

Conception et réalisation d'un système de métrologie et supervision des paramètres météorologiques et électriques d'une installation photovoltaïque

Design and realization of a metrology system and supervision of the meteorological and electrical parameters of a photovoltaic installation

Mohammed Nfaoui¹, Khalil El-Hami¹

¹ Université Hassan 1^{er}, Faculté Polydisciplinaire de Khouribga, Laboratoire des Nanosciences et Modélisations, Khouribga, Maroc, nfaoui.smp@gmail.com, khalil.elhami@uhp.ac.ma

RÉSUMÉ. Le Maroc dispose d'un gisement solaire important compte tenu des valeurs de la durée d'insolation enregistrées sur la quasi-totalité du territoire national dépassant les 2000 heures annuellement et pouvant atteindre même les 3900 heures sur les hauts plateaux et le Sahara Marocain. Par conséquent, la quantité d'énergie reçue au sol dépasse en moyenne 5 kWh/m^2 dans plusieurs régions du Maroc soit près de 700 kWh/m^2 par an au nord et de 2263 kWh/m^2 au sud, ce qui favorise l'utilisation de l'énergie solaire dans différents domaines tel que la production de l'électricité, le séchage agroalimentaire, etc. la connaissance de ce gisement est d'un intérêt majeur pour concevoir et dimensionner les systèmes énergétiques solaires. Ainsi, un bon dimensionnement n'est possible que si les mesures sont disponibles d'une manière continue dans l'espace et dans le temps. La meilleure manière pour ce faire est de se disposer d'une série de mesures météorologiques et physiques issues d'un système d'exploitation Androïde, et un microcontrôleur qui est relié à des capteurs pour chaque mesure, les données recueillies sont envoyées à un serveur centralisé qui traite l'information.

ABSTRACT. Morocco has a significant solar deposit considering the values of the duration of insolation recorded on almost the whole national territory exceeding the 2000 hours annually and reaching even 3900 hours in the highlands and the Moroccan Sahara. As a result, the amount of energy received on the ground exceeds 5 kWh / m^2 on average in several regions of Morocco, ie nearly 700 kWh / m^2 per year in the north and 2263 kWh/m^2 in the south, which favors the use of electricity. solar energy in different fields such as electricity generation, agro-food drying, etc. the knowledge of this deposit is of major interest to design and size solar energy systems. Thus, good sizing is possible only if the measurements are available continuously in space and time. The best way to do this is to have a series of meteorological and physical measurements from an Android operating system, and a microcontroller that is connected to sensors for each measurement, the collected data is sent to a server centralized system that processes information.

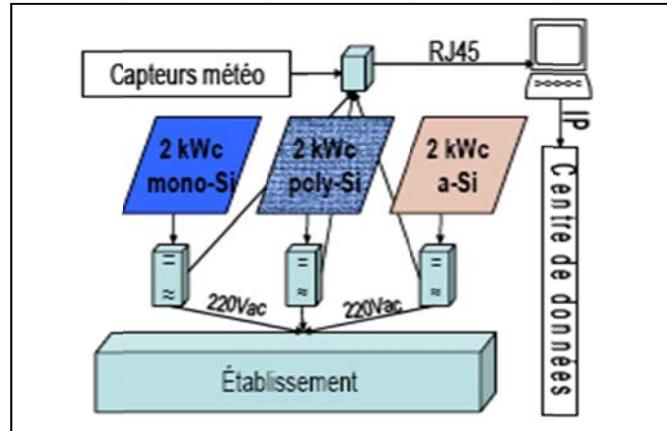
MOTS-CLÉS. Photovoltaïque, capteurs, Métrologie, L'acquisition des données, Transfert de données, système d'exploitation Android.

KEYWORDS. Photovoltaics, Sensors, Metrology, Data Acquisition, Data Transfer, Android Operating System.

1. Introduction

Pour le but de réalisation d'une carte des productivités d'une installation photovoltaïques, nous construirons une carte de productivité photovoltaïque. Sa calibration au sol pour l'utilisera des générateurs photovoltaïques identiques de (utilisant trois technologies silicium monocristallin, polycristallin et amorphe) connectés au réseau pour des objectifs spécifiques de :

- Développer la métrologie propre au notre laboratoire avec transfert de données à distance.
- Intégrer les mesures instantanées et en déduire les productivités mensuelles et annuelles.
- Proposer une méthode de correction et extrapolation aux données à long terme.
- Proposer une méthode pour l'étendre à l'ensemble du Maroc.
- Utiliser ceux-ci pour calibrer une carte des productivités du PV connecté au réseau.



Dans cet article, nous proposons une station métrologique de supervision et transfert des données pour les trois technologies photovoltaïques (monocristallin, polycristallin et amorphe) par l'utilisation d'un système d'exploitation Android.

Alors on commencera cet article par donner une idée générale sur le travail effectué ainsi que les outils et applications utilisées, avant de nous pencher vers une approche pratique.

2. Description générale de l'ensemble du système

Pour le principe de fonctionnement, un programme qu'on a développé est exécuté dans le microcontrôleur de la carte Arduino, qui permettant de récupérer toutes les données envoyées par les capteurs, avec un intervalle de temps défini. Une fois que les données sont reçues, elles seront transmises via Bluetooth à un Smartphone Android qui exécute une application Android qu'on a développé pour recevoir les données mesurées par chaque capteur et de les traiter si nécessaire.

Une fois les données reçues sont traitées, elles seront envoyées au serveur central qui loge une base de données.

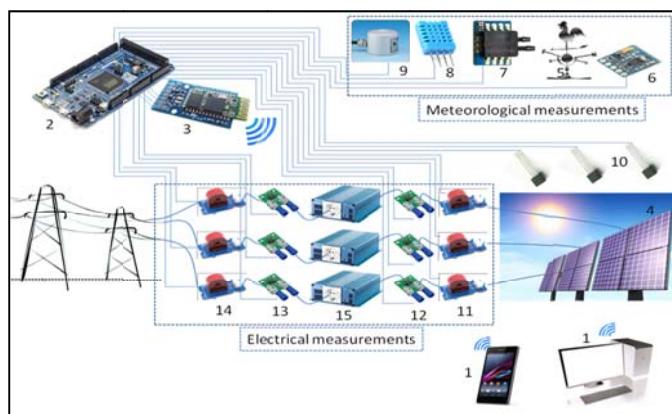


Figure 1. Schéma simplifié de l'ensemble du système

1 : système d'exploitation Android.

2 : carte Arduino.

3 : antenne Bluetooth.

4 : l'installation photovoltaïque.

5-9 : capteurs pour les mesures météorologiques.

10-14 : capteurs pour les mesures physiques.

15 : les onduleurs.

Les Mesures qui nous intéressent sont divisées en deux catégories :

Les variables météorologiques mesures.

- Mesure de la direction du vent.
- Mesure de la vitesse du vent.
- Mesure du rayonnement solaire.
- Mesure la température ambiante.

Les variables physiques mesures.

- Mesure les températures des modules.
- Mesure le courant continu de sortie des modules.
- Mesure la tension continue de sortie des modules.
- Mesure le courant alternatif de sortie des onduleurs.
- Mesure la tension alternative de sortie des onduleurs.

3. Métrologie et supervision

Le but de ce travail est de proposer une méthode de métrologique et supervision d'une installation photovoltaïques par l'utilisation d'un système d'exploitation Android. Les mesures des grandeurs électriques et météorologiques de l'installation peuvent être réalisées en utilisant des différents capteurs. L'acquisition de ces données est assurée à l'aide d'une carte Arduino basée sur un microcontrôleur.



Figure 2. Schéma simplifié des éléments utilisés

La carte Arduino est équipée d'une antenne Bluetooth permettant de transmettre les données mesurées à un terminal Android.

Avec un point d'accès 3G ou WiFi local, le Smartphone peut transmettre les données mesurées à la station de réception des données (serveur web).

Notre travail consiste en 4 étapes distinctes :

1. La collecte des données depuis les capteurs

2. La récupération de ces données sur la carte Arduino
3. Le transfert de ces données par une antenne bluetooth vers un système d'exploitation mobile Android
4. Le transfert les données récupéré par le système mobile vers la base de données de station de réception des données.

3.1. Montage des composants

L'instrumentation que nous avons utilisée, est l'ensemble des méthodes et moyens qui permettent sur un système donné de collecter des résultats de mesures des grandeurs physiques fournies par des capteurs, ceci de manière fiable, rapide, pertinente et économique. On peut également y inclure les procédures de traitement numérique des informations recueillies [1].

3.1.1. Carte Arduino :

Nous avons utilisé La carte Arduino Mega2560, cette carte dispose de toute une série de facilités pour communiquer avec un ordinateur [2].

Cette carte dispose :

- de 54 broches numériques d'entrées/sorties,
- de 16 entrées analogiques,
- de 4 UART (port série matériel),
- d'un quartz 16Mhz,
- d'une connexion USB,
- d'un connecteur d'alimentation jack,
- d'un connecteur ICSP,
- et d'un bouton de réinitialisation.

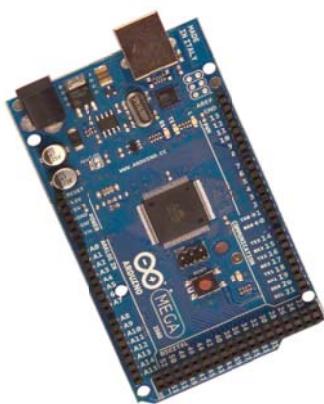


Figure 3. Vue de la carte Arduino Mega 2560

3.1.2. Les capteurs :

Pour mesurer tous les types de grandeurs physiques pour les traiter et les exploiter, on doit transformer la grandeur à mesurer en un signal facilement exploitable : une tension ou un courant électrique.

Afin d'optimiser notre travail et récolter le maximum de données possibles, nous avons procédé à l'essai des capteurs suivants :

- Capteurs de mesures météorologiques

Nous proposons également des produits et accessoires météorologiques, tout-en-un qui mesure intensité du rayonnement solaire, la vitesse et la direction du vent et la température ambiante.

- Capteurs de mesures électriques

Deux capteurs pour mesurer les grandeurs électrique, un capteur permet de mesurer l'intensité du courant électrique et l'autre pour mesurer la tension.



Figure 4. Les capteurs utilisés pour la mesure des paramètres météorologiques et électrique

Tous les capteurs analogiques et numériques cités ci-dessus. Chaque capteur comprend au moins 3 fils de branchement :

3.1.3. Branchement d'un capteur analogique

La carte Arduino MEGA 2560 comporte 15 entrées analogiques : A0, A1, ..., A15.

Les trois fils des capteurs analogiques sont à brancher de la façon suivante :

- fil rouge 5V capteur : 5V (alimentation) Arduino,
- fil noir GND capteur : GND (terre) Arduino,
- fil restant: ENTREE ANALOGIQUE (A0 à A15).

3.1.4. Branchement d'un capteur numérique

Arduino contient 54 entrées numériques.

Les capteurs numériques sont à brancher de la façon suivante:

- fil rouge 5V capteur : 5V (alimentation) Arduino
- fil noir GND capteur : GND (terre) Arduino
- fil vert sortie capteur : entrée numérique (digital) Arduino

3.1.5. Système d'exploitation Android

Un système d'exploitation mobileb est un système d'exploitation conçu pour fonctionner sur un appareil mobile. Ce type de système d'exploitation se concentre entre autres sur la gestion de la connectivité sans fil et celle des différents types d'interface.



Figure 5. L'Appareils mobiles Smartphones utilisé (Samsung trend plus)

Dans le guide du développeur, Android est défini comme étant une pile de logiciels, c'est-à-dire un ensemble de logiciels destinés à fournir une solution clé en main pour les appareils mobiles Smartphones .

Cette pile comporte un système d'exploitation (comprenant un noyau Linux), les applications clés telles que le navigateur web, le téléphone et le carnet d'adresses ainsi que des logiciels intermédiaires entre le système d'exploitation et les applications.

3.1.6. Mettre en pratique

Nous avons vu que l'utilisation d'Arduino n'est pas vraiment compliquée, il suffit de bien brancher les capteurs et passer après à la partie programmation.

Voici l'ensemble du schéma électrique de notre système :

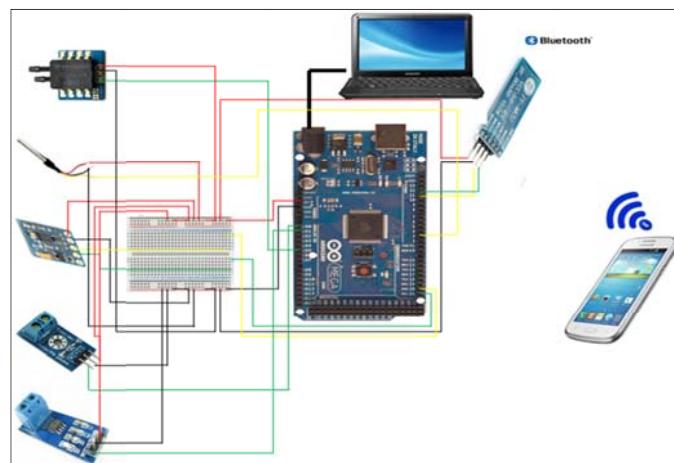


Figure 6. Montages réelle des différents capteurs à la carte Arduino

3.2. Partie de programmation réalisée

Pour ce travail nous avons créé un programme spécifique que nous avons téléchargé dans le microcontrôleur pour que la carte permettant de commander les différents composants et récupérer les mesures envoyé par les capteurs, puis de transmettre ces données mesurées à un Smartphone ensuite envoyé les données traité au serveur central via le réseau 3G.

3.2.1. Organigramme :

Comme illustré par l'organigramme de la figure ci-dessous, le programme élaboré en langage proche du C++ pour piloter la carte Arduino s'exécute en plusieurs étapes ;

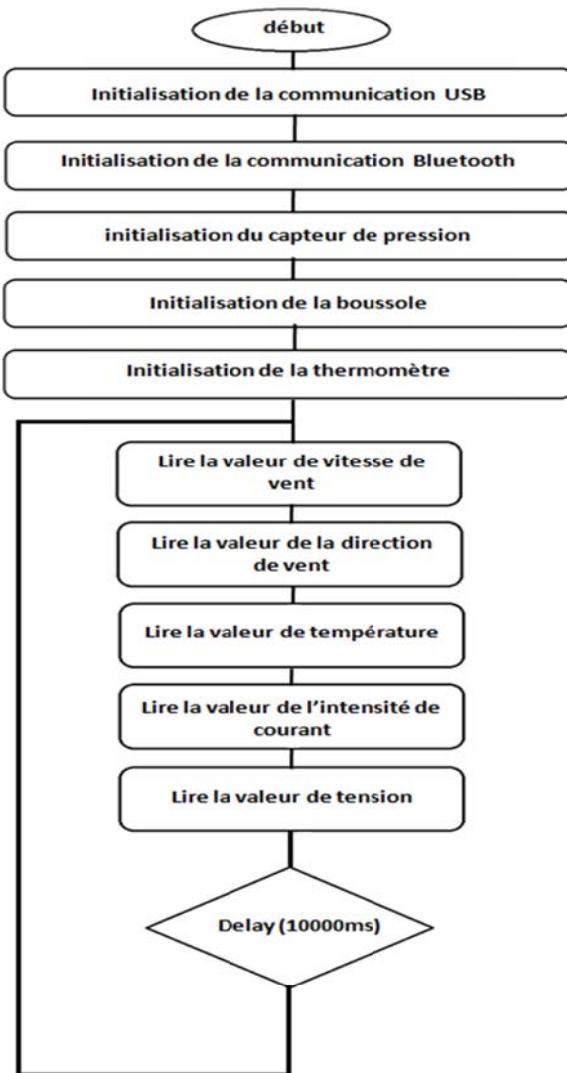


Figure 7. L'organigramme que va suivre notre programme de carte Arduino

Maintenant, il ne reste plus qu'implémenter la solution expliquée dans l'organigramme.

Notre travail consiste en 4 étapes distinctes :

3.2.2. Programme Arduino

Un programme Arduino est un fichier de code écrit en langage proche du C++. Une fois compilé et transféré vers la carte Arduino (via USB), on peut accéder aux résultats du programme via le terminal incorporé au logiciel [2].

Partie déclarative

- Inclusion des librairies utilisées

Les librairies peuvent être incluses dans notre programme à l'aide de l'instruction `#include` et nous permettent d'utiliser des fonctions élaborées avec notre carte Arduino (communication série par exemple). Les librairies nous permettent surtout d'interfacer et d'utiliser notre carte Arduino avec toutes sortes de matériel (afficheur LCD, clavier matriciel, etc...).

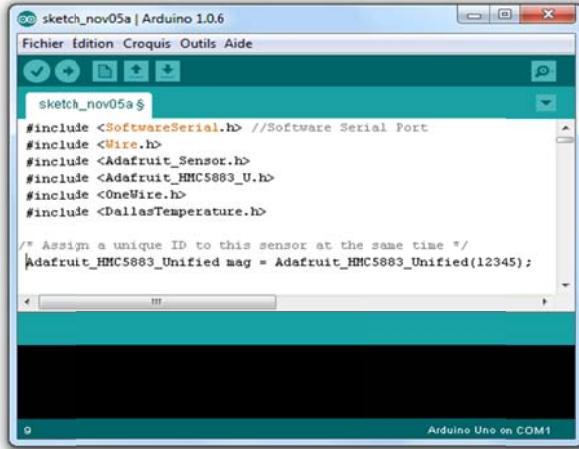


Figure 8. Interface des librairies incluses pour notre programme Arduino

– Déclaration des constantes et variables globales

Une variable est un espace de stockage nommé qui permet de stocker une valeur utilisable par la suite dans la boucle d'un programme. Une variable peut aussi bien représenter des données lues ou envoyées sur un des ports analogiques ou numériques, une étape de calcul pour associer ou traiter des données, que le numéro 'physique' de ces entrées ou sorties sur la carte. Une "variable" n'est donc pas exclusivement un paramètre variant dans le programme. Alors pour composer un programme Arduino, il est nécessaire de définir toutes les composantes d'entrée et de sortie qui vont affecter le montage matériel et les calculs à effectuer. Toutes les entrées et sorties seront des variables.

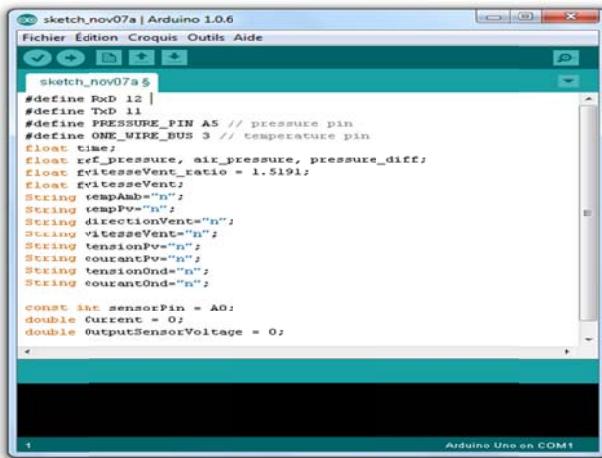


Figure 9. Interface Déclarative du logiciel Arduino

La fonction d'initialisation `setup()` et la boucle principale de la fonction `loop()`

Avec Arduino, nous devons utiliser un code minimal lorsque l'on crée un programme. Ce code permet de diviser le programme que nous allons créer en deux grosses parties.

```

void setup()      //fonction d'initialisation de la carte
{
    //contenu de l'initialisation
}

void loop()       //fonction principale, elle se répète
(s'exécute) à l'infini
{
    //contenu de votre programme
}

```

Fonction `setup()` : est appelée une seule fois lorsque le programme commence. C'est pourquoi c'est dans cette fonction que l'on va écrire le code qui n'a besoin d'être exécuté une seule fois. On appelle cette fonction : "fonction d'initialisation". On y retrouvera la mise en place des différentes sorties et quelques autres réglages.

Fonction `loop()` : où l'on va écrire le contenu du programme. Il faut savoir que cette fonction est appelée en permanence, c'est-à-dire qu'elle est exécutée une fois, puis lorsque son exécution est terminée, on la ré-exécute et encore et encore. On parle de boucle infinie.

3.2.3. *Récupération des données depuis les capteurs :*

Nous devons récupérer les données envoyées par les capteurs de manière régulière. Pour cela nous enverrons les valeurs mesurées depuis les capteurs en toutes 10 secondes [3].

Dans notre travail, la carte Arduino lit les valeurs relevées par les différents capteurs analogiques grâce à la fonction `analogRead()`. Ces valeurs ensuite converties en voltage grâce à des formules correctives de chaque capteur.

Il suffira ensuite de mettre les codes dans une boucle `loop()` pour recevoir ces données après un délai (10 secondes).

Il faut savoir également qu'on peut récupérer plusieurs données (de différents capteurs) durant le même envoi de données. Pour cela, il suffit de concaténer ces données sous forme de chaîne de caractères à envoyer après un délai fixe de 10 secondes.

- Mesure de la vitesse du vent.
- Mesure de la direction du vent.
- Mesure la température ambiante.
- Mesure le courant de sortie des modules.
- Mesure la tension de sortie des modules.

Une fois les valeurs mesurées acquises, il faudra désormais l'envoyer vers notre système d'exploitation Android. Pour cela, nous allons utiliser une antenne Bluetooth qui permet de transférer les valeurs reçues par la carte Arduino.

4. Le transfert les données par un module Bluetooth

Le module Bluetooth utilisé permet l'émission et la réception de données par l'intermédiaire d'une liaison Bluetooth. La communication entre le module et le microcontrôleur de la carte Arduino est réalisée par une interface série.

Ce module est l'assemblage d'une carte Bluetooth générique sur une carte simplifiant l'interface. A partir de cette interface la connexion à l'Arduino est basique :

- GND (carte Bluetooth) ⇒ Arduino GND
- VCC (carte Bluetooth) ⇒ Arduino 5V
- TX-O (carte Bluetooth) ⇒ Arduino RX (0)
- RX-I (carte Bluetooth) ⇒ Arduino TX (1)



Figure 10. Carte Bluetooth JY-MCU

Le module Bluetooth est constitué d'un émetteur-récepteur Bluetooth appartenant à la classe 2. Il est également constitué d'un dispositif de régulation lui permettant de fonctionner avec une tension d'alimentation comprise entre 3,3 V et 6 V.

4.1. Brochage

Le brochage de notre module Bluetooth est le suivant :

- VCC : Tension d'alimentation devant être comprise entre 3,3 V à 6V.
- GND : Tension de référence 0V.
- TX-O : Transmission des données. La broche TX-O est utilisée pour transmettre les données séries au microcontrôleur.
- RX-I : Réception des données. La broche RX-I est utilisée pour recevoir les données séries émises par le microcontrôleur.

4.2. Programme Arduino

La communication entre le module Bluetooth et le module Arduino est réalisée par l'intermédiaire du port série asynchrone accessible via les E/S numériques 11 (ligne Rx) et 12 (Ligne Tx).

Nous utiliserons donc la bibliothèque « SoftwareSerial » pour recevoir ou transmettre les données.

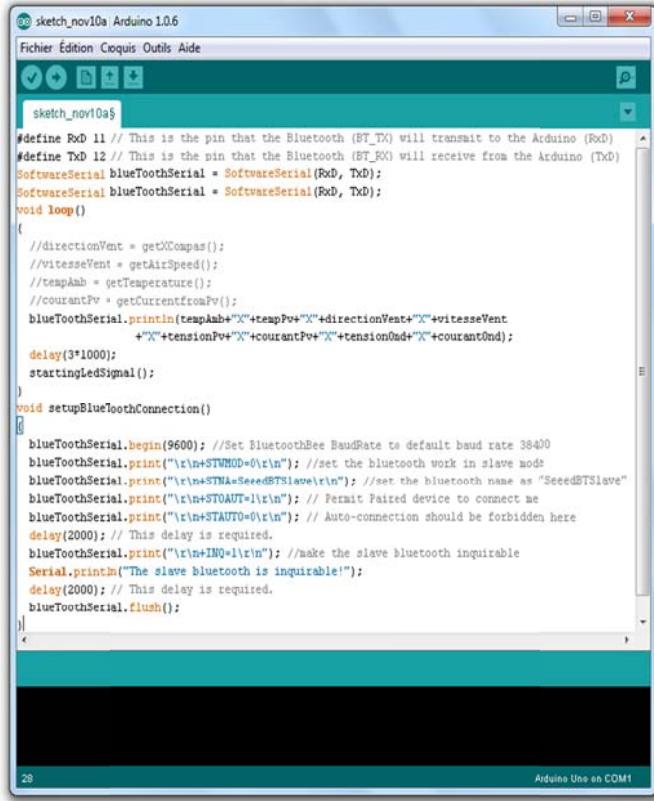


Figure 11. Interface du logiciel Arduino utilisé pour Le module Bluetooth

Ensuite Nous allons utiliser l'éditeur Eclipse pour créer une application Android qui permettra de communiquer par Bluetooth avec une carte Arduino via le module Bluetooth.

5. Réception des données depuis le Smartphone

Une fois les valeurs envoyées à partir d'Arduino au Smartphone Android, ce dernier vérifie si les données reçues répondent à la structure du protocole de communication utilisé, ensuite il fait l'extraction des données de chaque capteur et vérifie s'il est logique, et enfin il envoie les données vers le serveur via la connexion 3G.

Pour cela, nous allons utiliser dans notre programme l'ensemble des fonctions permettant d'effectuer un ensemble d'instructions par simple appel de la fonction dans le corps du programme principal.

Les fonctions permettent d'exécuter dans plusieurs parties du programme une série d'instructions, cela permet une simplicité du code et donc une taille de programme minimale.

Les fonctions que nous avons utilisées :

`protected void onCreate(Bundle savedInstanceState)` : fonction principale, elle est exécutée automatiquement au démarrage de l'application ;

`private void initMain()` : création de l'interface graphique ;

`private void checkBt()` : fonction qui vérifie si le Smartphone est connecté avec l'antenne Bluetooth, et si cette dernière est activée ;

`public void connect()` : fonction qui permet d'établir la connexion avec la carte arduino ;

`public void beginListenForData()` : cette fonction permet de recevoir les données envoyées par Bluetooth (elle écoute l'antenne) ;

`public boolean isConnected()` : fonction qui vérifie si le Smartphone est connecté à internet ou non ;

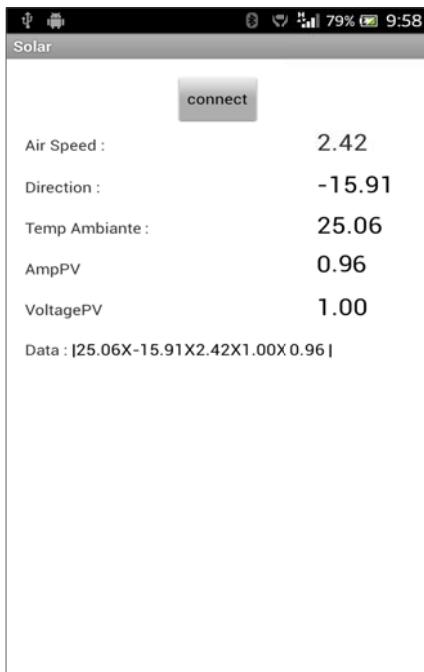
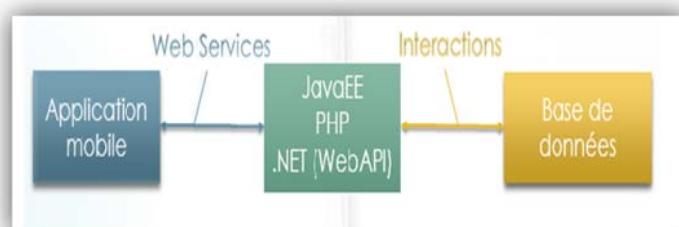


Figure 12. Interface Android pour la consultation des données envoyées par Le module Bluetooth

Le résultat final est une liste des données électriques et météorologiques mesurées par les capteurs.

7. Transfert les données vers la base de données de station centrale

Dans une ultime étape Nous proposons une application Android qui permet de transférer les valeurs reçues par le Bluetooth de Smartphone à un serveur de station centrale de réception des données via un réseau internet (dans notre cas c'est le réseau 3G) [4].



Après l'envoie des valeurs reçues par android vers le serveur, un script PHP récupère les données envoyées par le Smartphone, et ouvre une connexion avec la base de données de station centrale de réception des données MySql, et insert les données dedans.

Cette partie consiste en plusieurs étapes distinctes :

1. Extraction des données reçues depuis le Smartphone.
2. Construction du lien http portant les mesures des capteurs.
3. Réception des valeurs par le serveur de la base de données.
4. Insertion des données dans la base de données.

Les données que nous avons envoyées sont ordonnées sous forme de « tables », c'est-à-dire par regroupements de plusieurs valeurs sous forme d'un tableau.

La table comporte autant de champs que de capteurs, en plus d'un identifiant unique et de la date et l'heure de la mesure.

Le résultat final est une liste des données enregistrées.

8. Conclusion

Ce travail pour nous, était une occasion de nous approfondir dans le monde de la programmation des systèmes embarqués, et une opportunité de tester les différents capteurs disponibles au laboratoire de façons différentes.

Notre travail s'est déroulé en 3 étapes distinctes : On a d'abord commencé par nous familiariser avec le sujet, à savoir les différents capteurs, leurs modes de communication et l'environnement Android, avant de passer à la pratique et étudier chaque outil, d'abord la carte Arduino, pour enfin concevoir une application Android capable de gérer toutes ces données, chacun apportant les connaissances qu'il a ainsi pu acquérir auprès de l'outil qu'il étudiait.

References

- [1] Marobot .com, 'fournisseur de l'innovation Marocaine', <http://store.marobot.com/> 2016
- [2] Astalaseven, Eskimon et olyte "Arduino pour bien commencer en électronique et en programmation" www.siteduzero.com, 2016
- [3] Astalaseven ,Eskimon et olyte"Arduino pour bien commencer en électronique et en programmation" pp 179-193 2016.
- [4] L.Quentin et T M'hamed"Interface Android pour la consultation de données envoyées par un capteur", Rapport du Travail d'Etude et de Recherche du Master 1 Ingénierie Logiciel, Université de Bretagne occidentale, 2011.