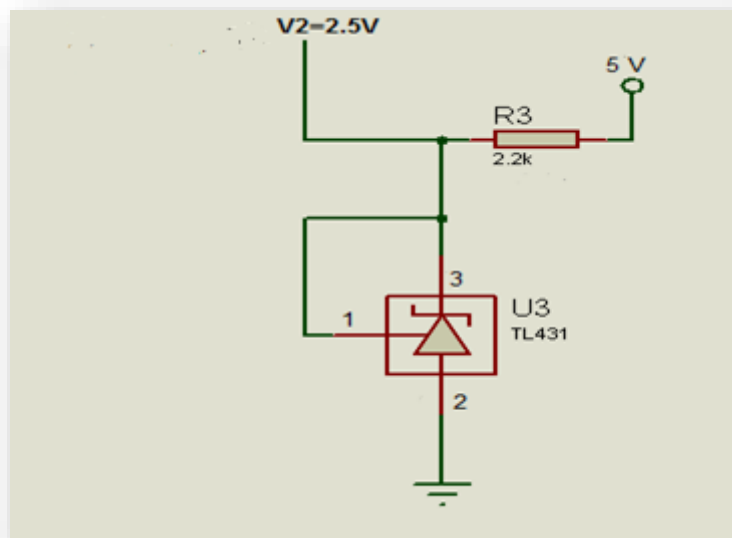


Figure II.7 : Tension de référence

III.1.4. Alimentation du circuit

L'ensemble de notre système est alimenté par le circuit illustré ci-dessous, qui délivre deux alimentations symétriques de +/- 9V (régulateurs 7809 et 7909) et +/- 5V en ajoutant en parallèle les régulateurs 7805 et 7905.

Pour avoir un système autonome, qui sera fonctionnel même en cas de coupure de courant, on a prévu une batterie de secours de 12 V accompagnée d'un convertisseur DC/DC pour avoir une alimentation symétrique.

Les condensateurs C1 de (2200 μ F) et C2 de (220 μ F) permettent de filtrer la tension d'entrée pour le régulateur, et les condensateurs C3 (100nF) permettent de lisser la sortie de ce dernier pour la charge.

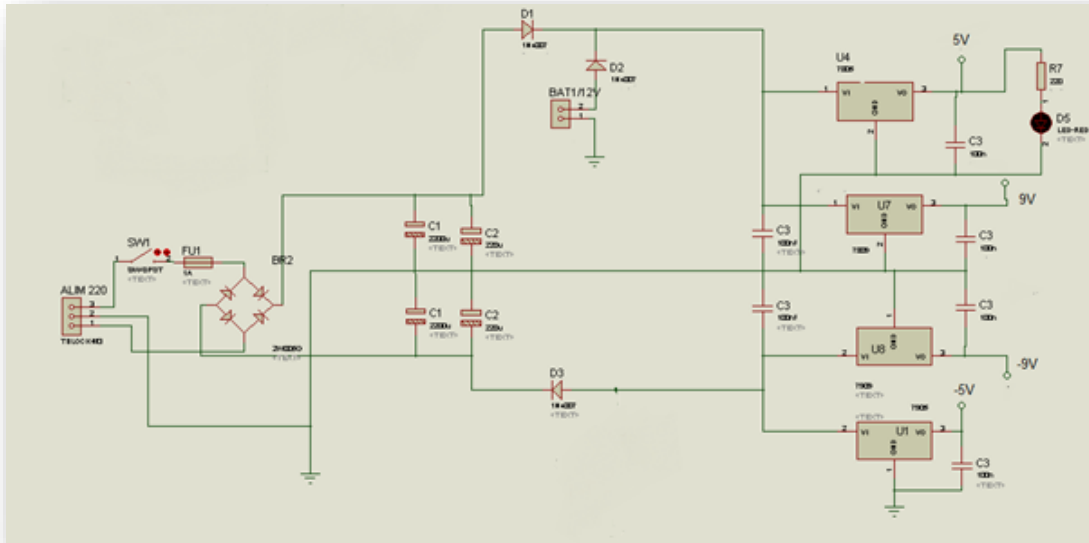


Figure II.8 : Schéma électrique de l'alimentation

La figure suivante représente le circuit analogique de notre système

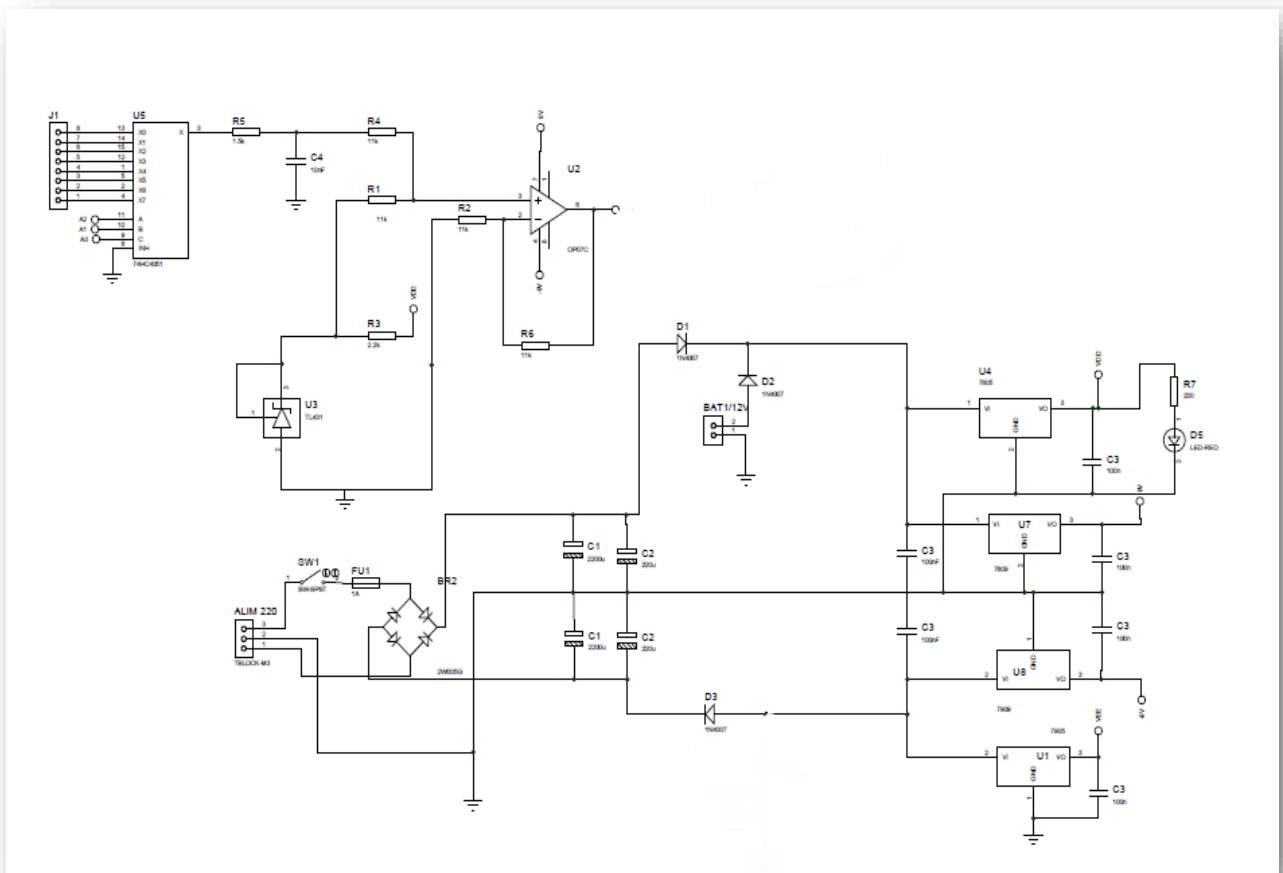


Figure II.9 : le circuit analogique du système

III.2. Partie numérique

Le data-logger réalisé est architecturé autour de la carte Arduino YUN qui est une carte électronique ayant des fonctionnalités avancées pour réaliser des prototypes d'objets connectés. L'Arduino Yun a le grand avantage de pouvoir communiquer directement avec un réseau local ou internet, soit en filaire ou en wifi. Elle simplifie donc considérablement la communication distante, via internet, entre un ordinateur et des capteurs. Cette possibilité est due au système linux qui y embarqué.

III.2.1. Présentation de la carte Arduino

Le système Arduino est une carte électronique basée au tour d'un microcontrôleur et de composants minimum pour réaliser des fonctions plus ou moins évoluées à bas cout. Elle possède une interface USB pour la programmer.

C'est une plateforme open-source qui est basée sur simple carte à microcontrôleur, et un logiciel, véritable environnement de développement intégré, pour écrire, compiler et transfère le programme vers la carte à microcontrôleur.

III.2.2. Applications

Arduino peut être utilisé pour développer des applications matérielles industrielles légères ou des objets interactifs (créations artistiques par exemple), et peut recevoir en entrées une très grande variété de capteurs. Arduino peut aussi contrôler une grande variété d'actionneurs (lumières, moteurs ou toutes autres sorties matériels).

Les projets Arduino peuvent être autonomes, ou communiquer avec des logiciels sur un ordinateur (flash, processing ou MaxMSP). Les cartes électroniques peuvent être fabriquées manuellement ou bien être achetées pré assemblées, le logiciel de développement open-source est téléchargeable gratuitement.

Le domaine d'utilisation du système Arduino est très vaste, on retrouve l'utilité dans tous les domaines possible, dans le quotidien comme dans les domaines professionnels, par exemple, contrôler les appareils domestiques, fabriquer votre propre robot, faire un jeu de lumières, communiquer avec l'ordinateur, télécommander un appareil mobile (modélisme), réaliser des systèmes électroniques tels qu'une bougie électronique, une calculatrice simplifiée, un synthétiseur, etc. tous ces systèmes seront conçus à base d'une carte Arduino et un panel assez large de composants électroniques.

III.2.3. Description de la carte Arduino Yún [5]

La carte Arduino Yún est construite autour du microcontrôleur ATmega32u4 et du contrôleur Wi-Fi Atheros AR9331. Le système sur puce (SoC) Atheros fonctionne sous la distribution Linux Linino, une version d'OpenWRT pour Arduino. La carte Yún possède une gestion intégrée des liaisons Ethernet et Wi-Fi, un port USB-A, un logement pour carte Micro SD, 20 broches d'E/S numériques (dont 7 peuvent servir de sorties MLI/PWM, et 12 d'entrées analogiques), un oscillateur à quartz de 16 MHz, un connecteur micro USB, une embase ICSP, et 3 boutons d'initialisation (reset).

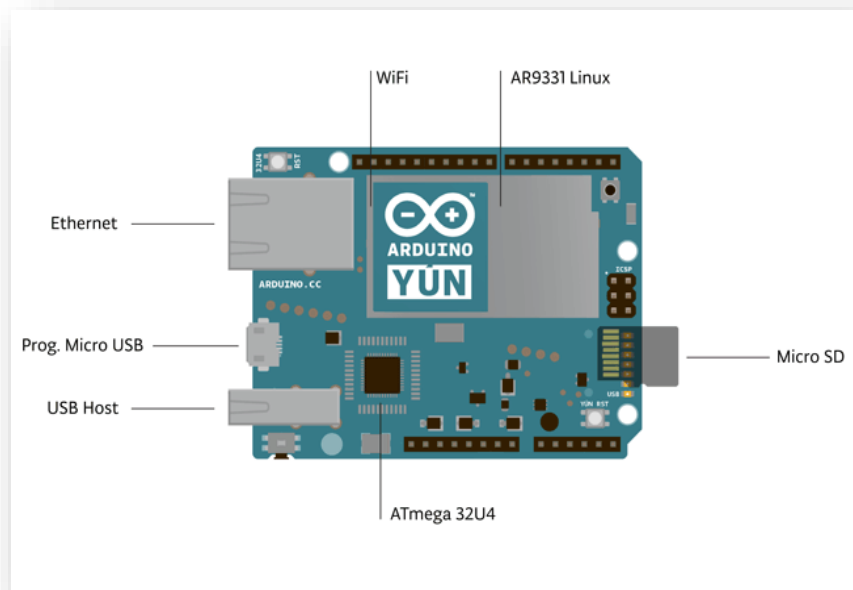


Figure II.10: Carte Arduino Yún

Comme évoqué dans un des paragraphes précédents, la carte Yún se distingue des autres cartes Arduino par sa capacité à communiquer avec la distribution Linux embarquée.

Elle est semblable à la carte Arduino Leonardo par le fait qu'elles partagent le même microcontrôleur et que ce dernier propose une communication USB.

Une fois reliée à un ordinateur, la carte Yún peut ainsi être reconnue par le système d'exploitation comme un port COM/série virtuel (ou CDC, Communication Device Class) et émuler une souris ou un clavier.

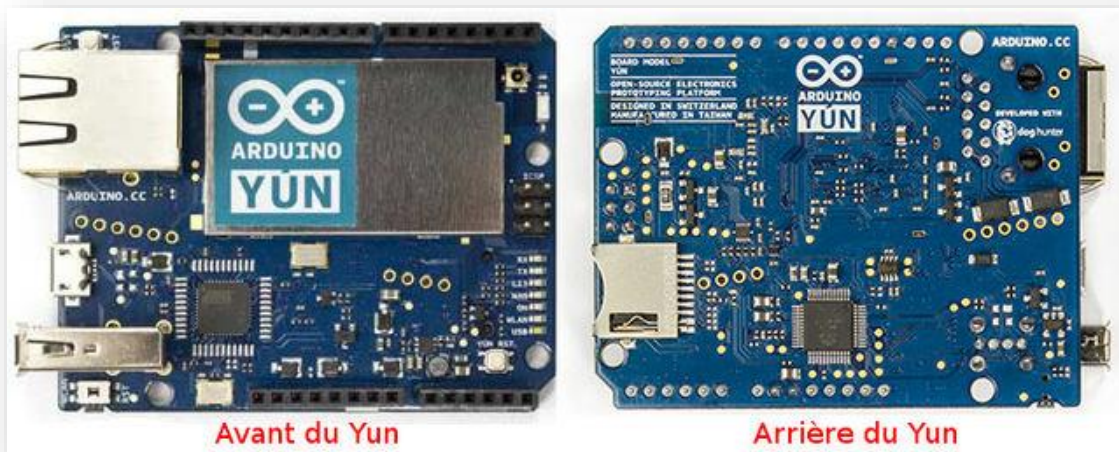


Figure II.11 : Carte Arduino vue avant et arrière

➤ On retrouve aussi plusieurs LEDs de statuts sur le Yún indiquant : l'alimentation (ON), la connexion Wifi(WLAN), la connexion Ethernet (WAN) et la connexion USB (USB). De plus, le pin 13 (L13) est connecté à l'une des leds de statut. (6)

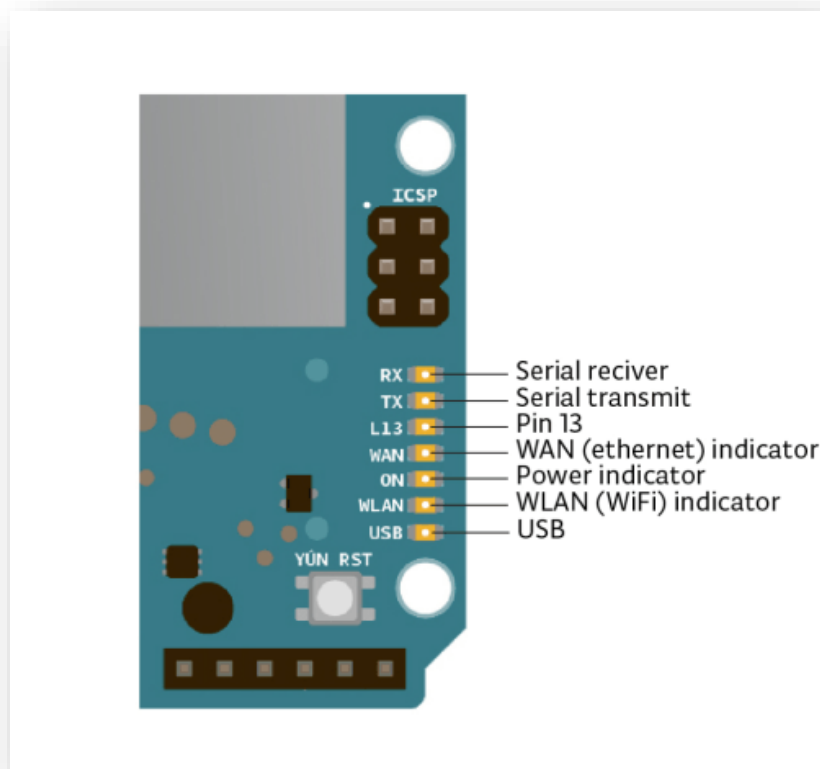


Figure II.12 : les différentes leds de la carte Arduino

➤ Les trois boutons de réinitialisation avec des fonctions différentes sur la carte sont représentés ainsi

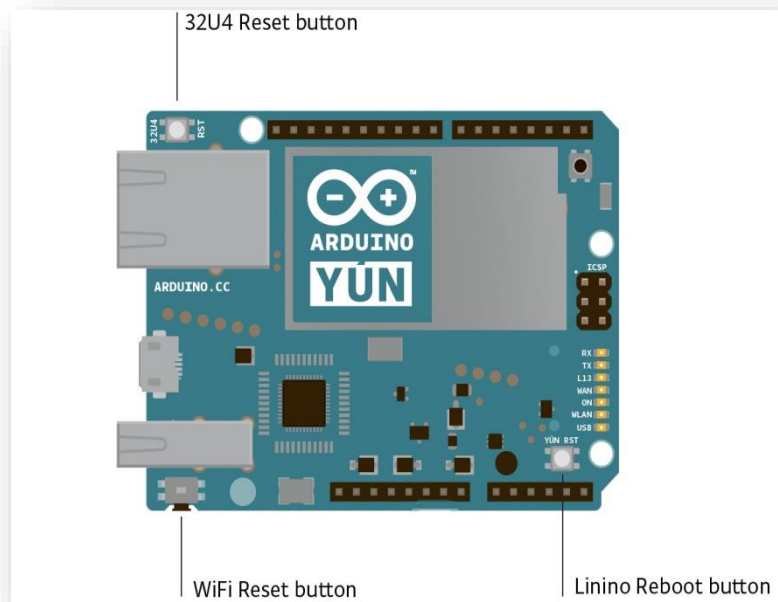


Figure II.13: les trois boutons de réinitialisation de la carte Arduino

- **Linux Reset** : Permet de réinitialiser le microprocesseur AR9331. Remise à zéro du AR9331 provoquera le redémarrage du système Linux. Toutes les données stockées dans la mémoire RAM seront perdues et tous les programmes qui sont en cours d'exécution seront terminés.

- **32U4 Reset** : permet de réinitialiser le microcontrôleur ATmega32U4.

- **Wifi Reset** : Ce bouton a une double fonction, en premier lieu, il sert à restaurer le WiFi à la configuration usine. Cette dernière, consiste à mettre le WiFi du YUN en mode point d'accès (AP) et lui attribuer l'adresse IP par défaut est 192.168.240.1, dans cet état, ordinateur peut se connecté au réseau, un WiFi qui apparaissent avec le nom SSID "Arduino Yun-XXXXXXXXXXXX", où les douze «X» représentent l'adresse MAC de Yún.

Une fois connecté, on peut rejoindre le panneau Web du Yún avec un navigateur au 192.168.240.1 ou "http://arduino.local" adresse.

La restauration de la configuration WiFi provoquera le redémarrage de l'environnement Linux.

Pour restaurer la configuration Wifi, on doit appuyer et maintenir le bouton Wifi Reset pendant 5 secondes. Lors de cette action, le bouton de la LED bleue WLAN se met à clignoter jusqu'à l'enregistrement de la procédure, il faut compter environ 5 seconds.

La deuxième fonction de la touche Wifi Reset est de restaurer l'image de linux à l'image d'usine par défaut. Pour restaurer l'environnement de Linux, Il faut appuyer sur le bouton

pendant 30 secondes. la restauration de l'image d'usine provoquera la perte de tous les fichiers enregistrés et logiciels installés sur la mémoire flash embarquée connectée à l'AR9331.[6]

III.2.4. Caractéristiques des deux processeurs [7]

La carte Arduino Yún est équipée de deux processeurs. L'un est un ATmega32U4 comme sur l'Arduino Leonardo. L'autre processeur est un Atheros 9331, fonctionnant sous Linux et dispose d'une pile WiFi basé sur OpenWRT. Cela offre les connectivités aux réseaux WiFi et Ethernet. Il est possible d'appeler des programmes ou des scripts personnalisés sur le système Linux depuis l'Arduino pour se connecter sur différents types de services "En Ligne". Leurs caractéristiques sont présentées séparément :

- **Arduino microcontrôleur ATmega32U4**

Microcontrôleur	ATmega32U4
Tension de fonctionnement	5V
Tension d'entrée	5
E / S numériques Pins	20
voies PWM	7
Pins d'entrée analogique	12
DC Courant par I O Pin /	40 mA
Courant DC pour 3.3V Pin	50 mA
Mémoire flash	32 Ko (dont 4 Ko utilisé par bootloader)
SRAM	2,5 KB
EEPROM	1 KB
Vitesse de l'horloge	16 MHz

Tableau II.1 : Les caractéristiques du microcontrôleur ATmega32U4 [7]

- **Microprocesseur Atheros AR9331**

Processeur	Atheros AR9331
Architecture	MIPS @ 400MHz
Tension de fonctionnement	3.3V
Ethernet	IEEE 802.3 10 / 100Mbit / s
Wifi	IEEE 802.11b / g / n
USB type A	2.0 Host
Lecteur de cartes	Micro-SD uniquement
RAM	64 Mo DDR2
Mémoire flash	16 Mo
SRAM	2,5 KB
EEPROM	1 KB
Vitesse de l'horloge	16 MHz
PoE 802.3af compatible support de la carte	Voir <i>Puissance</i>

Tableau II.2 : Les caractéristiques Atheros AR9331 [7]

III.2.5. Le pont « Bridge » [7]

Les deux microprocesseurs de la carte Yún peuvent communiquer entre eux grâce à une liaison série (port 0 (RX) et 1 (TX) permet de recevoir (RX) et transmettre (TX)).

D'un côté, il y a le processeur ATmega qui gère les capteurs et les échanges de données entre eux, L'autre est une Atheros 9331, fonctionnant sous Linux et la pile sans fil OpenWRT, qui permet au conseil de se connecter aux réseaux WiFi et Ethernet.

Il est possible d'appeler des programmes ou des scripts personnalisés sur le système Linux à travers l'Arduino et permettre de se connecter avec différents services Internet. Pour pouvoir communiquer entre eux, ils utilisent un bridge (pont).

Le schéma ci-dessous, représente les différents échanges entre les microprocesseurs à travers le bridge :

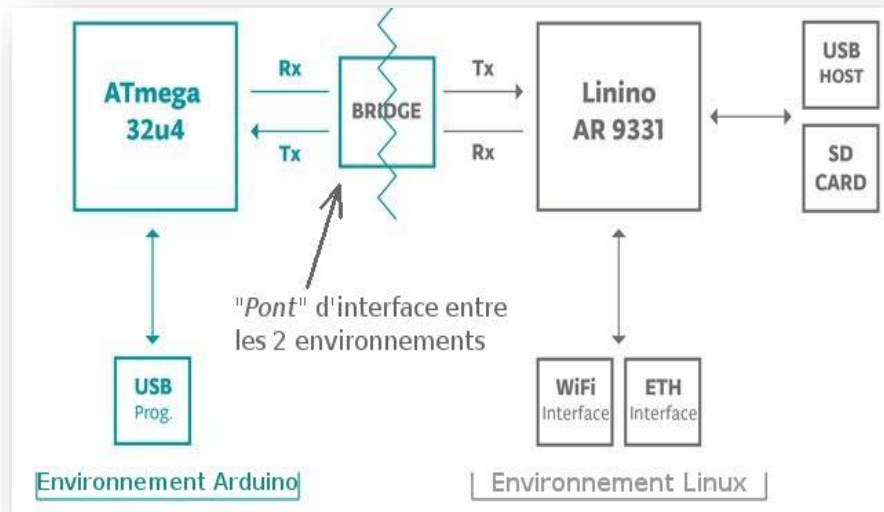


Figure II.14 : Pont de communication entre les deux microprocesseurs [7]

Le Bridge permet la communication dans les deux directions, agissant comme une interface pour envoyer des lignes de commandes à Linino.

La librairie Bridge simplifie la communication entre l'ATmega32U4 et l'AR9331. Bridge hérite de Stream et beaucoup de méthodes devraient être familières puisqu'elles sont semblables à la bibliothèque Serial et autres dérivés de Stream.

Les commandes Bridge en provenance du 32U4 sont interprétées par Python sur le processeur AR9331. Son rôle est d'exécuter des programmes sur la plateforme GNU/Linux lorsque cela est demandé par Arduino, fournir un espace de stockage pour partager des données telles que lecture de capteur entre Arduino et Internet. Le processeur AR9331 peut également recevoir des commandes depuis Internet et les passer directement à la partie Arduino.

III.2.5.1. Bibliothèques de bridge [8]

La bibliothèque Bridge facilite la communication entre les deux processeurs. Elle permet de lancer des scripts shell depuis des croquis Arduino, de communiquer avec des interfaces réseau, et de recevoir des données du processeur AR9331.

- **Process**

La bibliothèque *Process* est utilisée pour lancer des processus (des programmes) sur le processeur Linux. Elle est également utilisée pour lancer des shell scripts.

- **Console**

La bibliothèque *Console* peut être utilisée pour communiquer avec le moniteur réseau disponible dans Arduino IDE, par l'intermédiaire d'un shell. La fonctionnalité est très similaire au moniteur Série d'Arduino.

- **FileIO**

FileIO offre une interface vers le système de fichier Linino. La bibliothèque peut être utilisée pour lire et écrire des fichiers sur la carte SD.HttpClient.

- **Mailbox**

Mailbox offre une interface asynchrone, sans session pour permettre la communication entre Linino et Arduino.

- **YunClient**

Un équivalent du client HTTP pour Arduino, conçu sur le modèle de la classe EthernetClient.

- **YunServer**

Un équivalent du serveur HTTP pour Arduino, conçu sur le modèle de la classe EthernetServer.

- **Temboo**

Une interface vers Temboo permettant de réaliser facilement des connexions vers une large variété d'outils "en ligne".

- **Spacebrew**

Spacebrew est "une façon simple de connecter des objets interactifs les uns aux autres" qui utilise des WebSockets pour communiquer.

III.2.6. Alimentation de la carte Arduino Yún [13]

Il est recommandé d'alimenter la carte en 5 V via le connecteur micro USB. Si vous alimentez la carte via la broche Vin, les 5 V délivrés doivent être régulés. La carte ne comprend aucun régulateur de tension, donc toute tension supérieure l'endommagera.

La carte peut également être alimentée par Ethernet (compatibilité PoE, *Power over Etherne*), mais pour cela, on doit monter un module PoE sur la carte.

III.2.7. Réseau et configuration du Wifi [9]

III.2.7. 1. Connexion Arduino Yun sur Wifi

La carte Yún a l'avantage de supporter deux options pour se connecter au réseau : Ethernet et le Wifi, si Yun n'a pas déjà été connecté au WiFi, il va créer son propre hotspot Wifi.

- On branche la carte Yún à l'ordinateur avec câble micro USB.
- Pour que le Yun démarre il faut attendre environ 60 secondes.
- Lors du démarrage de la carte, elle va déclarer automatiquement un réseau Wifi intitulé Arduino Yun-XXXXXXXXXXXXX. On se connecte à ce Wifi depuis l'ordinateur.

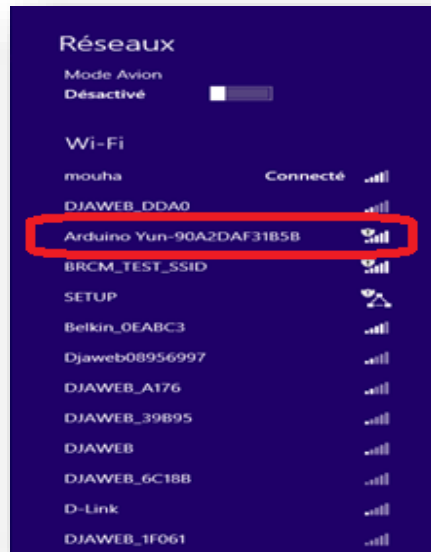


Figure II.15 : Arduino Yún sur Wifi

Une fois connecté au réseau Wifi, on lance le navigateur **http://arduino.local** -ou- directement **http://192.168.240.1** si la première option ne fonctionne pas.

La fenêtre représentée dans la **Figure II.17** est protégée par un mot de passe qui est par défaut « **arduino** ».

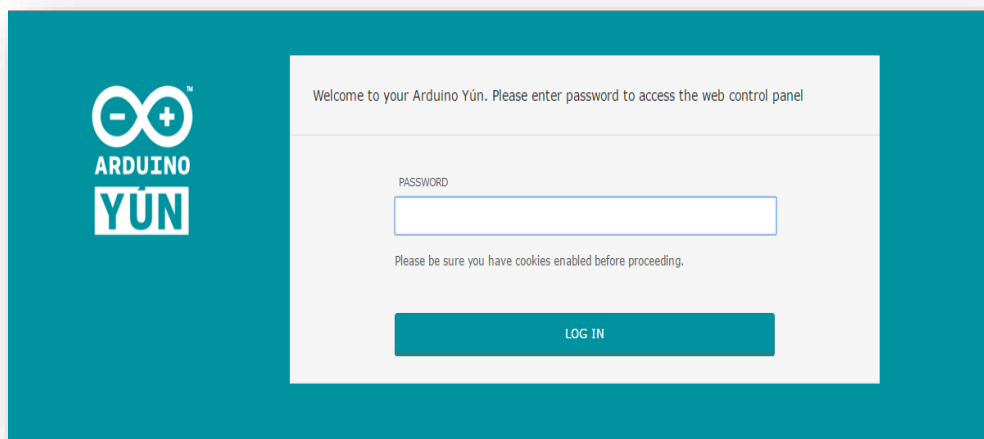


Figure II.16 : Plateforme d'accès a Arduino Yún

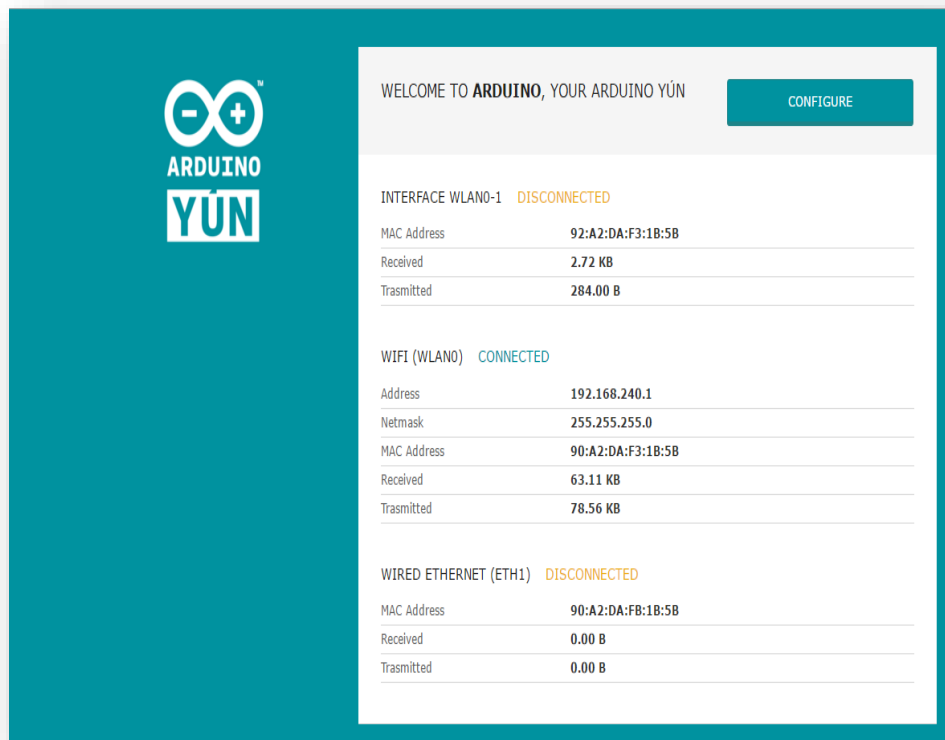


Figure II.17 : Interface Arduino après connexion au Wifi arduino Yún

On peut effectuer des modifications sur la carte arduino, comme changer le nom de la carte et le mot de passe. Une fois les modifications faites, on appuie sur “Configure et redémarrer ” pour finaliser l’opération et redémarrer la Yún.

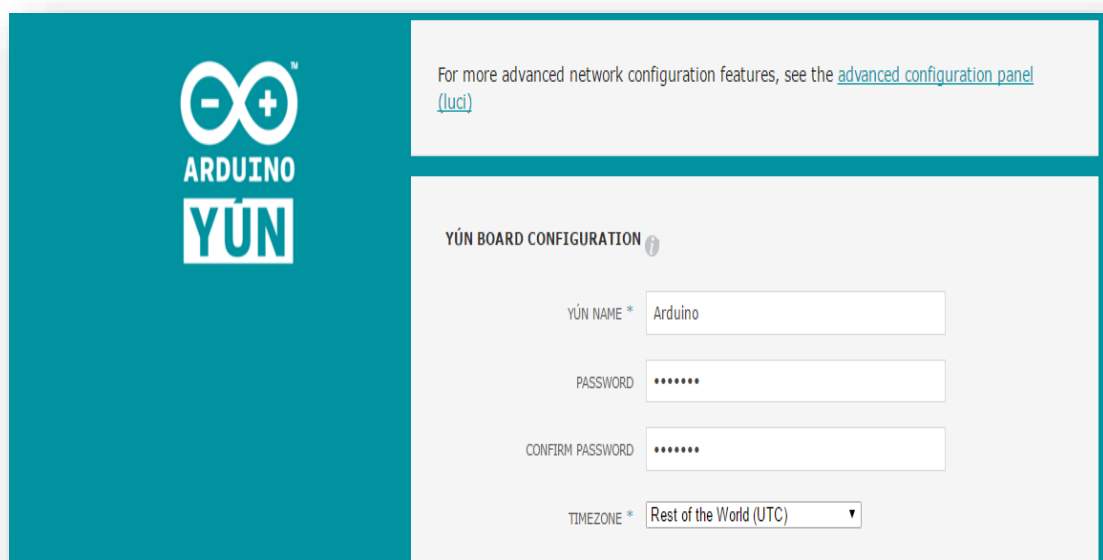


Figure II.18 : Icône de configuration de la carte Arduino Yún

III.2.7. 2. Connexion au réseau Wifi

Pour ce connecté au réseau wifi, il faut d'abord qu'on soit connecté à *Arduino YUN-09XXXX*

Sur l'icône de la **Figure II.19**, se fait la modification du nom de la carte et le mot de passe, ainsi la connexion à un réseau wifi on sélectionnant le réseau souhaité dans la liste et on introduit son mot de passe

Le défaut de cette application, est que si on obtient le mot de passe erroné, l'appareil ne nous renvoie pas un message d'erreur, si Arduino redémarre mais on voit toujours son Arduino Yun-902XXXX comme une option disponible, il y a une bonne chance que notre mot de passe Wifi est incorrect. [5]

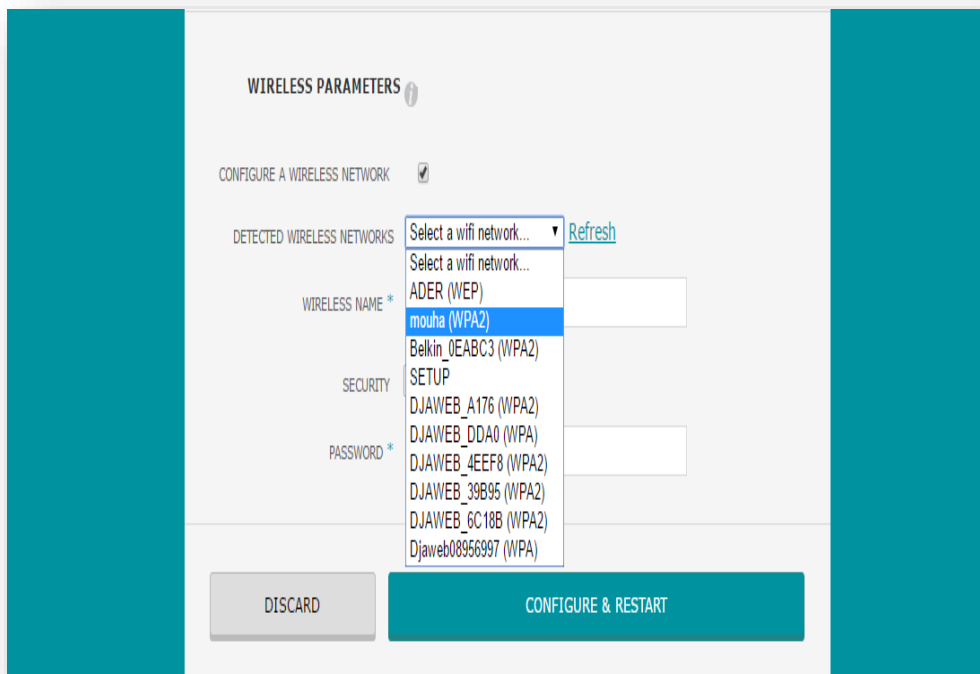


Figure II.19 : Connexion de la carte au réseau Wifi

- Si les choses se compliquent et qu'on souhaite revenir à la configuration usine du Wifi, on presse sur le bouton WLAN RST pendant au minimum **5 secondes** (mais surtout moins de **30 secondes** – pour éviter de restaurer l'environnement linux).

III.2.8. La carte SD Arduino [10]

La communication entre le microcontrôleur et la carte SD utilise SPI en employant La bibliothèque SD qui permet de lire et d'écrire sur des cartes SD , qui a lieu sur les broches numériques 11, 12, et 13 (sur la plupart des cartes Arduino). En outre, une autre broche doit être utilisée pour sélectionner la carte SD. Cela peut être la broche matériel SS - broche 10 (sur la plupart des cartes Arduino), ou une autre broche spécifiée dans l'appel à SD.begin () .

III.2.8.1. Formatage / Préparation de la carte [10]

Pour être sûr que la carte SD utilisé peut être lus par la bibliothèque SD, il faudrait la formater pour qu'un système de fichier que Arduino peut lire et écrire sera créer.

Il est souhaitable de ne pas formater les cartes SD fréquemment, car elle diminue leur durée de vie.

On aura besoin d'un lecteur de SD et de l'ordinateur pour formater la carte. La bibliothèque prend en charge lesFAT16 et FAT32 filesystems, mais l'utilisation FAT16 lorsque cela est possible.

III.9. L'environnement de programmation Arduino

La carte Arduino présente le noyau de notre système, et pour qu'on puisse la programmer, un logiciel compatible est mis à notre disposition avec cette carte, ce logiciel est Arduino IDE.

L'environnement de développement Arduino (IDE en anglais) est une application Java multi-plateforme, c'est-à-dire, fonctionnant sur plusieurs systèmes d'exploitation.

Ce dernier, joue le rôle d'éditeur de code et de compilateur, qui peut transférer le firmware (le programme) a travers de la liaison série asynchrone.

La figure ci-dessous, représente l'interface visuelle de l'environnement à l'ouverture,

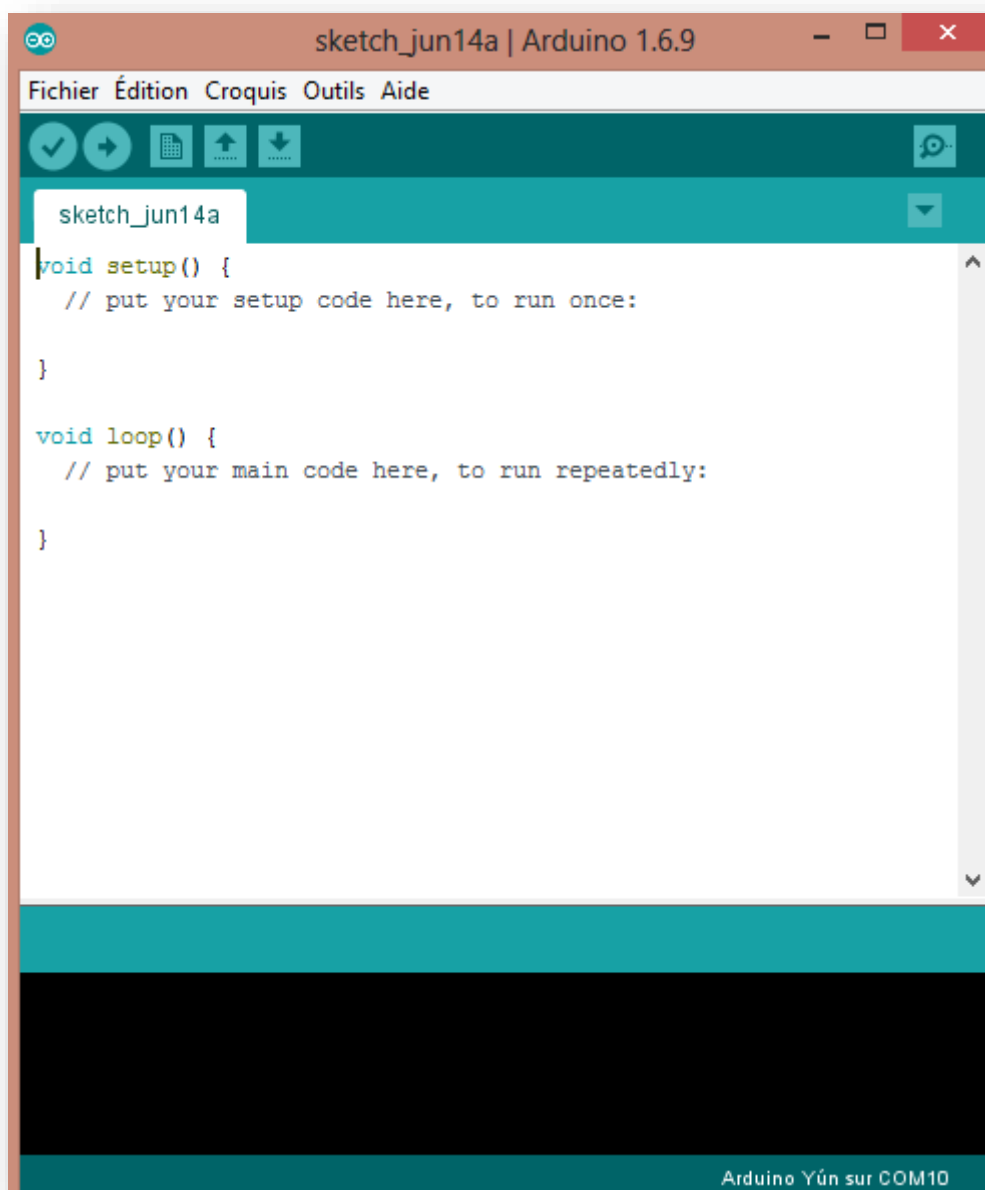


Figure II.21 : Fenêtre principale l'environnement de développement Arduino

III.9.1. Description du logiciel Arduino [15]

Le logiciel Arduino a pour fonctions principales :

- D'écrire et compiler des programmes pour la carte Arduino
- De se connecter avec la carte Arduino pour y transférer les programmes
- De communiquer avec la carte Arduino

Cet espace de développement intégré (EDI) dédié au langage Arduino et à la programmation des cartes Arduino comporte plusieurs barres qu'on va voir en détail ci-dessous :

- une **BARRE DE MENUS** comme pour tout logiciel
- une **BARRE DE BOUTONS** qui regroupe tous les raccourcis du logiciel afin de simplifier au maximum son utilisation.
- un **EDITEUR** (à coloration syntaxique) pour écrire le code du programme, avec onglets de navigation,
- une **ZONE DE MESSAGES** qui affiche indique l'état des actions en cours,
- une **CONSOLE TEXTE** qui affiche les messages concernant le résultat de la compilation du programme

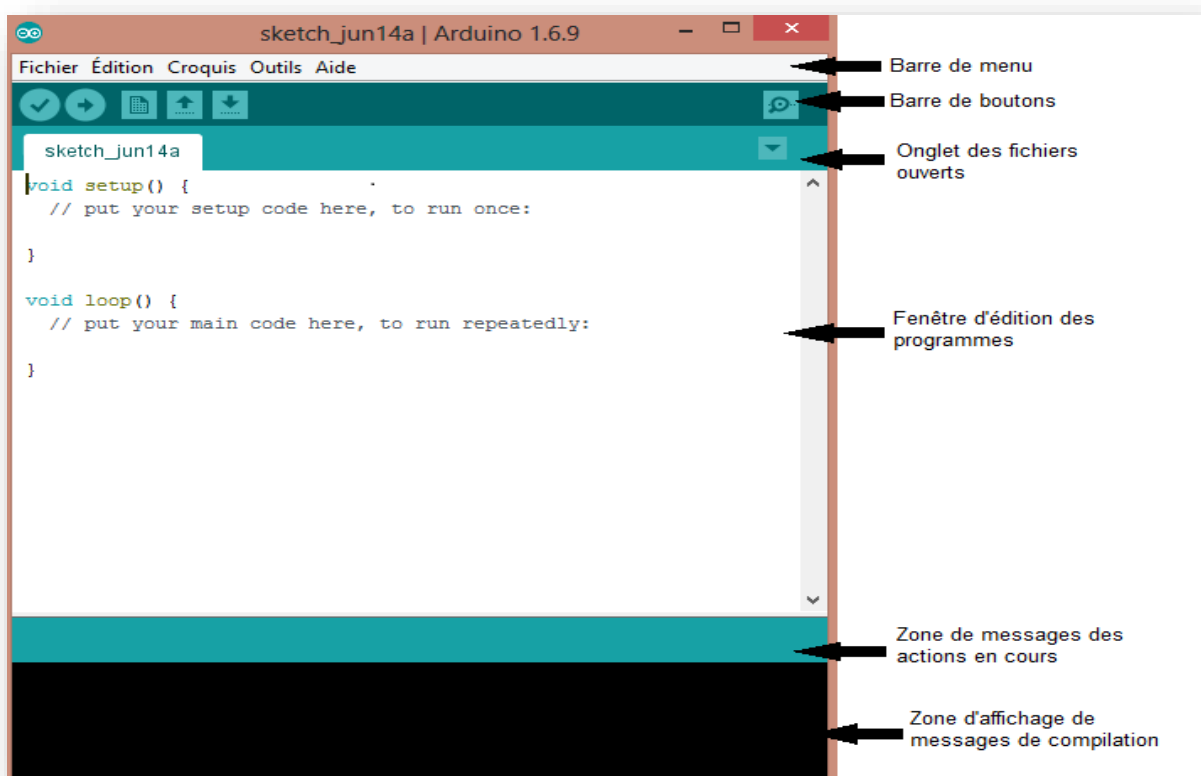


Figure II.22 : Les différentes parties de la fenêtre principale du logiciel Arduino

- Le logiciel Arduino intègre également :

Un **TERMINAL SERIE** qui permet d'afficher des messages textes reçus de la carte Arduino et d'envoyer des caractères vers la carte Arduino.

Cette fonctionnalité permet d'afficher sur l'ordinateur l'état de variables, de résultats de calculs ou de conversions analogique-numérique : un élément essentiel pour améliorer, tester et corriger ses programmes.



Figure II.23 : accès au moniteur série

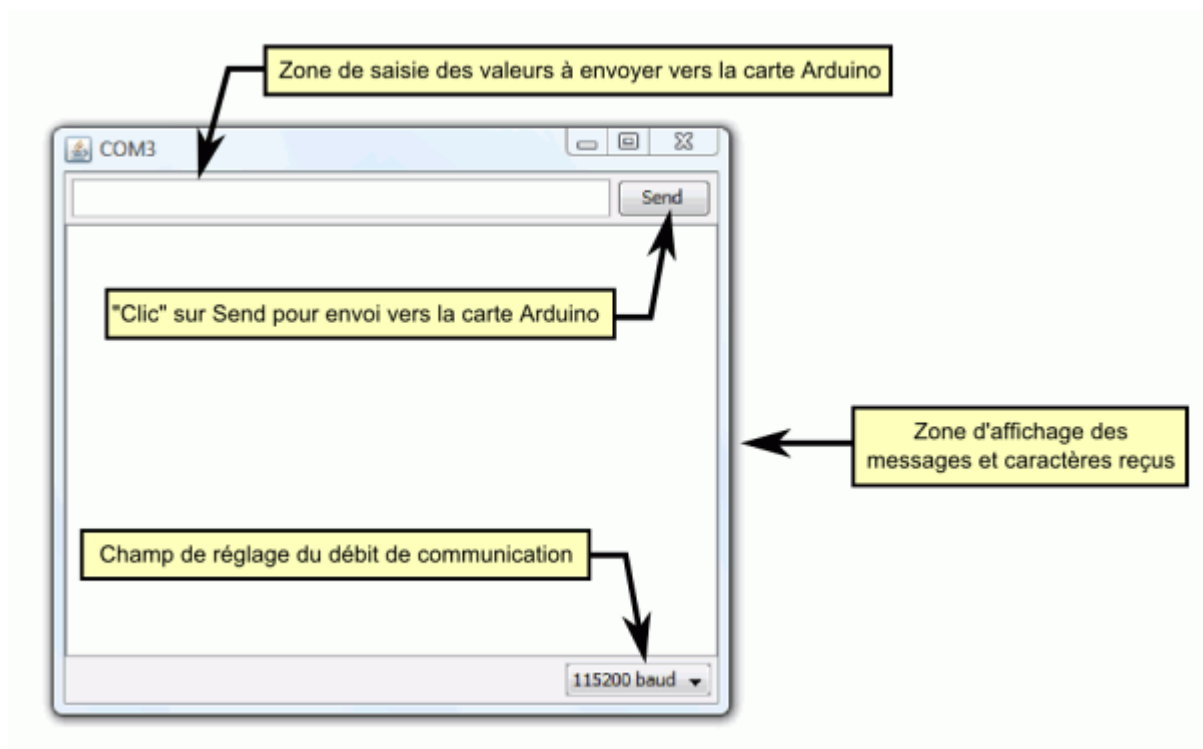


Figure II.24 : Fenêtre associée au moniteur série

III.9.2. Description de la barre des boutons

La barre des boutons nous permet de vérifier la syntaxe, le transfère des programmes, aussi la création, l'ouverture et sauvegarde d'un code écrit.



Figure II.25 : Structure de la barre des boutons

- Bouton 1 : Vérification de programme, il actionne un module qui cherche les erreurs dans votre programme
- Bouton 2 : Charger (téléverser) le programme dans la carte Arduino
- Bouton 3 : Création de nouveau programme.
- Bouton 4 : Ouvrir la liste de tous les programmes déjà sauvegarder, et choisir un programme on cliquant sur ce dernier.
- Bouton 5 : sauvegarde le programme avant la fermeture de la fenêtre

III.9.3. Description de la barre des menus

Les différents éléments du menu nous permettent de créer de nouveaux *sketchs*, de les sauvegarder, de gérer les préférences du logiciel et les paramètres de communication avec votre carte Arduino

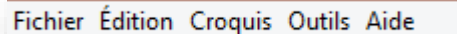


Figure II.26 : Structure de la barre des menus

IV- Organigramme

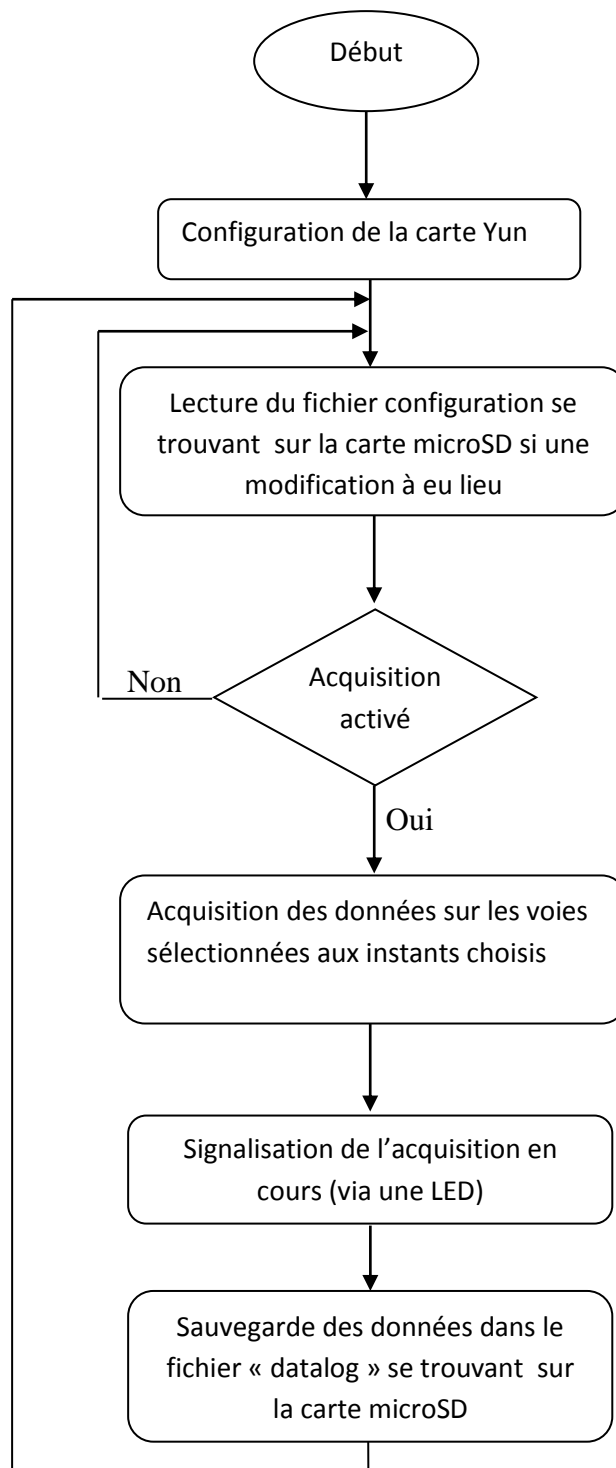


Figure II.26 : Organigramme de fonctionnement du système d'acquisition

1- Configuration de la carte Arduino YUN

La configuration de la carte arduino se fait comme suit :

- Affectation et configuration des différentes E/S utilisées
- Réglage de l'heure et la date sur la carte SD

2- Lecture du fichier configuration

Le fichier « config » qui se trouve dans la carte microSD contient les paramètres de configurations du système dont :

- **Le type d'acquisition** : l'acquisition des données se fait selon deux critères
0 : acquisition continue
1 : acquisition si variation
Pour le cas « acquisition si variable » l'écart de variation minimum en (mV) est d'environ 5mV
- **Les voies d'acquisition** :
L'acquisition peut se faire jusqu'à 8 voies, ce paramètre permet de choisir sur quelles voie se fera l'acquisition
1 2 3 4 5 6 7 8
- **Les temps d'acquisition** : dans le fichier config on peut programmer pour chaque voie un temps d'acquisition en second (s).
- **L'activation de l'acquisition** : 0 : acquisition désactivé
1 : acquisition activé

3- Le fichier « datalog »

Il sert à la sauvegarde des données acquises sous la forme suivante :

Date Heure CH1 CH2 CH3 CH4 CH5 CH6 CH7 CH8

4- Récupération des données

la récupération des données sauvegardée sur la carte SD peut se faire selon trois méthodes

- La récupération peut se faire par l'extraction de la carte SD dont la lecture se fera sur un ordinateur muni d'un lecteur de carte SD.
- Elle peut se faire aussi par le réseau via la carte Ethernet. (option à implémenter)
- Ou bien via le WIFI par l'intermédiaire d'une interface web dédiée à développer sous python (option à implémenter)

I. Introduction

Ce chapitre est consacré à la présentation de la réalisation ainsi que la description des différents tests effectués et les résultats obtenus.

II. Simulation et test de la partie analogique

Avant de réaliser notre carte électronique, nous avons vérifié le montage sur un logiciel de simulation Proteus, ensuite on l'a testé sur lab d'essai afin d'être sûr du bon fonctionnement de ce dernier, et sans aucune surprise, les résultats sont les suivants :

II.1. Test avec le logiciel de simulation proteus

Notre premier test est basé sur un logiciel de simulation Proteus qui est composé de deux logiciels principaux :

- ISIS, permettant entre autres la création de schémas et la simulation électrique,
- ARES, dédié à la création de circuits imprimés, la figure suivante représente le résultat de la simulation obtenu.

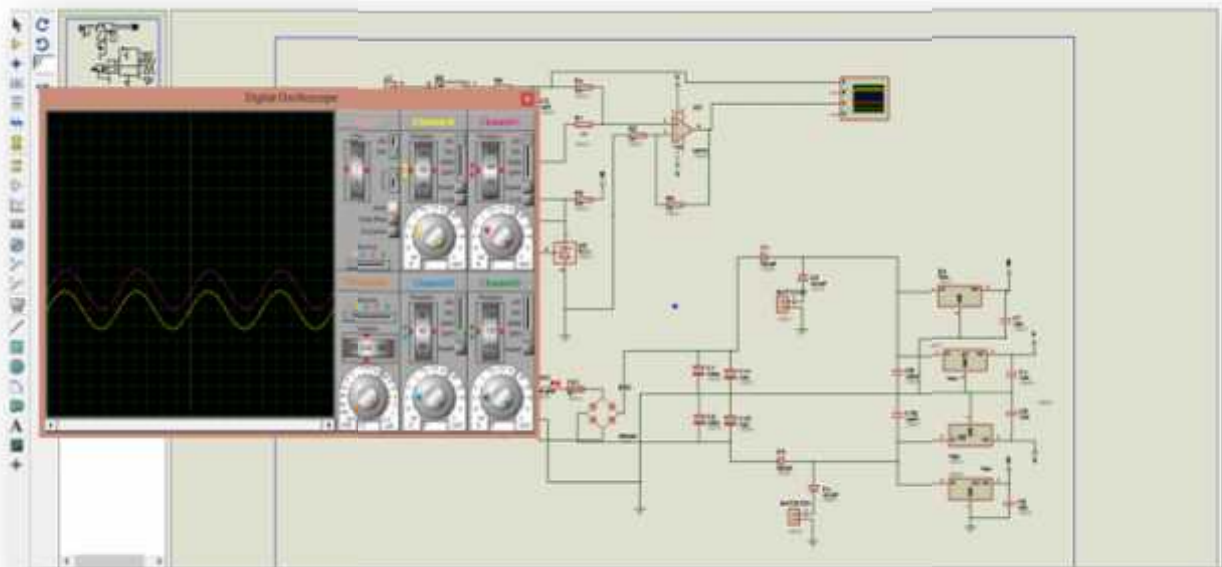


Figure III.1 : Résultat obtenu avec la simulation sous proteus

Après le fonctionnement du circuit avec le logiciel de simulation Proteus, on est passé à l'essai Lab d'essai qui se trouve sur la **Figure III.2**. Et le résultat obtenu est représenté sur la **Figure III.3**.

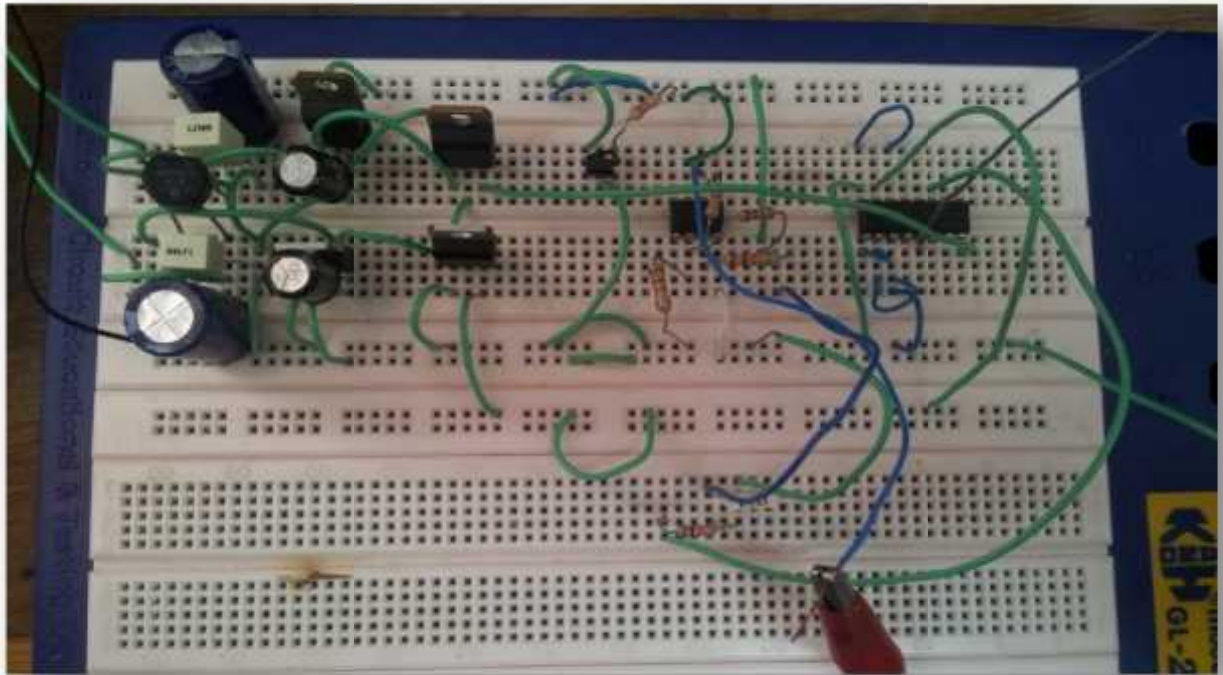


Figure III.2 : circuit électronique sur Lab d'essai

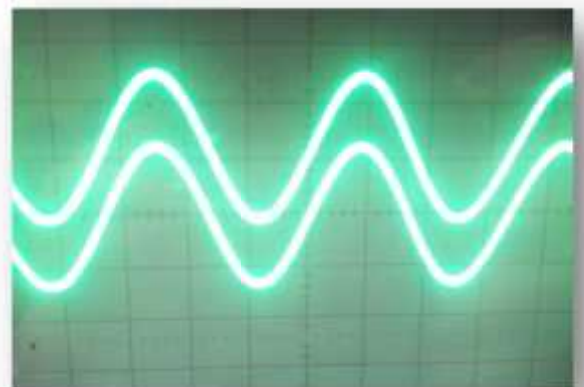
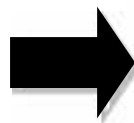
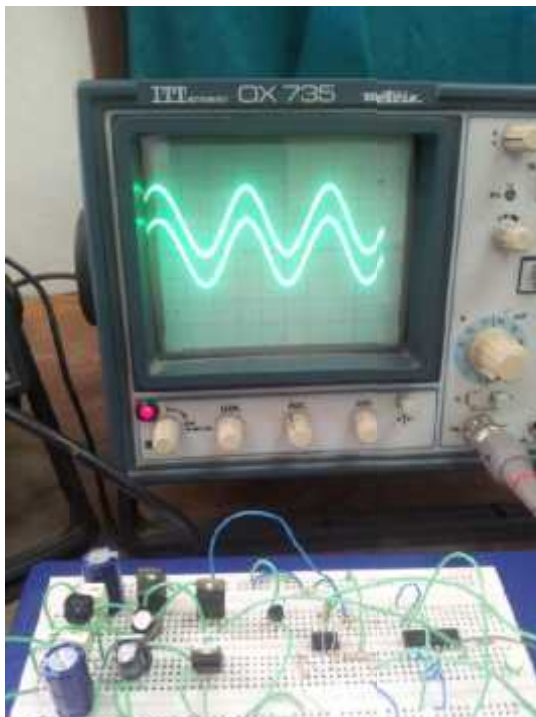


Figure III.3 : Signal obtenu à la sortie de l'amplificateur

III. Circuit imprimé

Une fois le circuit testé, on a procédé à la réalisation de son PCB (circuit imprimé). Les figures suivantes représentent le typon du circuit avec un schéma d'implantation des composants, et une vue 3D de la carte électronique.

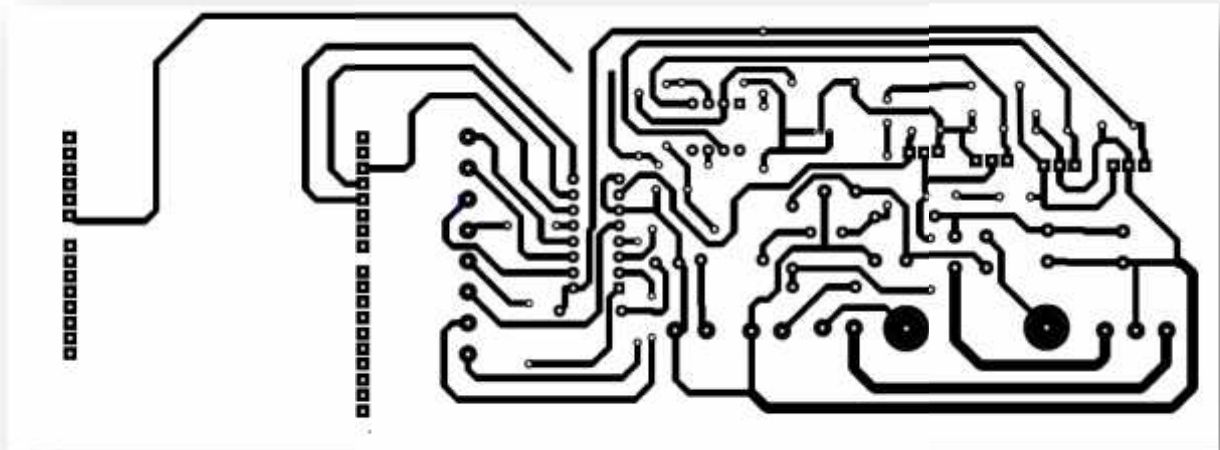


Figure II.4 : Circuit imprimé du système électronique

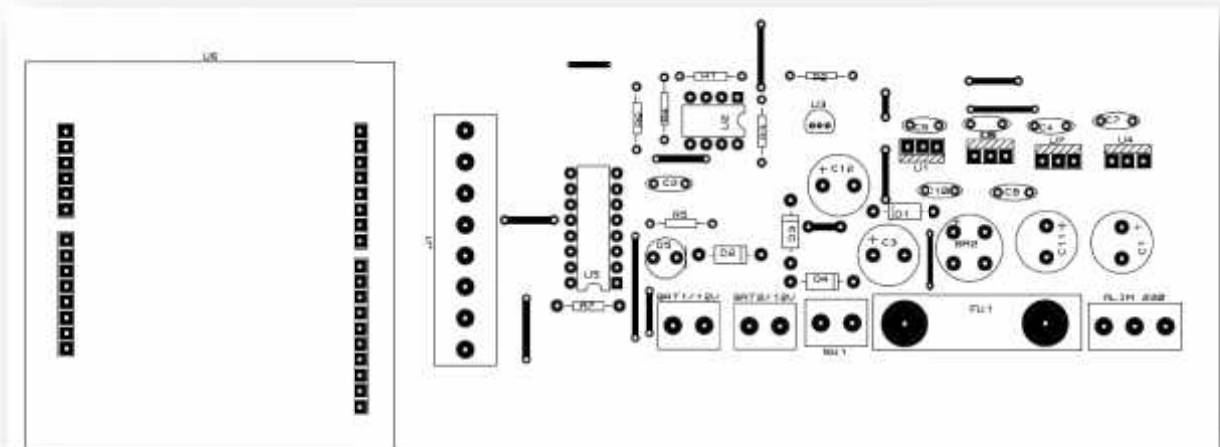


Figure II.5: Schéma d'implantation des composants de la carte électronique

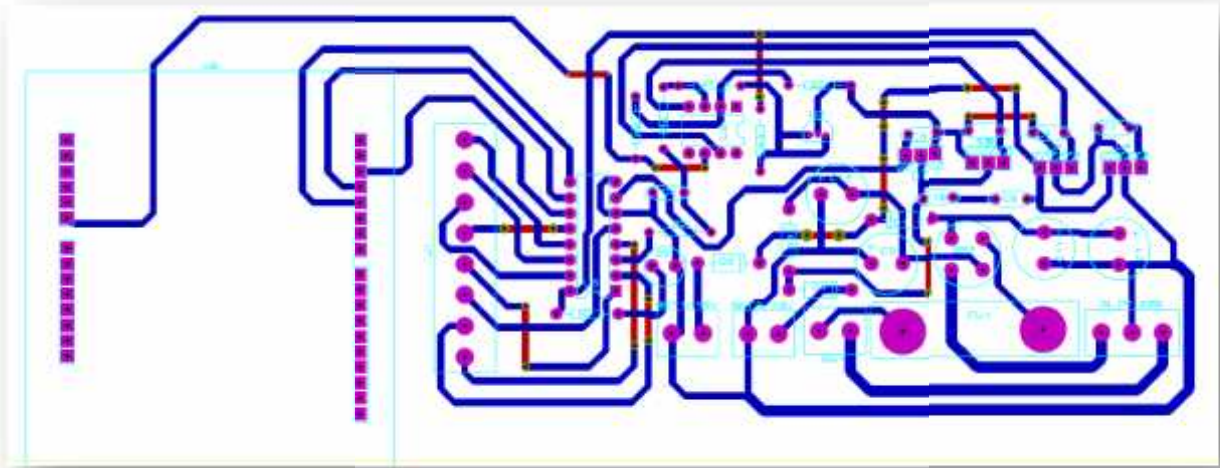


Figure II.6 : Schéma du circuit imprimé avec l'implantation des composants sur la carte électronique

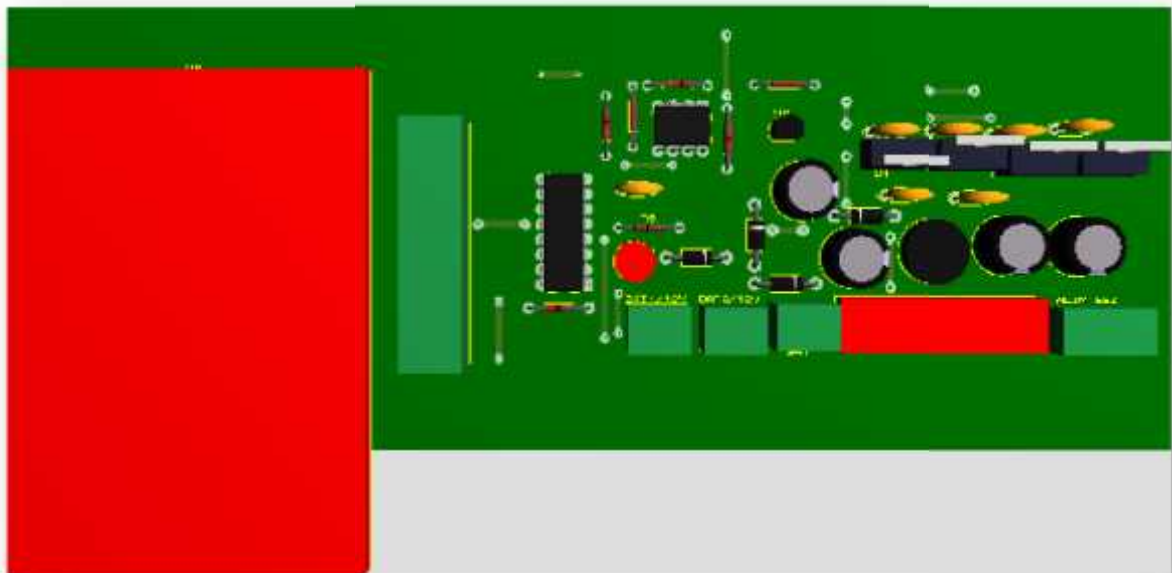


Figure II.8 : Vue 3D de la carte électronique

IV. Nomenclature de la carte électrique

Le tableau suivant présente la liste des composants du système réalisé

Type	Référence	Valeurs	Quantités
Resistance	R1,R2,R6,R4	11K	4
Resistance	R3	2.2K	1
Resistance	R5	1.5K	1
Resistance	R7	220k	1
Condensateur	C1	2200uF	2
Condensateur	C2	220uF	2
Condensateur	C3	100nF	6
Condensateur	C4	10nF	1
Amplificateur	U2	Op07	1
multiplexeur	U5	74HC4051	1
Régulateur	U4	7805	1
Régulateur	U1	7905	1
Régulateur	U7	7809	1
Régulateur	U8	7909	1
Régulateur	U3	TL431	1
Diode	D1, D2, D3, D4	1N4007	4
Pont diode	BR2	C1500R	1
Switch	SW1	-	1
Connecteur FU	FU1	-	1
Led	D5	-	1
Connecteur 8B	J1	-	1
Connecteur 3B	ALIM220	-	1
Connecteur 2B	BAT1	-	2

Tableau II.3 : Nomenclature de la carte électrique

V. Mise en boîtier du data-logger

La **Figure II.8** représente la carte électronique réalisée mise en boîtier



Figure II.8 : Carte électronique du data-logger

Les deux figures (**Figure II.9** et **Figure II.10**) représentent les deux faces du boîtier

La face avant du boîtier contient les 8 entrées du data-logger, une entrée de GND, ainsi que deux leds, une rouge et une blanche, l'une clignote lorsqu'il y a acquisition et l'autre s'allume lorsque le circuit est alimenté.



Figure II.9 : Face avant du boîtier

Quand à la face arrière du boîtier, on retrouve les trois connecteurs de la carte Arduino, un connecteur USB A, un connecteur USB B, ainsi le connecteur Ethernet. Sur la même face, on retrouve aussi deux entrées pour connecter la batterie de secours, un fusible et un secteur pour alimenter le data-logger avec 220V



Figure II.10 : Face arrière du boîtier

VI- Logiciel de gestion

Nous avons réalisé un logiciel qui permet de piloter notre système, pour cela nous avons utilisé le logiciel de programmation Arduino, nous allons vous présenter quelque routine de programmation

-) Pour pouvoir communiquer avec la carte microSD on a besoin de faire appel à la bibliothèque « FileIO »

```
#include <FileIO.h>
```

-) Fonction qui permet de faire l'acquisition des données sur la pin analogique (0), on les ajoutant à la chaîne, en séparant les valeurs par une virgule est

```
For (count=0; count<=7; count++) {
```

```
    r0 = bitRead(count,0);
```

```
    r1 = bitRead(count,1);
```

```
    r2 = bitRead(count,2);
```

```
    digitalWrite (8, r0);
```

```
    digitalWrite (9, r1);
```

```
    digitalWrite (10, r2);
```

```
int readValue = analogRead(0);  
float Voltage = (((5.0/1023.0)*readValue)-2.5);  
  dataString += String (Voltage);  
  dataString += " , ";  
}
```

VII. Test du système et résultats obtenus

Après avoir réalisé notre système d'acquisition qui est le data-logger, nous avons effectués des tests au niveau du laboratoire d'électrochimie d'université de Tizi-Ouzou . Nous avons plongé trois électrodes de travailles et une électrode de référence dans une solution corrosif,

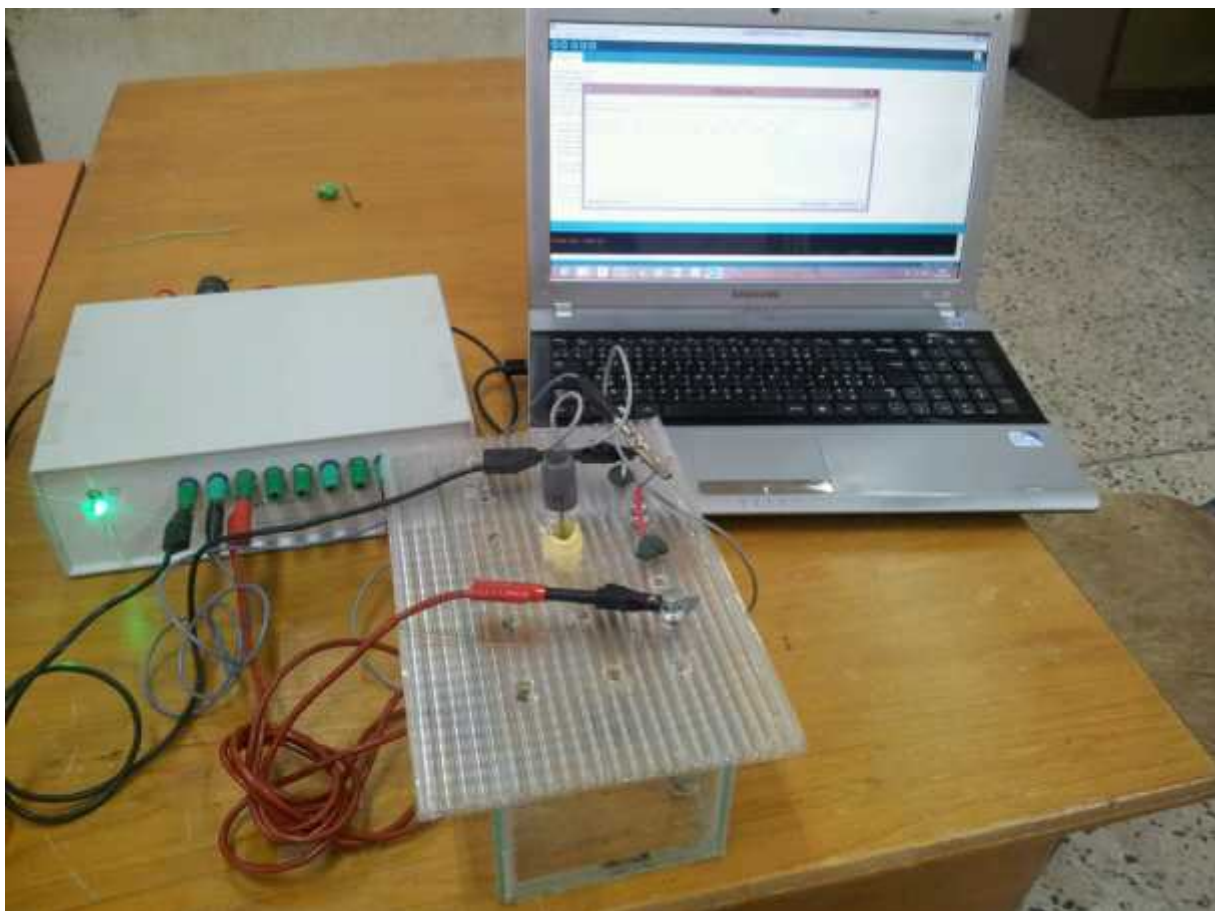


Figure II.11 : Test du système d'acquisition réalisé

Les résultats obtenus sont illustrés dans la **Figure II.11**

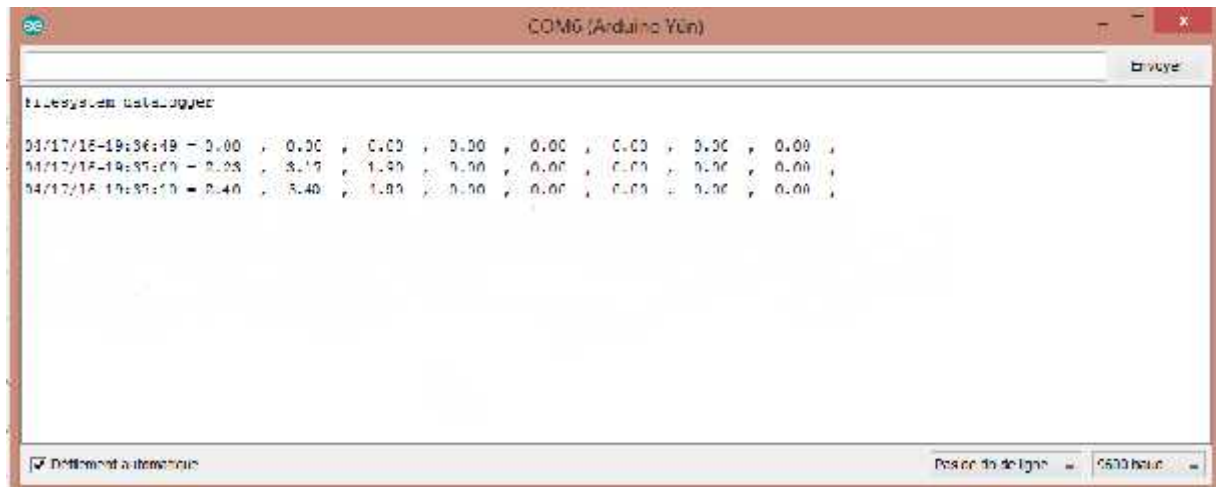


Figure II.11 : Résultats du obtenus

Conclusion générale

A travers notre projet de fin d'étude, nous sommes arrivés à réaliser un système qui « monitorise » un ensemble d'échantillons immergés dans un électrolyte généralement une solution corrosive. Ce système nous permet de faire une mesure simultanée d'une différence de potentiel, entre une électrode de travail et une électrode de référence. Ces mesures permettent d'évaluer la capacité anticorrosive du matériau d'électrode utilisé.

Notre système représente donc un enregistreur de données à 8 voies qui est un dispositif électronique permettant d'acquérir des données à intervalles réguliers, nommé « data-logger », il a été conçu à base d'une carte Arduino de type Yun. L'utilisation de cette carte est justifiée par ses différentes caractéristiques dont un lecteur de carte microSD intégré pour enregistrer les résultats et la possibilité de communiquer avec d'autres systèmes via les interfaces réseaux Ethernet ou Wifi.

Tout travail étant perfectible, on peut proposer comme perspectives à ce travail :

- la miniaturisation de la carte réalisée, en utilisant des CMS par exemple,
- l'introduction d'une RTC pour régler l'heure et la date de la carte Arduino.
- Rajouter un afficheur LCD au système sur la face avant du boîtier pour l'affichage d'informations sur le fonctionnement du système quand il est en mode autonome.
- Utilisation d'un flash disque sur le port USB au lieu d'une carte microSD pour faciliter l'accès aux données stockées de l'extérieur du boîtier.
- Développer une interface web qui permettra d'afficher les données sans extraire la carte microSD via l'interface Wifi.
- Utilisation d'un amplificateur à gain programmable qui permettra d'avoir des gains supérieurs à 1 dans le cas de faibles signaux

Références bibliographiques

- [1] <http://dspace.univ-tlemcen.dz/bitstream/112/4161/1/ebm90.pdf>
 - [2] <http://www.gorge.net.au/data-logger-what-it-is-uses-and-types/>
 - [3] <http://www.onsetcomp.com/what-is-a-data-logger>
 - [4] <http://www.jakar.cz/en/introduction-to-data-loggers>
 - [5] <https://www.elektor.fr/arduino-yun>
 - [6] <https://www.jameco.com/Jameco/Products/ProdDS/2193441QuickGuide.pdf>
 - [7] <https://www.arduino.cc/en/Main/ArduinoBoardYun>
 - [8] <https://www.arduino.cc/en/Reference/YunBridgeLibrary>
 - [9] <https://blogs.msdn.microsoft.com/jdupuy/2015/01/22/zone61-guide-pour-configurer-votre-carte-arduino-yn/>
 - [10] <https://www.arduino.cc/en/Reference/SD>
 - [11] <https://www.twilio.com/blog/2015/02/arduino-wifi-getting-started-arduino-yun.html>
 - [12] <https://blogs.msdn.microsoft.com/jdupuy/2015/01/22/zone61-guide-pour-configurer-votre-carte-arduino-yn/>
 - [13] <https://www.elektor.fr/arduino-yun>
 - [14] <https://www.jameco.com/Jameco/Products/ProdDS/2193441QuickGuide.pdf>
 - [15] http://www.mon-club-elec.fr/pmwiki_reference_arduino/pmwiki.php?n=Main.Debuter
Presentation Logiciel
- http://wiki.mchobby.be/index.php?title=Arduino_Yun-Librairie
- https://www.ni.com/data_logger/

Résumé

Le projet réalisé est un système qui « monitorise » un ensemble d'échantillons immergés dans un électrolyte généralement une solution corrosive. Ce système permet de faire une mesure simultanée d'une différence de potentiel, entre une électrode de travail et une électrode de référence. Ces mesures permettent d'évaluer la capacité anticorrosive du matériau d'électrode utilisé.

Le système représente un data-logger à 8 voies qui est un dispositif électronique permettant d'acquérir des données à intervalles réguliers, il a été conçu à base d'une carte Arduino de type Yun. L'utilisation de cette carte est justifiée par ses différentes caractéristiques dont un lecteur de carte microSD intégré pour enregistrer les résultats et la possibilité de communiquer avec d'autres systèmes via les interfaces réseaux Ethernet ou Wifi.

Mots clés : data-logger, corrosion, acquisition, sauvegarde, ...