

UNIVERSITE KASDI MERBAH OUARGLA

Faculté des Sciences Appliquées

Département de Génie Electrique

جامعة كاسدي مرباح ورقلة

Université Kasdi Merbah Ouargla



Mémoire

MASTER ACADEMIQUE

Domaine : Sciences et technologies

Filière : Génie électrique

Spécialité : Electrotechnique Industrielle

Présenté par :

Rabhi Chaima

Thème:

Interface de mesure et affichage des paramètres
internes et externes d'un panneau photovoltaïque
sous ANDROID

Soutenu publiquement

Le :08/07/2019

Devant le jury :

M^r Kouras Sid Ali

MCA

Président

UKM Ouargla

M^r Rezoug Mohamed Redha

MCB

Encadreur/rapporteur

UKM Ouargla

M^r Bouchala Tarek

MCA

Examineur

UKM Ouargla

Année universitaire 2018/2019

بسم الله الرحمن الرحيم

Remerciement

Tout d'abord je tiens à remercier ALLAH le tout puissant de m'avoir donné le courage et la force de terminer ce travail épineux.

Comme je tiens aussi à remercier mon encadreur Mr Rezoug Mohamed Redha pour son encadrement et son soutien tout le long de ce travail, et qui ne m'a pas ménagé d'aucun effort de m'aider et m'orienté, que Dieu le garde et enrichi ses connaissances pour la science et jouir de sa santé pour son bien-être.

Je voudrai aussi remercier les membres de jury d'avoir fait l'honneur d'accepter le jury de ce travail.

Je souhaite également remercier tous les enseignants de département de génie électrique

Je souhaite remercier mes chères parents qui se sont montrés très présents, mes frères et soeurs Djalal, Abdaldjabar, Zakaria, Fouzi, Samira, Zahra et toute la famille Rabhi et Khalifa

Et mes amis

Chapitre I

Figure I.1 Fenêtre d'accès au App Inventor	4
Figure I.2 Fenêtre de démarrage du App Inventor	5
Figure I.3 L'interface d'accueil du App Inventor 'partie design'	5
Figure I.4 Imprime écran ; (a) icône App inventor sous un smart phone, (b) interface de connectivité sous smart phone	6
Figure I.5 Générateur de code de connectivité sous PC	6
Figure I.6 Protocole de connexion code barre	7
Figure I.7 Bouton d'accès aux fenêtres blocs	8
Figure I.8 Fenêtre de la programmation par bloc	8
Figure I.9 Allure de l'application sous smart phone	9

Chapitre II

Figure II.1 Description de la carte Arduino Mega 2560	13
Figure II.2 Schéma synoptique d'un capteur	14
Figure II.3 Structure interne du DHT22	17
Figure II.4 (a) Le Boîtier de DHT22 ; (b) Schéma de connexion de DHT22 avec l'arduino	17
Figure II.5 Conception interne du capteur BH 1750	18
Figure II.6 (a) Le Boîtier de capteur BH 1750 ; (b) Schéma de connexion de BH 1750 avec l'Arduino	18
Figure II.7 Schéma explicatif du capteur	19
Figure II.8 (a) Le boîtier Capteur ACS 712 ; (b) Schéma de connexion de ACS 712 avec l'Arduino	20
Figure II.9 (a) Le boîtier capteur B25 ; (b) Schéma de connexion de B25 avec l'Arduino	20
Figure II.10 (a) HC-06 Bluetooth ; (b) Schéma de connexion de HC-06 avec l'arduino	22
Figure II.11 L'affichage des valeurs récoltées sur le terminal du PC	22
Figure II.12 Dispositif générale réalisé sur CAO	23
Figure II.13 Dispositif général	23

Chapitre III

Figure III.1 Zone de déclaration	26
Figure III.2 Configuration initiale	27
Figure III.3 Fonction principale loop	28

Figure III.4 Boucle d’affichage	28
Figure III.5 Fonction de vérification de DHT	29
Figure III.6 Fonction de Seuil	29
Figure III.7 Boucle de mesure de tension	29
Figure III.8 Tracée ‘Humidité’	30
Figure III.9 Tracée ‘Température	30
Figure III.10 Tracée ‘Tension du panneau’	31
Figure III.11 Tracée ‘Intensité de Lumière’	31

Chapitre I

Tableau I.1 Signification des blocs de notre application	11
--	----

Chapitre II

Tableau II.1 Caractéristiques techniques du DHT 22	16
Tableau II.2 Caractéristiques techniques du capteur BH 1750	18
Tableau II.3 Caractéristiques techniques du ACS 712	19
Tableau II.4 Tableau de commande du Bluetooth	21

Chapitre III

Tableau III.1 Comparaison des moyennes	32
--	----

Sommaire

Liste des figures	I
Liste des tableaux	II

Introduction Générale

1. Généralités	1
2. Position du problème	1
3. Objectif du projet	2
4. Présentation du mémoire	2

Chapitre I

App Inventor comme plate forme de développemnt

I.1 Introduction	3
I.2 Système D'exploitation	3
I.3 Android comme système D'exploitation	3
I.4 Programmation orientée objet	4
I.4.1 Qu'est-ce qu'un App Inventor	4
I.4.2 Pas à Pas Vers App Inventor	4
I.4.3 Démarrage de l'AICompanion sur L'appariel	5
I.4.4 Connexion de l'App Inventor sur le Smart Phone pour Test	6
I.4.5 Visualisation de L'application sur L'appareil Connecté	7
I.4.6 Accéder aux Editeur de Blocs	7
I.4.7 Editeur de Blocs	8
I.5 Programmation par Bloc sous App Inventor	9
I.6 Conclusion	11

Chapitre II

Réalisation d'un dispositif Expérimental

II.1 Introduction	12
II.2 Système embarqué	12
II.2.1 La carte Arduino	12
II.2.2 Téléphone intelligent	13
II.3 Définition d'un Capteur	13
II.3.1 Classification des Capteurs	14
II.3.2 Evolution des Capteurs	14
II.4 Grandeur à Mesure	14
II.4.1 Grandeur éclairément	14
II.4.2 Grandeur Humidité	15

II.4.3 Grandeur Température	15
II.4.4 Grandeur Tension	15
II.4.5 Grandeur Courant	15
II.5 Unités de mesure	16
II.5.1 Le capteur DHT 22	16
II.5.2 Capteur de Lumière BH 1750	17
II.5.3 Le capteur ACS 712	19
II.5.4 Le capteur B25	20
II.6 Protocole de communication	21
II.7 Assemblage du dispositif	23
II.8 Conclusion	24

Chapitre III

L'environnement de développement IDE de l'Arduino

III.1 Introduction	25
III.2 L'environnement IDE de l'Arduino	25
III.3 Structure de programme	25
III.3.1 Partie déclaration	26
III.3.2 Partie Fonctions	27
III.4 Mesures et tests expérimentaux	29
III.5 Conclusion	32

Conclusion Générale

1. Généralités	33
2. Problèmes rencontrés	33
3. Perspectives du Projet	33

Annexes

Annexe A	34
Annexe B	35

Références

Introduction Générale

1. Généralités :

De nos jours les systèmes embarqués ainsi que les systèmes intelligents ne cessent de croître en quantité et en qualité. Dû à l'obligation des besoins du marché, à la concurrence, à la diversité des produits pour satisfaire les besoins des utilisateurs avec une certaine optimisation. Ces derniers sont exposés à une certaine évolution de l'environnement à l'intermédiaire des entrées et sorties et les différents moyens de communication ainsi que les interfaces électroniques de traitement.

Aujourd'hui les nouvelles technologies et la disponibilité des techniques sans fils nous permettent de créer des nouvelles dispositifs dans plusieurs domaines tel que : l'industrie, l'agriculture, l'électrotechniques ... etc. Parmi eux, la commande domotique, les instruments intelligents qui sont déjà opérationnels sous plusieurs applications.

L'énorme flux de la vente des smart phones, des tablettes et les instruments domotiques se fait en même temps comme adaptation rapide des moyens de communications et la connectivité tels que: Wifi, Bluetooth, infra rouge .etc. nous ont conduit de travailler sur ce qui est considéré comme système dans leurs noyaux.

Android est sûrement la plate-forme la plus utilisée pour les équipements domotiques. et à marquer sa présence même dans l'industrie comme système d'interfaçage de mesure et affichage sur plusieurs instruments.

Afin de répondre à cette évolution, nous allons essayé de travailler sur un domaine connu comme sujet de recherche tel que les systèmes photovoltaïques. Nous réaliserons une carte d'interfaçage à l'aide d'une Arduino qui mesurera à tout moment les paramètres qui contournent l'installation d'un panneau photovoltaïque par l'intermédiaire d'un smart phone via la connectivité Bluetooth servant comme moyen d'affichage et offrir au public un moyen d'utilisation comme base de données.

2. Position du problème :

La surveillance et le contrôle des grandeurs physiques quelques soient leurs natures une de plus grandes problématiques de nos jours nous oblige toujours à mettre tous nos moyens pour aboutir à une précision de mesure. Les techniques d'affichage sont le moyen de présentation d'une information au moyen de divers phénomènes physiques ou chimiques qui sont illustrées récemment à l'aide des applications portables surtout sous smart phone.

Dans une installation photovoltaïque le changement climatique brusque implique une surveillance permanente en temps réel afin de mieux contrôler le système PV.

3. Objectif du projet :

Dans ce projet trois objectifs ont été visés :

- Comme premier objectif dans ce travail est de se familiariser avec un environnement connu qui est App Inventor de Google et de créer une application qui porte l'extension d'installation APK sous le système d'exploitation Android. Cette application affiche toutes les grandeurs qui rentrent comme paramètres entourant le panneau solaire.
- Notre deuxième objectif est de s'adapter à un système embarqué nommé Arduino ainsi que quelques modules qui nous servent comme capteurs.
- Le troisième but consiste à réaliser une carte électronique capable de mesurer à l'aide de quelques capteurs les grandeurs suivantes (température, humidité, intensité de lumière, tension du sortie du panneau ...etc) et les transmettre au smart phone à travers un module Bluetooth en expliquant les différents blocs de sa construction.

4. Présentation du mémoire :

Ce mémoire est structuré par une introduction générale suivie par trois chapitres qui traitent comme suit:

- Dans le premier chapitre, nous avons donné des généralités et des définitions sur les systèmes d'exploitations et les systèmes embarqués comme première partie. Nous avons consacré la deuxième partie aux détails des différentes étapes pour travailler sous un tel environnement qui gère les applications Android dont on a procédé de réaliser un programme d'affichage et mesure sous un environnement MIT de Google. Ce programme est considéré comme un soft primaire pour notre projet.
- Le deuxième chapitre illustre la conception électronique du dispositif qui se base essentiellement sur une carte Arduino Mega où on détaillé tous les blocs de conception ainsi que les divers capteurs qui rentrent comme des modules de mesure pour les différentes grandeurs (température, humidité, éclairement, voltage et courant de panneaux).
- Dans le troisième chapitre, nous exposons les différentes étapes du soft secondaire du notre projet qui présente le développement d'un programme sous l'environnement IDE de l'Arduino, ainsi que quelques résultats pratiques afin de démontrer la démarche logique de ce programme qui travaille simultanément avec notre application Android.

Finalement, on achève notre travail par une conclusion générale qui englobe la synthèse des trois chapitres et de laisser un champs de travail comme perspectives d'avenir.

Chapitre I

App Inventor comme plate

forme de développement

I.1 Introduction

Parmi les systèmes d'exploitation qui sont généralement utilisés dans notre vie quotidienne, on trouve Windows avec ses versions, Linux et ses gammes et Android avec ses améliorations. Ce dernier occupe une large place comme systèmes fonctionnels dans les équipements électroniques et est favorisé par les constructeurs précisément pour les instruments domotiques (smart phones, téléviseurs, caméras, imprimantes...etc.). la quasi totale des personnes possède au moins un smart phone ce qui nous a poussé à sélectionner ce système.

Pour les informaticiens, cette apparition a ouvert un grand domaine de travail afin de créer des environnements qui nous servent comme outils de programmation. La disponibilité de ces environnements nous a laissé le choix d'utiliser le plus avantageux, le moins coûteux et le plus exploité dont; il s'agit de l'environnement App Inventor de Google.

Le but de ce chapitre est d'éclaircir la partie soft qui a une relation avec le programme qui porte l'extension APK ainsi que le "notre" qui va nous servir comme une interface d'affichage pour les grandeurs entourant le panneau photovoltaïque.

Tous les éléments qui ont été utilisés sous l'environnement App Inventor pour réaliser cette application ont été placés dans ce tableau avec leur propre rôle indiqué.

I.2 Système D'exploitation

Un système d'exploitation abrégé parfois SE, (ou Operating System dit OS en anglais) est l'ensemble de programmes qui gouvernent les différents composants de l'appareil informatique (écran, disque dur, mémoire, processeur...etc) et lui permet de fonctionner.

Il est chargé en premier lieu lors du démarrage puisqu'il est l'interface principale entre l'utilisateur et le matériel informatique. Il permet aussi de faire fonctionner les différents périphériques (carte graphique, carte son, clavier, souris...etc).

L'utilisation des pilotes (drivers en anglais) permet au système d'exploitation de communiquer et de donner des instructions aux périphériques connectés. On trouve un système d'exploitation pré installé sur tous les types d'appareils informatiques : tablette tactile, ordinateur personnel, smart phone, console de jeux...etc [1].

I.3 Android Comme Système D'exploitation

Android est le système d'exploitation mobile fondé sur le noyau Linux et développé actuellement par Google en juin 2007. Il équipe actuellement la plupart des smart phone (téléphones intelligents). Au commencement, le système a été conçu pour les smart phones et

les tablettes, ensuite il s'est diversifié dans plusieurs objets connectés tels que les ordinateurs les télévisions (Android TV), les voitures (Android Auto), les Chromebook, les smartwatch (Wear OS).

Android est un système qui nous facilite le contrôle de nos objets, il gère le téléchargement, la navigation, la commande des capteurs...etc.

I.4 Programmation orientée objet

La programmation orientée objet est une méthodologie de programmation récente utilisée par la popularité des programmes connus (Java, C++, C#, Net, ...) [2].

L'utilisation de la notion d'objet par la programmation orientée rassemble toutes les variables ainsi que les opérations à l'aide d'une illustration graphique; cela signifie aussi qu'il y a une manipulation des objets informatiques "Classes" qui possèdent des propriétés dont ils peuvent avoir leurs propres héritiers qui ont leurs méthodes.

I.4.1 Qu'est-ce qu'un App Inventor

App Inventor est un outil de développement en ligne destinée aux téléphones et aux tablettes qui fonctionnent sur le système d'exploitation Android. Il a besoin d'une liaison Internet et d'un navigateur Internet comme Mozilla Firefox ou Chrome. C'est une application open source web fournie par Google, maintenue actuellement par l'Institut de technologie du Massachusetts (MIT). Elle est accessible à chaque utilisateur disposant d'un compte Gmail.

I.4.2 Pas à Pas Vers App Inventor

1. On se connecte sur App Inventor avec notre nom d'utilisateur et le mot de passe de Gmail de (Google).
2. On se connecte directement sur ai2.appinventor.mit.edu où on clique sur le bouton orange "Create" du site web d'App Inventor (appinventor.mit.edu). La figure ci dessous présente le premier accès à cet environnement de programmation.



Figure I.1 Fenêtre d'accès au App Inventor

En cliquant sur "Continuer", le site nous mènera à un écran d'accueil.



Figure I.2 Fenêtre de démarrage du App Inventor

Quelques obligations doivent être respectées afin de créer un nom pour le projet (les tirets bas sont autorisés mais pas les espaces) et on valide par OK. Une fois dans l'espace Designer, il est possible d'afficher l'interface utilisateur de l'application [3].

Avec Design Windows, ou "Designer", on peut visualiser l'apparence de notre application et sélectionner les composants qui doivent être dans notre application. Nous pouvons aussi choisir des options pour l'interface utilisateur tels que (Text Boxes, Buttons, Images et des fonctionnalités tels que GPS, Text-to -Speech et Sensors...etc).

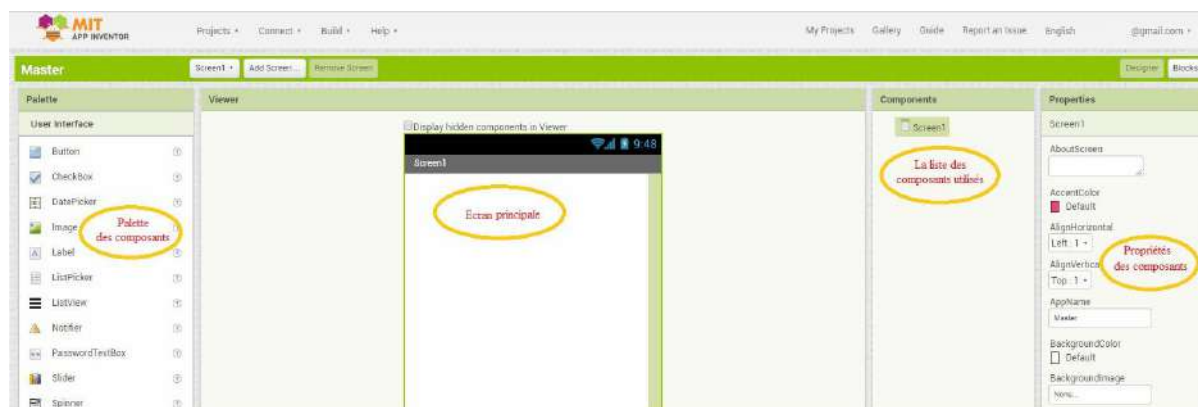


Figure I.3 L'interface d'accueil du App Inventor "partie design"

- Menu principal (enregistrer, simuler, ...).
- Menu pour choisir les projets (ils sont sauvegardés sur les serveurs du MIT).
- Choix de la langue.
- Menu pour choisir le mode "designer" (pose des éléments de l'interface) et le mode "blocks" (là où on fait le programme).
- Fenêtre contenant les éléments qu'on peut choisir (bouton, texte, ...).
- Fenêtre principale représentant l'écran du smart phone; c'est sur cet écran qu'on va placer les éléments qui nous intéressent.
- Fenêtre de paramétrage de l'élément sélectionné.

I.4.3 Démarrage de l'AICompanion sur L'appareil

Depuis le smart phone ou la tablette, on clique sur l'icône qui permet à MIT AI Companion de démarrer l'application. Pendant cela, le smart phone et l'ordinateur doivent être

connectés sur le même réseau. Si WiFi est inaccessible, des instructions de la connexion USB seront disponibles sur le site web [4].

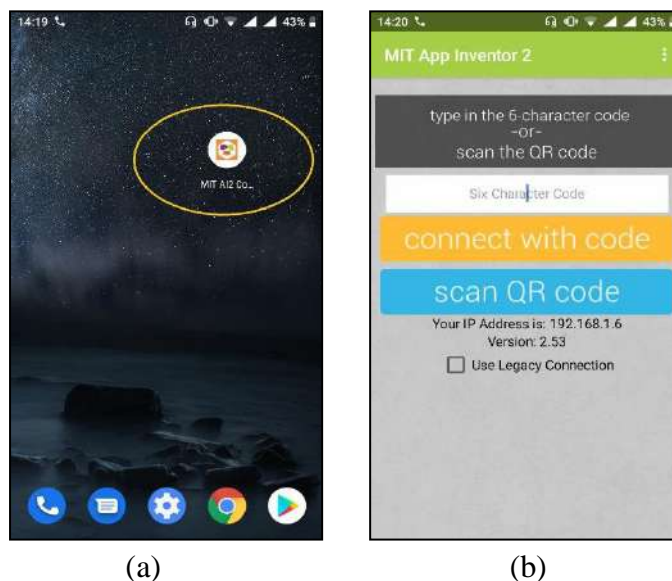


Figure I.4 Imprime écran; (a) icône App Inventor sous un smart phone, (b) interface de connectivité sous un smart phone

I.4.4 Connexion de l'App Inventor sur le Smart Phone pour Test

Une des fonctions les plus intéressantes de App Inventor c'est qu'on peut visualiser et tester l'application sur un appareil connecté pendant la construction. Le test réel est effectué comme suit.

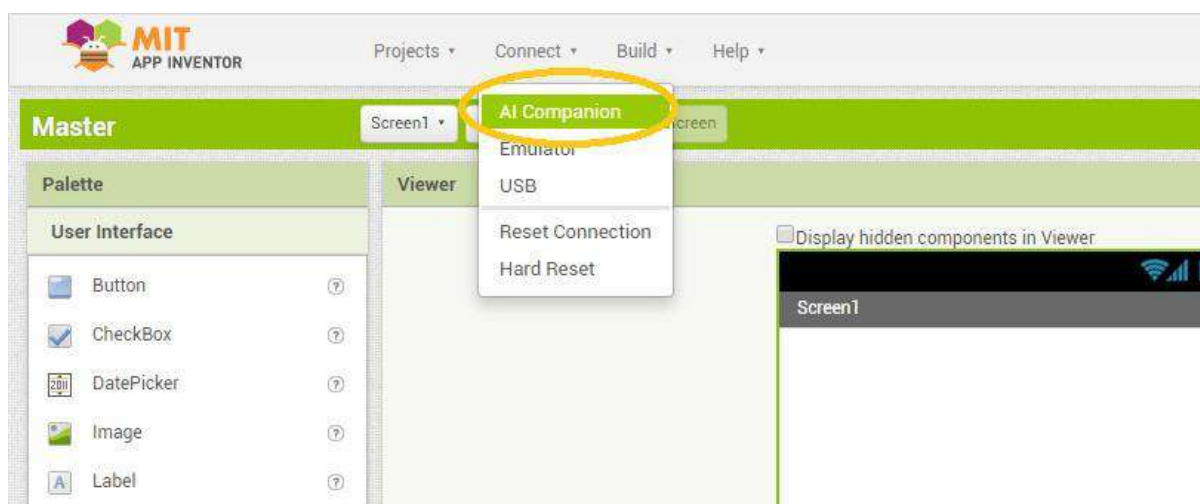


Figure I.5 Générateur de code de connectivité sous PC

On obtient le code de connexion de six caractères, on le scanne ou on le tape sur Companion App. Sur le menu connecté, on sélectionne "AI Companion". On peut se connecter en :

- 1 - Scannant le code QR et en cliquant sur «Scan QR code» (#1)
- 2 - En tapant le code dans la fenêtre de texte puis on clique sur "Connect with code" (#2)

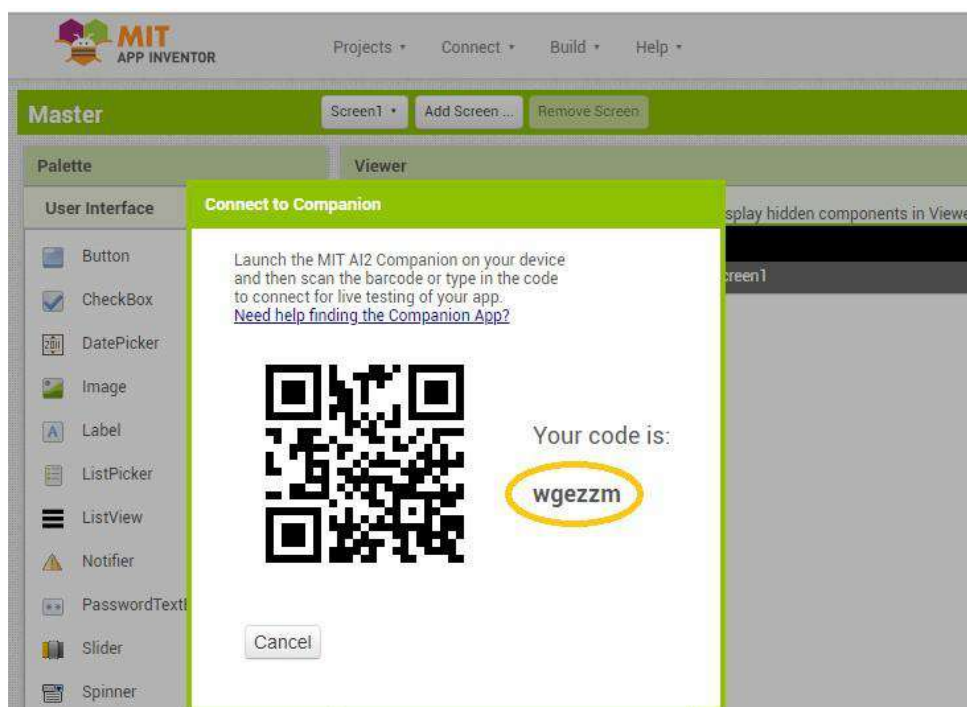


Figure I.6 Protocole de connexion code barre

I.4.5 Visualisation de L'application sur L'appareil Connecté

On peut vérifier le fonctionnement de la connexion en visualisant notre application sur l'appareil connecté. L'application sur le smart phone change avec la progression du projet.

I.4.6 Accéder aux Editeur de Blocs

Pour écrire des lignes de commande à l'application, on accède à l'éditeur de blocs à travers "Blocks". L'éditeur de blocs nous permet de connaître le comportement de l'application. Les boutons "Designer" et "Blocks" peuvent représenter des onglets qui servent à passer d'un espace à l'autre.

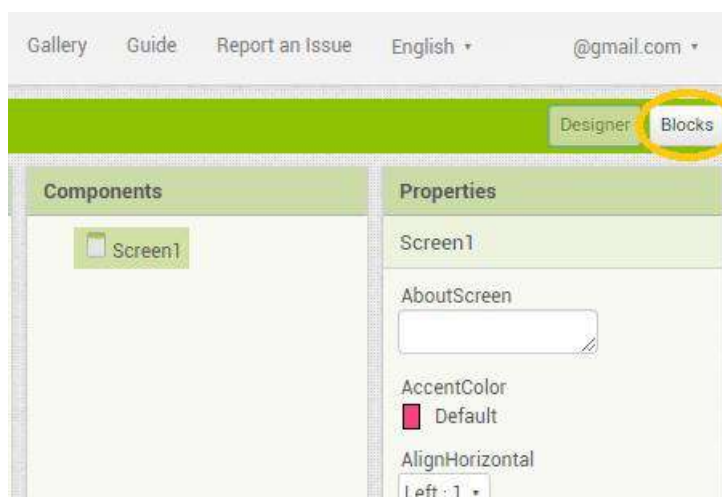


Figure I.7 Bouton d'accès aux fenêtres blocs

I.4.7 Editeur de Blocs

L'éditeur de blocs nous permet de programmer le comportement de notre application d'affichage et mesure. Ces blocs intégrés (briques logicielles) sont capables de gérer des opérations comme (les maths, le contrôle, la logique... etc). Ci-dessous, on trouve les blocs affiliés à chaque composant qui rentre dans la conception de notre application [5].

Pour apparaître en tant que bloc d'un composant spécifique dans l'éditeur de blocs, nous devons d'abord ajouter ce composant à l'application via le Designer.



Figure I.8 Fenêtre de la programmation par bloc

Les éléments utilisés dans notre "Screen1" (en bas à gauche) exposent bien quelques significations pour quelques blocs et éclaircir la facilité de cet environnement.

- Contrôle : Représente les éléments de test de boucles. comme titre d'exemple
- Logique : Représente les éléments de la liaison la logique, "vrai" et "faux", ...
- Math : Regrouper les outils mathématiques (multiplication, soustraction, addition... etc).
- Text : Gestionnaire de texte.

I.5 Programmation par bloc sous App Inventor

D'après tout ce qui a été éclairci auparavant, la réalisation d'une telle application prend la tâche d'afficher les différents grandeurs contournant un panneau photovoltaïque qui doit passé par quelques étapes. Tout d'abord, on commence par un assemblage de différents composants graphiques sous la fenêtre principale pour donner l'apparence à notre application ensuite, on affectera nos commandes par la programmation avec blocs.

La figure ci-dessous montre l'allure de notre application avec un bouton d'accès pour activer le Bluetooth qui porte un icône bleu ainsi que six zones de textes qui prennent la tâche d'afficher les valeurs des grandeurs et l'état de la connexion.

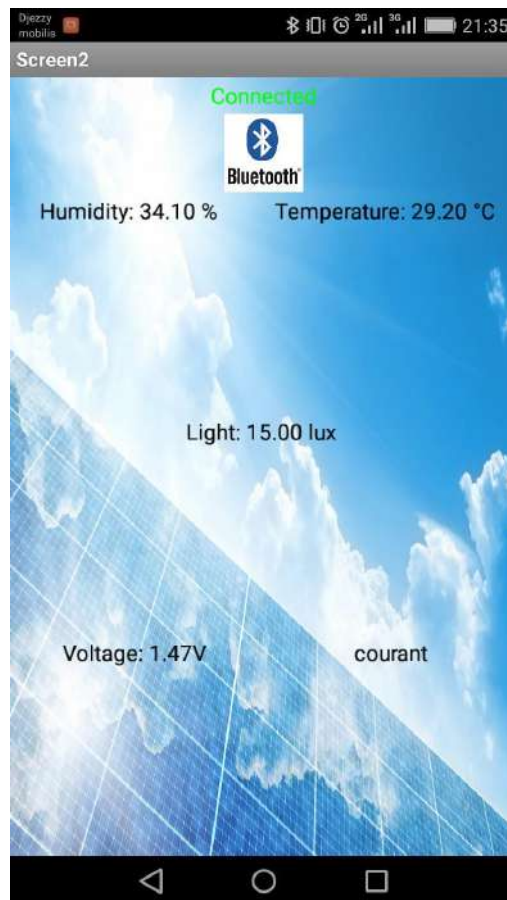

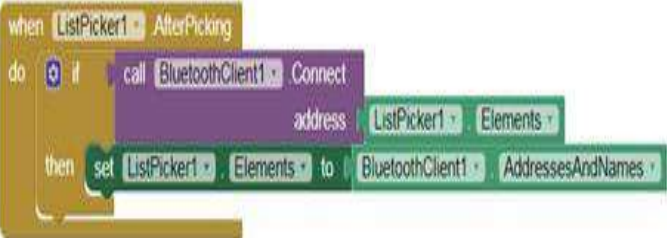





Figure I.9 Allure de l'application sous smart phone

Nous n'écrivons pas de lignes de code mais nous utiliserons un mode de programmation graphique. Le tableau qui suit résume toutes les commandes réalisées par bloc dans notre application.

Dans l'annexe (A) on trouve, les deux interfaces d'affichage de notre application et dans l'annexe (B), on trouve l'allure globale de l'ensemble bloc réalisé sur l'espace de travail de l'interface blocks.

Blocs	Fonction
	<p>Cet événement se produit avant que la liste d'éléments ne soit affichée et peut être utilisée avant qu'elle ne soit affichée. Donc, dans ce block on prépare l'adresse et le nom de l'appareil qui porte le module Bluetooth</p>
	<p>Si le module Bluetooth est disponible il est connecté par son adresse dans le cas contraire on sera obligé de refaire l'opération</p>
	<p>Ce bloc visualise une zone de texte dans lequel le composant "label" affiche l'état du module Bluetooth avec le couleurs vert et rouge à l'aide d'un composant "timer" qui vérifie le changement tout les 1000 ms</p>
	<p>Ce bloc est utilisé pour créer des variables globales. Il prend n'importe quel type de valeur comme argument. En nommant ce bloc selon nos besoins, ces variables seront utilisées dans toutes les procédures et tous les événements afin que ce bloc soit autonome.</p>
	<p>Quelques blocs associés pour recevoir une trame d'informations à partir du module Bluetooth</p>


	<p>Quelques blocs associés pour déchiffrer la trame émise par la carte Arduino en distribuant les valeurs des grandeurs mesurées composant "label" pour afficher les grandeurs en tant que caractères</p>
---	---

Tableau I.1 Signification des blocs de notre application

I.6 Conclusion

Afin de faciliter la tâche de la programmation sans taper la moindre instruction, la programmation a été réalisée à travers l'association de quelques briques logicielles seulement en les paramétrant (langage Scratch). L'utilisation et la manipulation de block de App Inventor de Google rendent la réalisation des applications Android plus faciles que de travailler avec des programmes tel que Android Studio qui nécessite une connaissance approfondie dans la programmation en Java.

Dans le chapitre qui suit, nous allons éclaircir les étapes de la réalisation de notre dispositif qui repose sur quelques capteurs. On les considère comme des entrées du système et des sorties sur le smart phone comme affichage.

Chapitre II

Réalisation d'un dispositif

Expérimental

II.1 Introduction

Dans ce chapitre, nous allons brièvement présenter un aperçu du dispositif expérimental qui est la réalisation d'une carte électronique qui mesure la globalité des grandeurs influant sur le rendement d'un panneau photovoltaïque à l'aide d'une carte Arduino à connectivité Bluetooth.

Après avoir expliqué et réalisé l'application Android exécutable sur un smart phone développé sous l'environnement App Inventor dans le chapitre précédent, nous allons aborder maintenant tous les blocs rentrant dans la conception de l'application expérimentale commençant par la mise en évidence sur le noyau de notre dispositif qui est une carte Arduino Mega suivie par l'explication des différents capteurs utilisés ainsi que le protocole d'échange d'informations.

II.2 Système embarqué

Un système embarqué est déterminé comme un système électronique informatisé autonome élaboré dans une tâche bien précise. Le terme désigne aussi le matériel informatique ainsi que le logiciel utilisés. Ses ressources sont généralement limitées [6].

Dans notre travail, nous sommes exposés à deux types de ce système (une carte Arduino et un smart phone).

II.2.1 La carte Arduino

Dans les années 2000 en Italie, le projet Arduino est apparu comme un mini système embarqué; il représente les efforts d'un groupe d'enseignants et d'étudiants de l'école de Design d'Interaction d'Ivrea. Leur objectif était de démocratiser l'utilisation des microcontrôleurs en simplifiant les manipulations avec une illustration sur une carte électronique plus petite et facile à accéder.

Ce projet a évolué rapidement et a trouvé sa place dans le monde de l'électronique informatisé. Il regroupe désormais une communauté nombreuse et active, ce qui a permis à l'Arduino de continuer de s'améliorer à une vitesse remarquable.

Une des cartes les plus populaires "Arduino Mega" est un modèle évolué de la carte Uno; elle dispose de 54 Entrées/Sorties (comparant avec 14 pour l'Arduino Uno), avec plus de mémoires et un processeur plus puissant sur une surface qui mesure 101,6 mm x 53,3 mm (ce qui est positif pour manipuler l'électronique embarqué). La figure ci dessous présente une description de la carte utilisée.

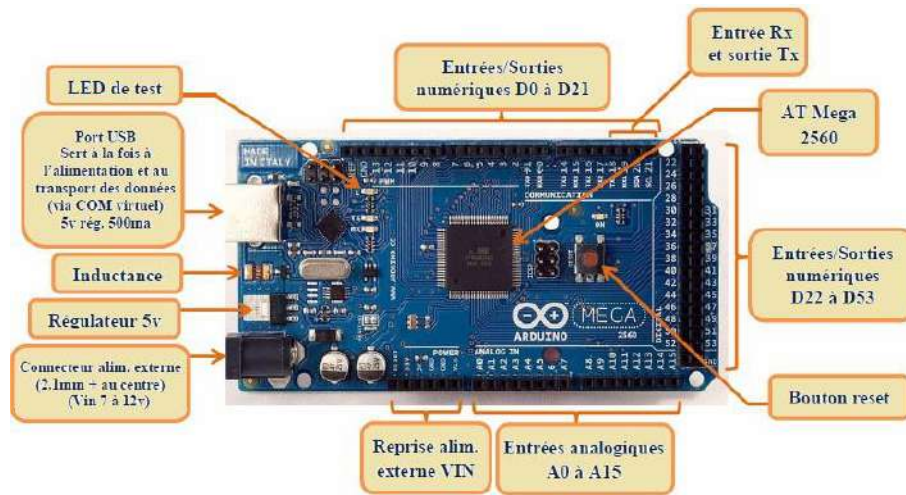


Figure II.1 Description de la carte Arduino Mega 2560

II.2.2 Téléphone intelligent

Le smart phone ou "téléphone intelligent" signifie un téléphone mobile muni de fonctionnalités avancées similaires à celles d'un ordinateur. Doté d'un processeur performant, il embarque une gamme de capteurs (accéléromètre, boussole, gyroscope...etc.) qui lui permettent d'exécuter des applications destinées aux activités physiques. Il est connu par ses multiples liaisons de connectivité (Bluetooth, GPS, WiFi, NFC...etc).

II.3 Définition d'un Capteur

Un capteur est un élément électronique capable de convertir l'état de la quantité réelle observée en une quantité utilisable. Mesurer n'importe quelle grandeur physique c'est donner une valeur quantitative en prenant comme référence une grandeur de même nature appelée unité. Pour éclaircir la définition exacte d'un capteur quelques termes doivent être connus.

- Le mesurande : Représente simplement la grandeur à mesurer.
- Le mesurage : Est l'ensemble de toutes les opérations qui permettent de déterminer la valeur (d'un mesurande).
- La mesure : Représente le résultat du mesurage.

De ce fait, un capteur présente un dispositif dont les caractéristiques et les propriétés physiques sont sensibles à un mesurande. Quand ce dernier est soumis à ce mesurande, il produit une réponse sous la forme d'une quantité électrique utilisable. La figure suivante résume ce qu'est un capteur.

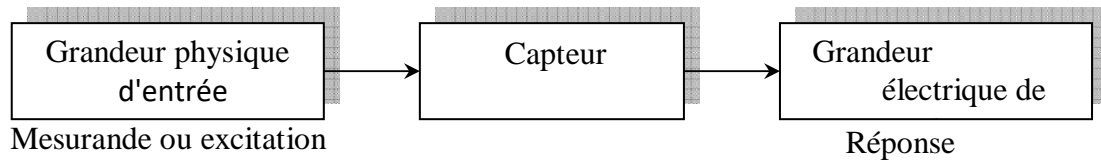


Figure II.2 Schéma synoptique d'un capteur

II.3.1 Classification des Capteurs

Il existe plusieurs façons de classer les capteurs:

- selon le mesurande qu'il traduit (capteur d'humidité, température, position, pression... etc).
- En fonction de son rôle dans une opération industrielle (contrôle d'un produit, surveillant, sécurité... etc).
- La nature du signal fournit à la sortie (soit une nature numérique ou analogique).

II.3.2 Évolution des Capteurs

Comme dans le reste de l'industrie, les capteurs analogiques cèdent le place aux capteurs numériques qui se matérialisent par l'apparition des capteurs intelligents. Leur objectif est le même, simplifier la mise en place des boucles de surveillance et contrôle. Pour cela, ils utilisent une liaison unique entre les différents intervenants de la boucle de régulation (capteurs, régulateurs, étalonnage, actionneurs), liaison qui sert à la fois au dialogue entre ces intervenants et à leur alimentation en énergie. Ainsi, l'ajout d'un intervenant dans une boucle complexe se résume en deux interventions [7]:

- Le montage de l'intervenant sur le but.
- L'adaptation par l'intermédiaire d'un logiciel du fonctionnement de la mesure.

II.4 Grandeurs à Mesure

Tous les panneaux solaires sont exposés à plusieurs grandeurs qui influent directement sur leurs rendements. Ces grandeurs peuvent être internes ou externes.

II.4.1 Grandeur éclairement

L'éclairement lumineux est la grandeur définie par la photométrie correspondant à la sensation humaine de l'éclairement. Plus un objet n'étant totalement pas noir et ne produit pas de lumière par lui-même est éclairé, plus il est visible distinctement. La photométrie définit

cette grandeur uniquement afin de pouvoir la calculer en connaissant l'intensité lumineuse des sources de lumière, leur distance et leur direction [8].

II.4.2 Grandeur Humidité

L'humidité relative de l'air correspond à la quantité de vapeur d'eau dans l'air par rapport à la capacité maximale dont ce dernier peut en contenir. L'humidité relative de l'air s'exprime en pourcentage (%) et dépend de la température et de la pression.

L'humidité mesure la quantité de vapeur d'eau présentée dans l'air sans compter l'eau liquide et la glace. Pour que des nuages se forment et qu'ils y aient des précipitations, l'air doit arriver à une humidité relative légèrement supérieure à 100% dans le voisinage des gouttelettes qui se forment.

II.4.3 Grandeur Température

La température est une propriété physique qui se réfère aux notions communes de chaleur ou pas, mais sa signification formelle en thermodynamique est plus complexe. La température est étroitement liée à l'énergie interne et l'enthalpie d'un système. Elle est généralement mesurée avec un thermomètre [9].

Par définition, une température est une mesure numérique d'une chaleur, sa détermination se fait par détection de rayonnement thermique, la vitesse des particules, l'énergie cinétique ou par le comportement de la masse d'un matériau thermométrique.

II.4.4 Grandeur Tension

une grandeur fréquemment utilisée chez les électriciens "tension électrique", elle donne la notion de la différence de potentiel électrique entre deux points d'un montage électrique. La tension électrique présente la circulation du champ électrique tout le long du circuit électrique mesuré en volts. la signification du terme tension électrique est souvent approchée à celle de la différence de potentiel électrique (DDP) entre deux points d'un circuit électrique.

II.4.5 Grandeur Courant

Le courant électrique coïncide au mouvement interne d'un électron dans un conducteur de charges habituellement négatives. La direction conventionnelle du courant est dirigée du pôle positif vers le pôle négatif. Le sens exact du mouvement des porteurs de charges est différent de cette dernière. Le mouvement des électrons est emmené du pôle négatif au pôle positif.

II.5 Unités de Mesure

Afin de réaliser notre dispositif, quelques grandeurs doivent être récoltées par un ensemble de capteurs dont leurs fonctions varient selon la nature de la grandeur.

II.5.1 Le Capteur DHT 22

Le DHT22 est un capteur à bas coût qui permet d'avoir la valeur de la température et de l'humidité ambiante d'une façon numérique. Il emploie un capteur capacitif désigné à la grandeur "humidité" et une thermistance pour mesurer la température et l'humidité de l'air afin de les transmettre via un bus série numériquement. Sous une boucle de deux secondes, une opération d'actualisation se déclenche pour réinitialiser les données.

La connexion de ce capteur est simple, il suffit d'attacher la première PIN à gauche à l'alimentation (3Volts ou 5Volts), la deuxième PIN nommée DATA sur une PIN de la carte Arduino déclarée comme entrée (INPUT) et quatrième PIN de droite à la masse (GND).

Ce capteur ressemble au DHT11 et il dispose d'une plus grande précision sur une grande zone de mesures.

Ce capteur doit être lié généralement à une résistance de pull up entre (4,7K Ω et 10K Ω) pour attacher la PIN "DATA" au "V_{CC}".

Sur le tableau qui suit quelques propriétés caractérisent notre composant ainsi qu'un schéma explicatif qui montre sa conception interne.

Caractéristiques techniques	
Alimentation	3 to 5V alimentation et DATA
Consommation	2.5mA max pendant la conversion
Etendue de mesure température	de -40°C à 80°C avec une précision $\pm 0.5^\circ\text{C}$
Etendue de mesure humidité	de 0% à 100% avec une précision à 2-5%
Chantournage	à 0.5 Hz (toute les 2 seconds)
Dimension	27mm x 59mm x 13.5mm (1.05" x 2.32" x 0.53")
nombre de broches	4 pins, 0.1" espacement

Tableau II.1 Caractéristiques techniques du DHT11

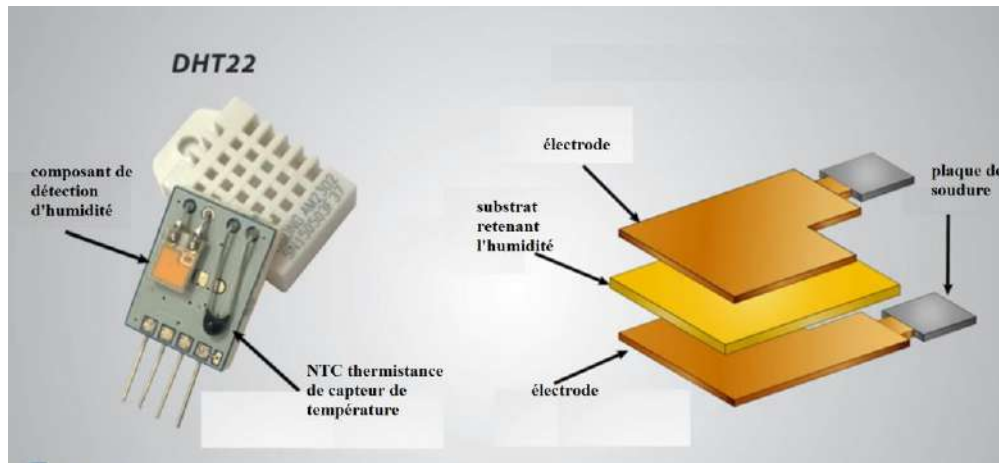


Figure II.3 Structure interne du DHT11

La figure II.4 montre le boîtier du capteur ainsi que son brochage avec l'Arduino.

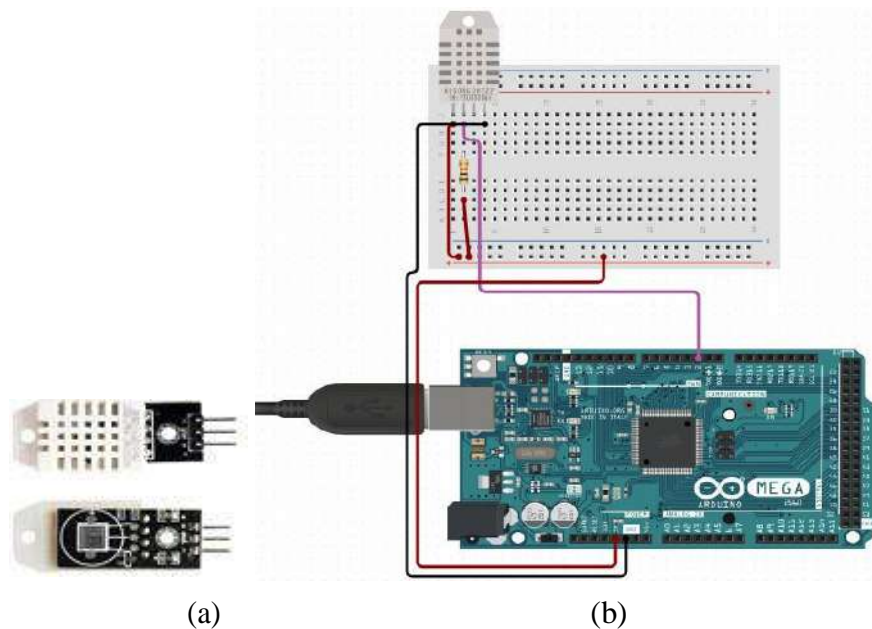


Figure II.4 (a) Le boîtier de DHT 22; (b) Schéma de connexion de DHT 22 avec l'Arduino

II.5.2 Capteur de lumière BH 1750

Une cellule photoélectrique (appelée photo résistance ou cellule photovoltaïque) est un dispositif doté d'un capteur photosensible où ses propriétés électriques (tension, résistance, ...) varient en fonction de l'intensité du rayonnement lumineux capté.

Nous allons utiliser le capteur BH1750, qui est un capteur numérique de lumière ambiante sur un bus de transfert sous le protocole I²C. Il peut détecter une large gamme à haute résolution (1-65535 lux). Le tableau qui suit donne quelques caractéristiques de ce composant suivi par son schéma descriptif.

Caractéristiques techniques	
Tension d'alimentation	$V = 4.5 \text{ V}$
Température de fonctionnement	$T = -40 - +85 \text{ }^{\circ}\text{C}$
Température de stockage	$T_s = -40 - +100 \text{ }^{\circ}\text{C}$
Courant	$I = 7 \text{ mA}$
Puissance	$P = 260 \text{ mW}$

Tableau II.2 Caractéristiques techniques du capteur BH 1750

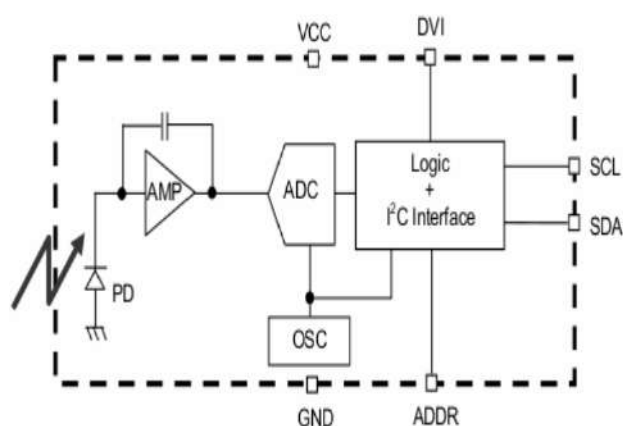


Figure II.5 Conception interne du capteur BH1750

La figure II.6 montre le boîtier du capteur ainsi que son brochage avec l'Arduino.

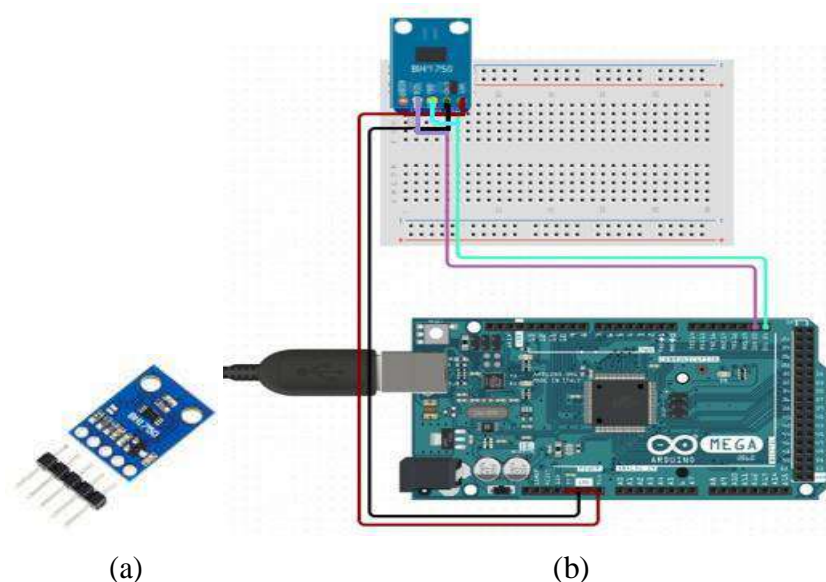


Figure II.6 (a) Le boîtier de capteur BH 1750; (b) Schéma de connexion de BH 1750 avec l'Arduino

II.5.3 Le capteur ACS 712

L'ACS712 est le senseur qui se relie en série avec des charges sur un circuit alternatif ou continu et qui permet de mesurer le courant qui le traverse. Ce capteur utilise le champ magnétique produit par le courant (l'effet hall) afin de mesurer le courant. Le ACS712 génère à sa sortie une tension continue proportionnelle à la valeur du courant à raison de 0.100V/A (100 mV par ampère). Il est donc possible de lire cette tension sur une entrée analogique de l'Arduino. Ce senseur est sensible au champ magnétique ce qui pourrait perturber la lectures. Rapprocher un aimant de l'ACS 712 modifie directement la tension de sortie du capteur pour cela, il est essentiel de faire une lecture à vide (sans courant) afin atteindre la valeur de calibrage du senseur. Quelques caractéristiques de ce composant sont données sur tableau II.3.

Caractéristiques techniques	
Sensibilité	100 mV par ampère
Tension logique	4.5V - 5.5V
Consommation	10mA
Délai de répercussion sur la sortie	5 μ S
Erreur	1.5% à 25°C
Résistance interne de conduction	1.2 mOhms
Poids	~2gr
Bruit	130mA

Tableau II.3 Caractéristiques techniques du ACS712

Afin de filtrer le bruit une capacité de 10nF est associée. Le bruit peut être significativement réduit en utilisant une capacité de 470nF.

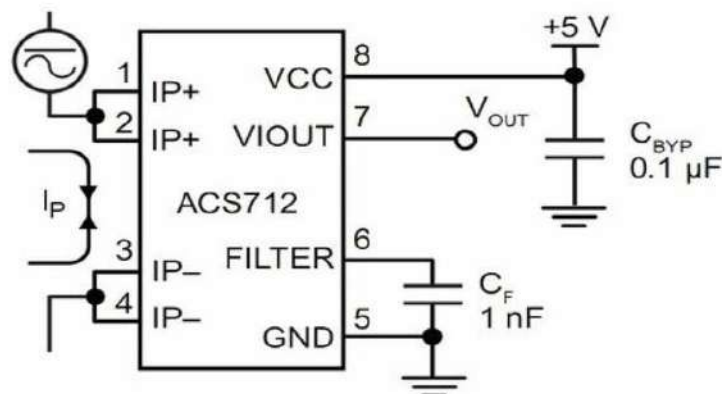


Figure II.7 Schéma explicatif du capteur

Le choix de capteur est ACS 712 - 05B-T sa PIN de référence est reliée à la PIN analogique A10 de la carte Arduino comme montré sur la figure.

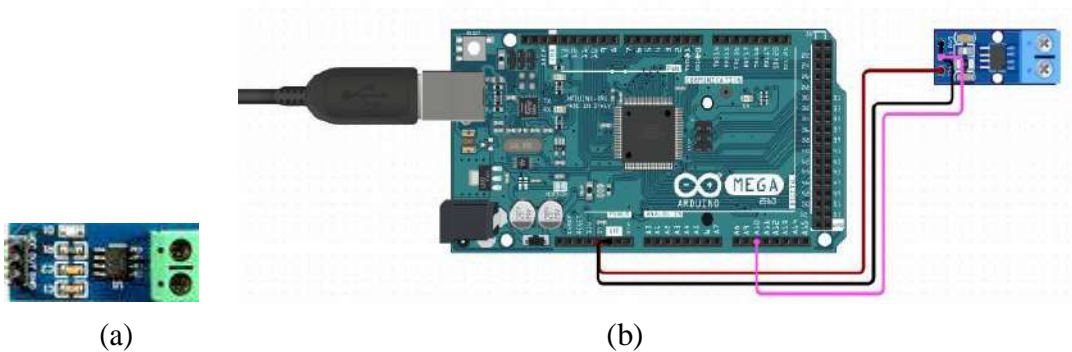


Figure II.8 (a) Le boîtier Capteur ACS 712; (b) Schéma de connexion de ACS 712 avec l'Arduino

II.5.4 Le capteur B25

Le Capteur de tension B25 est un détecteur de tension de 0 à 25V basé sur le principe de diviseur par résistances. La valeur de la tension d'entrée est réduite de 5 fois, la valeur maximale de l'entrée analogique de l'Arduino est de 5 Volts, donc la tension d'entrée de la détection de tension ne peut pas être supérieure à 5 Volts multiplié par 5 = 25 Volts (si un système 3.3 Volts est utilisé, la tension d'entrée ne peut pas être plus grande que 3.3Volts multiplié par 5 = 16.5 Volts).

- Tension d'entrée : DC 0 ~ 25V.
 - Plage de mesure : DC 0.02445V ~ 25V.
- Résolution de mesure : 0.00489V.

Sur la PIN 9 de la carte Arduino Mega en chante la PIN de référence de ce capteur comme le montre la figure ci dessous.

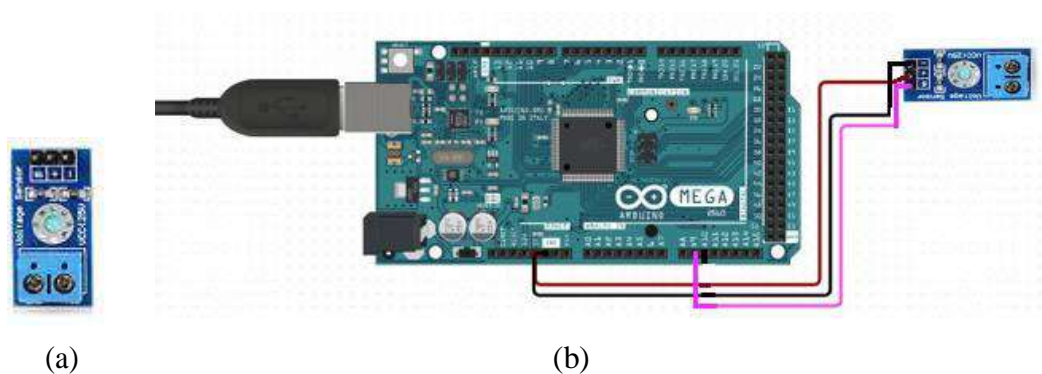


Figure II.9 (a) Le boîtier capteur B25; (b) Schéma de connexion de B25 avec l'Arduino

II.6 Protocole de communication

le protocole de communication est un terme utilisable dans l'informatique et la télécommunication. Il définit certaines règles pour un type de communication particulier. Tout ce qui est appelé protocole est employé pour communiquer entre deux systèmes différents sur la même couche d'abstraction.

Le Bluetooth est un des protocoles le plus connu dans le domaine de communication sans fil de type série conçu pour connecter des appareils mobiles entre eux ou autres appareils. L'utilisation la plus courante consiste à connecter le téléphone à un haut-parleur sans fil ou à des écouteurs. Avec l'aide du module HC-06, nous pouvons établir un contact direct entre la carte Arduino et le smart phone.

La tension d'allumage pour ce module doit être prise entre (3,3 et 5) Volts, mais la broche RX ne peut recevoir qu'une tension maximale de 3,3 V. Il serait nécessaire de prévoir un pont diviseur de tension pour réduire la tension de 5 Volts fournie par la carte Arduino afin de ne pas endommager la broche RX du module Bluetooth.

Dans le tableau suivant, nous éclaircirons les commandes les plus importantes que nous avons utilisées pour configurer le module Bluetooth HC-06.

Commande	Fonction
AT+VERSION?	Donner la version Bluetooth HC-05.
AT+ORGL	Restaurer les paramètres par défaut
AT+NAME?	Donner le nom du dispositif du Bluetooth.
AT+NAME=MyName	Placer le nom avec "le nom"
AT+ROLE?	Donner le rôle.
AT+ROLE=role-number	Placer le nombre de rôle. 0 par défaut.
AT+PSWD?	Le mode passe du Bluetooth à 4 chiffres
AT+PSWD= passcode	Placer le mode passe du Bluetooth de 4 chiffres
AT+UART?	Pour donner la vitesse baud, le bit de parité et le bit d'arrêt.
AT+UART	Placer la vitesse baud, avec parité bit et stop bit. 9600 par défaut.

Tableau II.4 Tableau de commande du Bluetooth

La liaison du module Bluetooth avec la carte Arduino se fait à travers les deux PINs TX0 et RX0 comme présentée sur la figure qui suit.

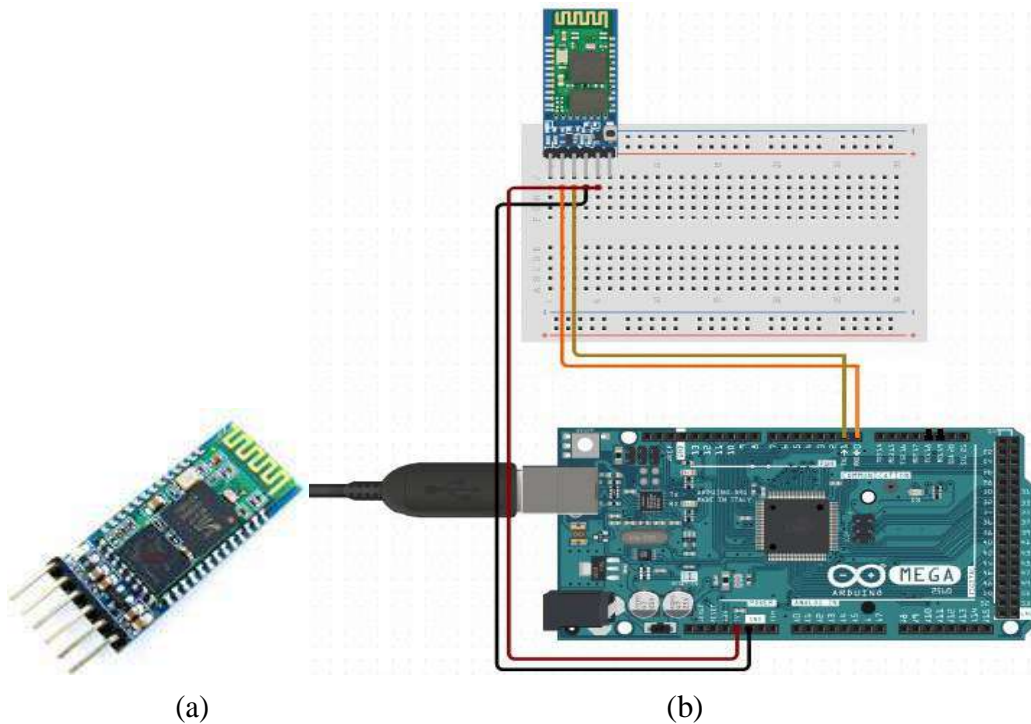


Figure II.10 (a) HC-06 Bluetooth; (b) Schéma de connexion de HC-06 avec l'Arduino

Afin de confirmer la communication, le terminal série du PC nous montre toutes les valeurs émises par les capteurs comme illustré sur la figure ci dessous.

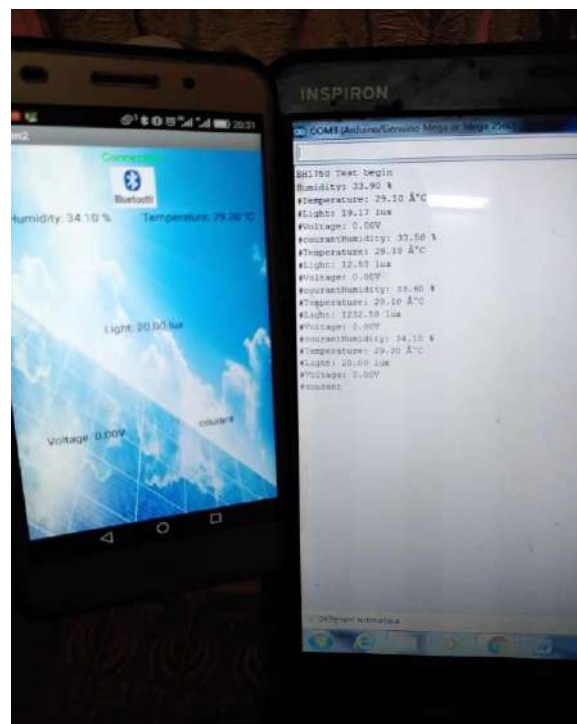


Figure II.11 L'affichage des valeurs récoltées sur le terminal du PC

II.7 Assemblage du dispositif

Pour passer à notre réalisation pratique, une étape doit être effectuée qui est la réalisation virtuelle sous un environnement CAO avec l'utilisation d'un site dynamique [10] nous permettant de dessiner des schémas électroniques et programmer le microcontrôleur de notre carte Arduino. Les deux figures ci dessous présentent respectivement le dispositif globale réalisé virtuellement et réellement.

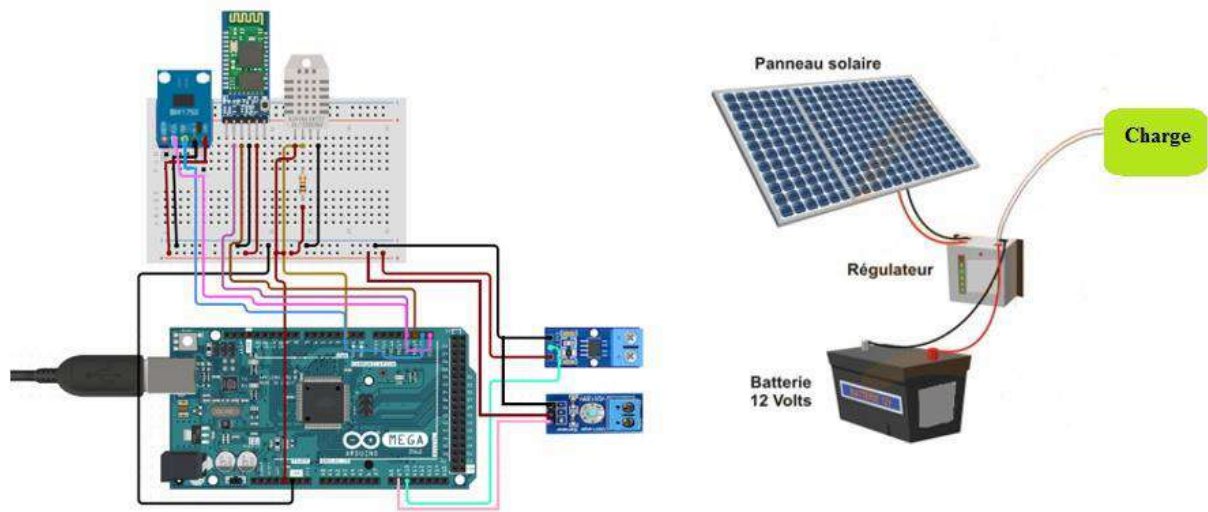


Figure II.12 Dispositif général réalisé sous CAO

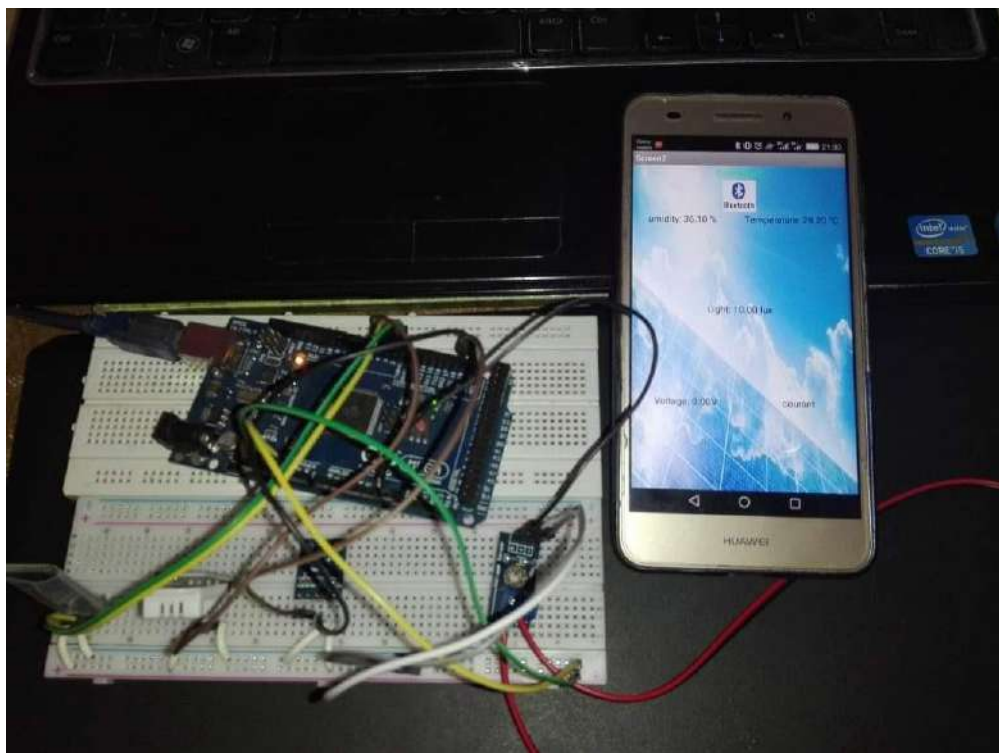


Figure II.13 Dispositif général

II.8 Conclusion

Cette partie a donné lieu à la réalisation pratique d'un dispositif électronique pour mesurer les différentes valeurs des grandeurs récoltées par les capteurs. L'ensemble des travaux rapporte dans ce chapitre :

- La conception des éléments électroniques qui composent la carte réalisée à l'aide du site dynamique sur web [10]
- Le choix des composants adéquats pour chaque grandeur.
- Le choix convenable du système embarqué tel que l'Arduino Mega.

La démarche des différentes étapes de la programmation sera présentée dans le chapitre suivant.

Chapitre III

L'environnement

de développement

IDE de l'Arduino

III.1 Introduction

Dans les chapitres précédents, nous avons réalisé une application sous Android qui communique via un module Bluetooth avec notre dispositif électronique. Ce dernier se base essentiellement sur la carte Arduino Mega pour mesurer et afficher les grandeurs entourant le panneau photovoltaïque. La programmation de ce système embarqué se fait avec plusieurs logiciels. Pour notre carte Arduino nous avons effectué cette programmation seulement en utilisant son environnement de développement qui est IDE de l'Arduino étant un programme open source installée sur divers systèmes d'exploitation.

Sachant que le rendement d'un panneau photovoltaïque est affecté par quelques paramètres climatiques tel que la température et l'éclairement; de cela, nous avons choisi les capteurs adéquat afin de mesurer ces grandeurs et les afficher sur un smart phone.

Dans ce qui suit, nous allons essayé d'expliquer la majorité des commandes exécutées sous l'environnement IDE.

III.2 L'environnement IDE de l'Arduino

Le choix du logiciel de la programmation de notre module Arduino Mega est une application libre à multi plates formes, agissant en tant qu'éditeur de code et compilateur. Il peut transférer le fichier hexadécimal "firmware" et le programme via des multiples liaisons séries selon les modules (USB, RS-232, Bluetooth, Wifi...etc.). Il est possible de se passer de l'interface Arduino, de compiler et de transférer le code à travers l'interface des instructions de commande.

Les instructions utilisées dans ce langage de programmation sont celles du langage C++ mais elles sont compilées avec le AVR-G++. Ce logiciel est lié à une bibliothèque de développement qui permet d'utiliser la carte Arduino avec ses entrées / sorties. Tout utilisateur maîtrisant les langages C ou le C++ peut participer dans le développement de ce programme sur les plates-formes de l'Arduino.

III.3 Structure du programme

Le code est structuré en sketches. Tout sketch est dessiné sur un répertoire et au moins un fichier d'une extension ".ino". Sur l'éditeur de cet environnement, la fonction `main()` est incluse mais elle est dissimulée. Une obligation doit être écrite qui est la déclaration des deux fonctions "`setup()` et `loop()`" semblables à celles de C et C++ "héritage et classes".

III.3.1 Partie déclaration

Dans cette partie, nos cinq variables comme il est mentionné sur la figure ci-dessous sont des espaces réservés dans la mémoire du microcontrôleur de notre carte Arduino comme un case dont la taille est juste assez pour un unique type d'information. Elles comportent un nom qui permet un accès facile.

Pour ne pas avoir écrit des programmes qui présentent l'interaction de nos composants utilisés, des librairies sont disponibles sur Net et ont le rôle de rassembler toutes les fonctions utiles pour que nous pouvons traiter et gérer nos instructions. L'instruction `#include <la librairie.h>` sera alors ajoutée au début du programme.

- `#include <DHT.h>`. Afin d'intégrer toutes les commandes du capteur de la température humidité
- `#include <Wire.h>`. Cette librairie nous permet de communiquer avec le capteur BH1750 qui utilise le protocole I2C.
- `#include <BH1750.h>`. Afin de distinguer toutes les commandes du capteur de l'intensité lumière sur la ligne SDA des données "la broche numérique 20" et la ligne SCL d'horloge "la broche numérique 21".

Une autre déclaration réside dans le mot clé `"#define"` où on a défini quelques adressages "entrées, sorties".



```
m
#include "DHT.h"

#define DHTPIN 2    //
#define DHTTYPE DHT22 //
#define TH 4
#include <Wire.h>
#include <BH1750.h>

BH1750 lightMeter;

int maxHum = 60;
int maxTemp = 40;
int offset =20;//
DHT dht(DHTPIN, DHTTYPE);
int val11;
float val2;
```

Figure III.1 Zone de déclaration

III.3.2 Partie fonctions

D'après la syntaxe du langage IDE de l'Arduino, la fonction `setup()` présente la configuration initiale des différentes entrées-sorties ainsi que les librairies.

```
void setup() {  
  pinMode(TH, OUTPUT);  
  Serial.begin(9600);  
  // initialisation I2C bus  
  Wire.begin();  
  lightMeter.begin();  
  Serial.println(F("BH1750 Test begin"));  
  dht.begin();  
}
```

Figure III.2 Configuration initiale

Explication de quelques instructions rentrant dans notre programme.

- `pinMode` : Définir la broche sélectionnée pour qu'elle soit en entrée ou en sortie.
- `Serial.begin()` : Initialiser le port de communication.
- `Wire.begin()` : Initialiser la librairie du bus I²C.
- `Serial.println(" ")` : Instruction d'affichage sur un terminal.

Quant à la fonction `loop()` comme montrée sur la figure ci dessous, elle présente la boucle principale qui s'exécute répétitivement.

Les structures de contrôle présentent des blocs d'instructions qui s'exécutent en fonction d'un certain nombre de conditions "fonctions désignées sous le nom de procédure ou sous une déclaration prise d'une librairie".

Les déclarations suivantes ; `dht.readHumidity()`, `dht.readTemperature()`, `lightMeter.readLightLevel()` présentent des fonctions données par nos librairies incluses.

```
void loop() {
  // Attend quelques secondes entre les mesures.
  delay(2000);
  // La lecture de la température ou de l'humidité prend environ 250 millisecondes!
  float h = dht.readHumidity();
  // lucture de temperature en Celsius
  float t = dht.readTemperature();
  // mesure de l'intencité de Lumière
  float lux = lightMeter.readLightLevel();
  // Vérifiez si les lectures ont échoué et quittez tôt (pour réessayer).
  if (isnan(h) || isnan(t)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }

  if(h > maxHum || t > maxTemp) {
    digitalWrite(TH, HIGH);
  } else {
    digitalWrite(TH, LOW);
  }
  {
    float temp;
    val11=analogRead(A9);
    temp=val11/4.092;
    val2=(temp/10);
  }
}
```

Figure III.3 Fonction principale loop

Sur la figure qui suit, on réalise un affichage qui porte toutes les variables et toutes les grandeurs mesurées sur une chaîne de caractère avec une boucle répétitive d'une minute.

```
Serial.print("Humidity: ");
Serial.print(h);
Serial.println(" %\t");
Serial.print("#");
Serial.print("Temperature: ");
Serial.print(t);
Serial.println(" °C ");
Serial.print("#");
Serial.print("Light: ");
Serial.print(lux);
Serial.println(" lux");
Serial.print("#");
Serial.print("Voltage: ");
Serial.print(val2);
Serial.println("V");
delay(60000);
```

Figure III.4 Boucle d'affichage

- La structure de contrôle suivante présente le code de vérification de la validité du capteur DHT22.

```
if (isnan(h) || isnan(t)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
}
```

Figure III.5 Fonction de vérification de DHT 22

- La structure de contrôle suivante est pour surveiller le seuil des valeurs "température et humidité".

```
if(h > maxHum || t > maxTemp) {
    digitalWrite(TH, HIGH);
} else {
    digitalWrite(TH, LOW);
}
```

Figure III.6 Fonction du seuil

- cette boucle présente les instructions du capteur de tension B25.

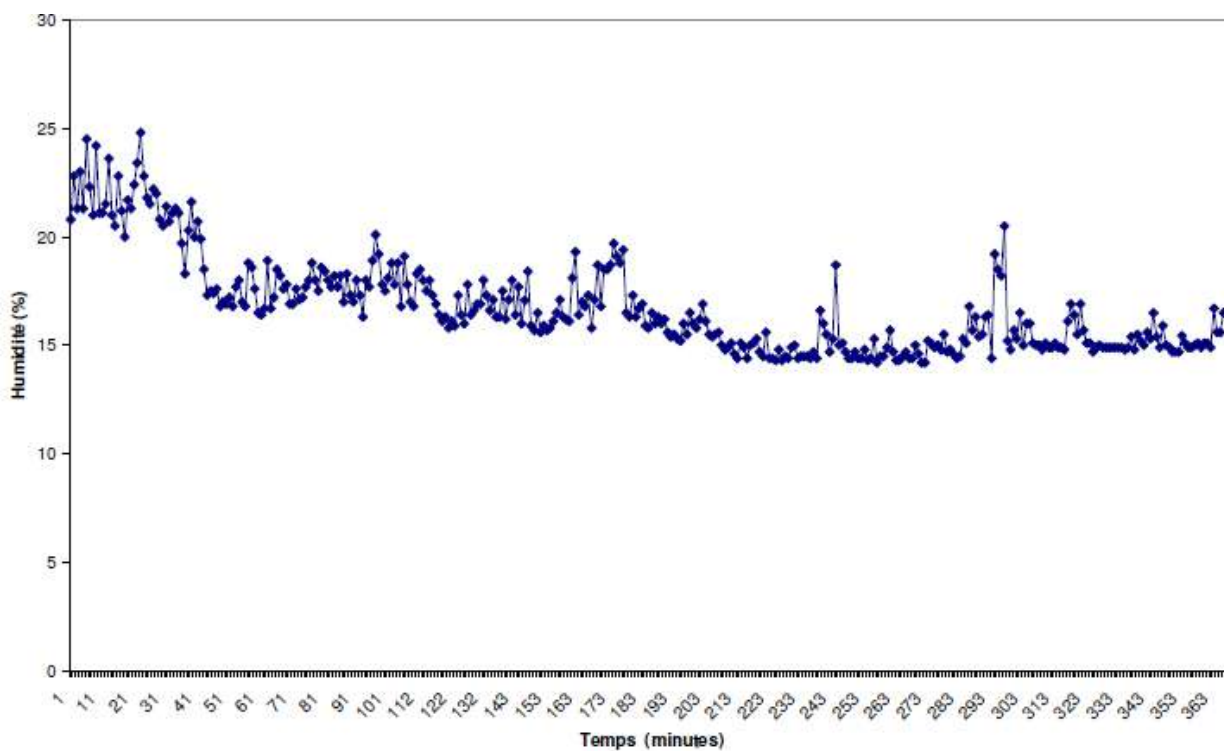
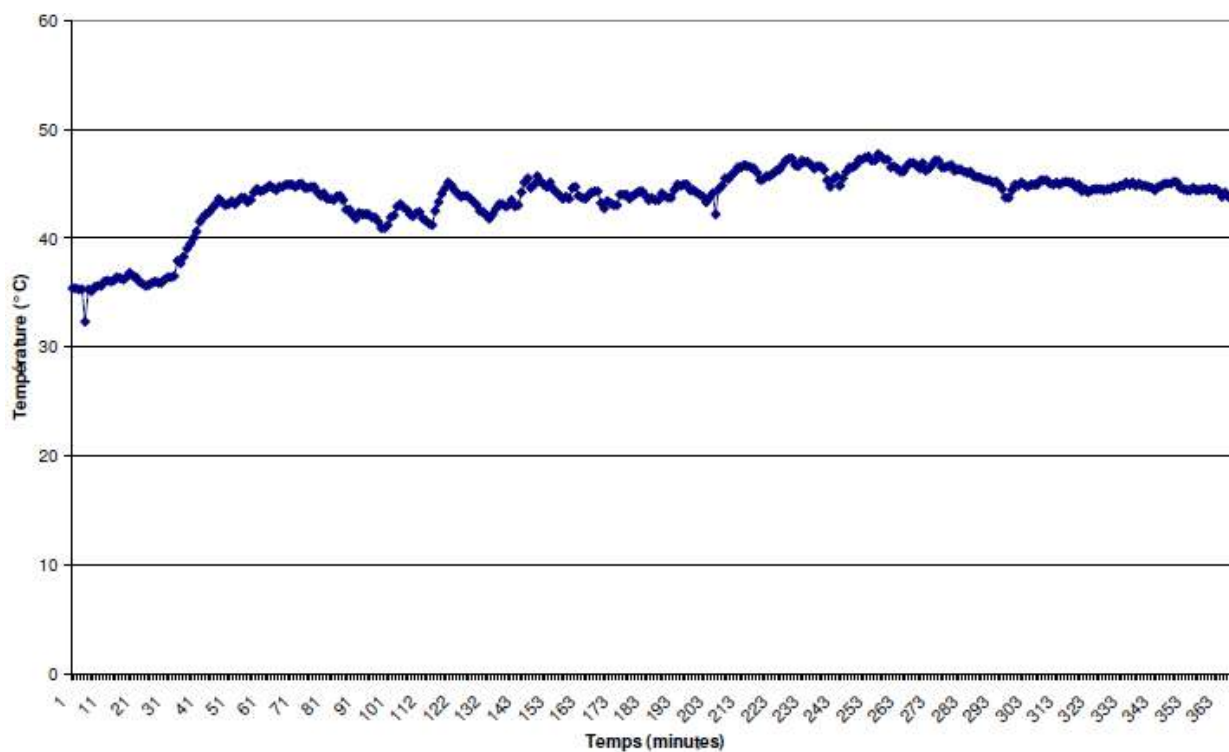
```
{
    float temp;
    val11=analogRead(A9);
    temp=val11/4.092;
    val2=(temp/10);
}
```

Figure III.7 boucle de mesure de tension

III.4 Mesures et tests expérimentaux

Afin de valider les résultats du dispositif et montrer l'efficacité de la carte réalisée, nous avons cédé quelques mesures durant une journée le 19/06/2019 avec un pas de mesure d'une minute. Comme dit précédemment, les grandeurs sont mesurées à l'aide de quelques capteurs et les résultats sont présentés graphiquement sur quelques figures comme montrées ci dessous.

Une ressemblance totale entre les résultats obtenus sur l'affichage du smart phone et l'affichage le port série du PC a été confirmée. Le test a été fait dans la région qui porte les coordonnées polaires latitude 31.9527° N, longitude 5.3335° E correspondant à l'endroit "ville de Ouargla".

*Figure III.8 Tracée "Humidité"**Figure III.9 Tracée "Température"*

Quand la température augmente → l'humidité diminue

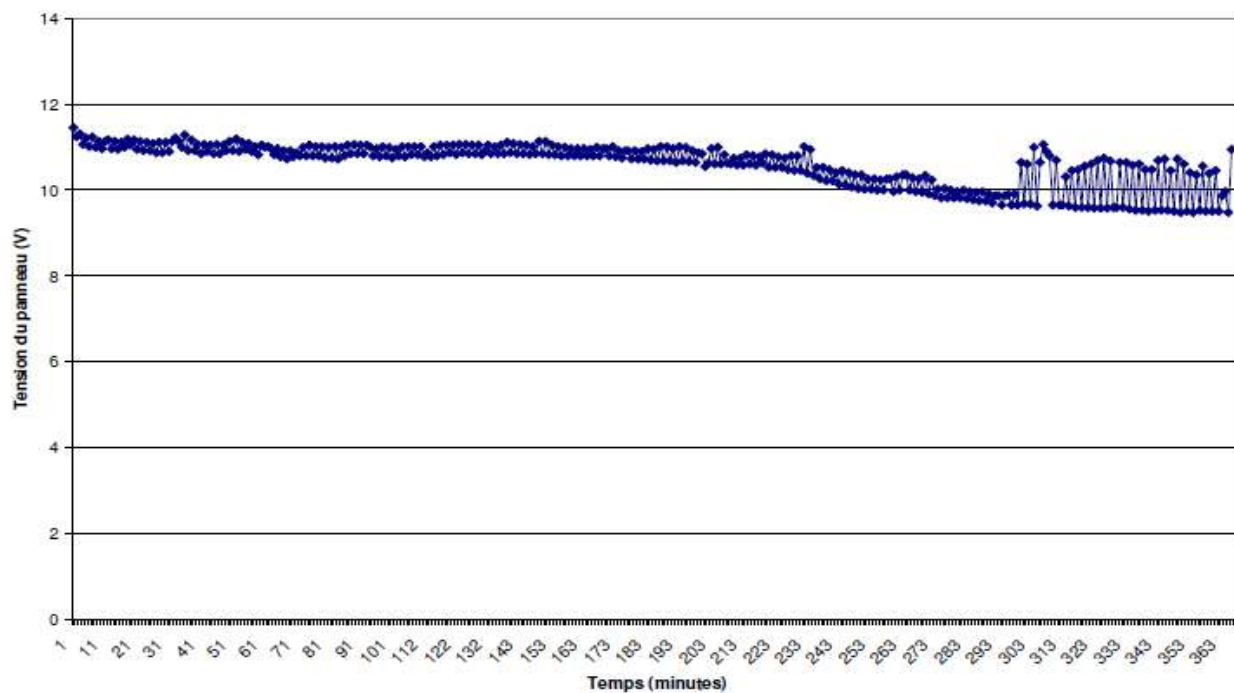


Figure III.10 Tracée "Tension de panneau"

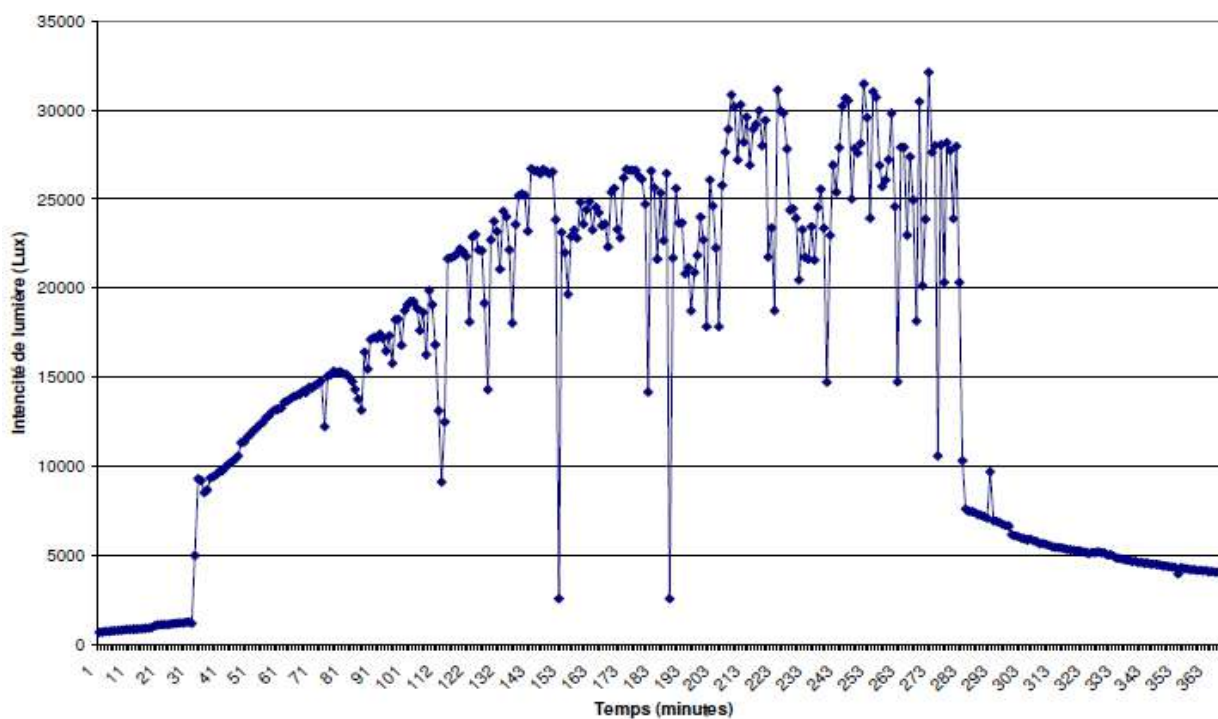


Figure III.11 Tracée "Intensité de lumière"

Afin de confirmer les résultats obtenus durant cette journée, nous avons calculé les moyennes de quelques grandeurs et les comparer avec celles données par une station de

surveillance météorologique déclarées sur un site [11] afin de voir les pourcentages de précision de notre dispositif.

Le tableau qui suit présente la comparaison des valeurs moyennes de nos grandeurs et celles d'une station.

	Notre valeur moyennes	Valeur moyennes station
Température	43.33°C	44°C
Humidité	16.69%	17%
l'éclairement	43294.91	/

Tableau III.1 Comparaison des moyennes

L'augmentation de température influe directement sur la tension fournit par le panneau

III.5 Conclusion

Utiliser un langage de programmation est d'apprendre son vocabulaire et une syntaxe bien définis. Dans ce chapitre, nous avons abordé certaines de ces règles et d'autres considérations relatives à l'écriture d'un programme Arduino. Afin de faciliter la programmation, nous avons utilisé quelques librairies correspond aux quelques capteurs.

Sur le terminal de notre PC à travers l'environnement IDE de l'Arduino, nous avons récolté les valeurs de nos grandeurs et les sauvegarder sur un fichier Excel pour tracer les différentes valeurs émises sur le port série. Ces valeurs peuvent être comme base de données pour autres utilisateurs cherchant à récolter les grandeurs (température, l'humidité, l'éclairement).

Conclusion Générale

1. Généralités

La disponibilité des smart phones comme le plus proche outil chez les gens nous a poussé à le sélectionner comme support pour une application qui occupe un vaste domaine de recherche de la photovoltaïque. Cette dernière est liée instantanément à quelques grandeurs influant sur son rendement de cela, une surveillance continuellement doit être réalisée en associant au smart phone un autre système embarqué qui intègre à son rôle quelques capteurs pour récolter les perturbations météo logiques.

Afin de développer une telle application Android, nous avons utilisé un environnement de développement App Inventor de Google [12]

L'assistance d'une carte Arduino Mega comme noyau d'un programme compilé et développé sous l'IDE de l'Arduino nous facilite à communiquer avec le smart phone via la connectivité Bluetooth.

2. Problèmes rencontrés

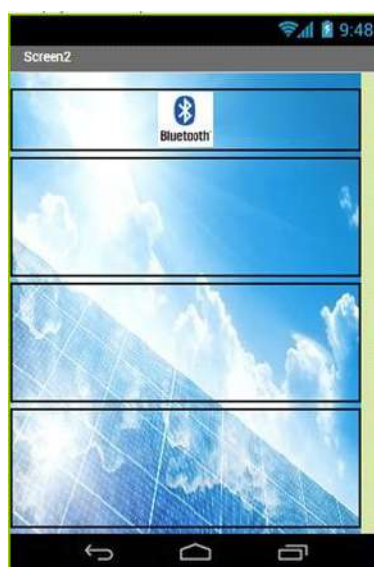
La réalisation n'est pas sans difficultés. Il convient de noter que nous avons rencontré plusieurs problèmes, en particulier lors de la réalisation de la carte.

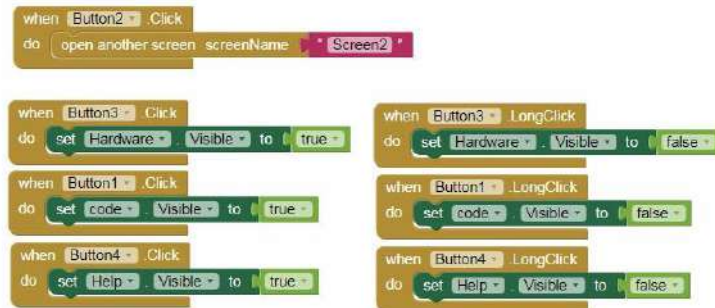
L'insuffisance du temps pour terminer un projet contenant une réalisation pratique.

Manque d'expérience dans le domaine de la programmation et la façon de gérer les problématiques.

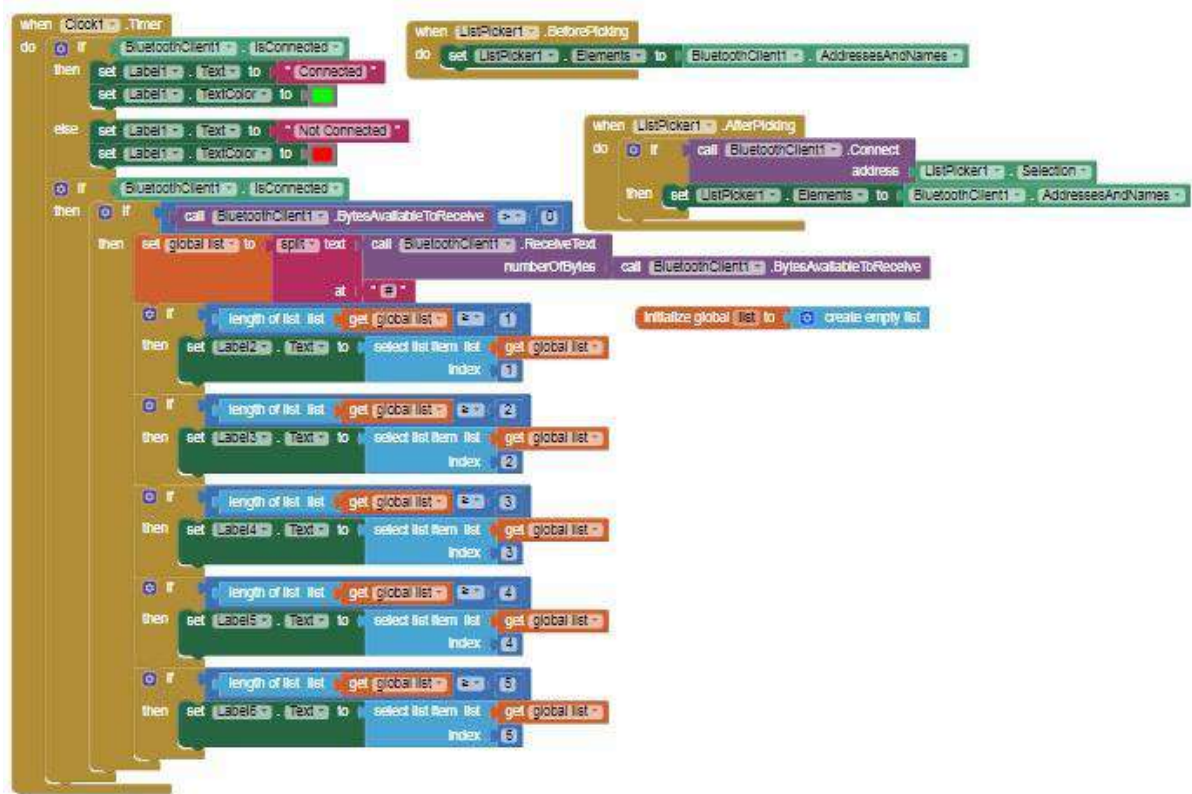
3. Perspectives du projet

Nous espérons sincèrement que ce projet pourra servir de base à de nouvelles études plus approfondies pour l'intégrer à des systèmes plus complexes comme par exemple surveillance à distance par une carte Wifi en utilisant le Net comme support d'affichage.

*Screen 1**Screen 2*



Bloc du Screen 1



Bloc globale du Screen 2

Références

- [1] Hugues DELALIN "Cours de Systèmes d'Exploitation" Departement Service et Réseaux de Communication IUT de Lens - Université d'Artois Année 2005/2006 - SRC 1ère année.
- [2] Gauthier Picard "Initiation à la programmation orientée objet avec le langage Java" Pôle Informatique 2013-2014. école nationale supérieure des mines SAINT - ETIENNE.
- [3] Mit app inventor — explore mit app inventor, 2015, online; last accessed 14-May-2015. [Online]. Available: <http://appinventor.mit.edu/explore>.
- [4] Blockly, 2015, online; last accessed 10-April 2015. [Online]. Available: <https://developers.google.com/blockly/>.
- [5] Tutorials for app inventor 2, 2015, online; last accessed 14-May-2015. [Online]. Available: <http://appinventor.mit.edu/explore/ai2/tutorialsd41d.html>.
- [6] Corinne Alonso "Conception et Commande de Systèmes Electriques Embarqués" Licence Professionnelle, université paul sabatier UPS -Dpt EEA Toulouse.
- [7] www.acim-jouanin.fr.
- [8] <http://www.utc.fr/~tthomass/Themes/Unitesr>.
- [9] I. Sermet-Gaudelus *, I. Chadelat, G. Lenoir "La mesure de la température en pratique pédiatrique quotidienne" Archives de pédiatrie 12 (2005) 1292–1300.
- [10] <https://www.tinkercad.com/>.
- [11] <https://www.accuweather.com/fr/dz/ouargla/5687/june-weather/5687>
- [12] App Inventor 2 « Concevez des applications Android pour mobile »

Résumé

Ce projet nous permet de développer nos connaissances dans le monde de l'électronique informatisé et de réaliser une association entre deux systèmes embarqués qui sont respectivement une carte Arduino Mega et un smart phone par une collectivité Bluetooth.

Le dispositif réalisé sert à mesurer et afficher toutes les grandeurs qui rentrent comme paramètres influant sur le rendement d'un panneau photovoltaïque à l'aide d'une gamme de capteurs. Les deux environnements de développement App Inventor et IDE de l'Arduino nous aident à simplifier la programmation afin d'arriver à résoudre notre problématique.

Mots clé : Système embarqués, Arduino Mega, Smart phone, Bluetooth, App Inventor, IDE

Abstract :

This project allows us to develop our knowledge in the world of computerized electronics and to create an association between two embedded systems that are respectively an Arduino Mega board and a smart phone by a Bluetooth community.

The realized device is used to measure and display all the quantities that enter as parameters influencing the performance of a photovoltaic panel using a range of sensors. Arduino's two App Inventor and IDE development environments help us simplify programming to solve our problem.

Key words: Embedded system, Arduino Mega, Smart phone, Bluetooth, App Inventor, IDE

الملخص :

يسمح لنا هذا المشروع بتطوير معرفتنا في عالم الإلكترونيات المحوسبة وإنشاء ارتباط بين نظامين مضمنين على التوالي هما لوحة أردوينو ميغا وهاتف ذكي من قبل مجتمع البلوتوث

يستخدم الجهاز المحقق لقياس وعرض جميع القيم التي تدخل كمعلومات تؤثر على أداء لوحة الطاقة الضوئية باستخدام مجموعة من المستشعرات. تساعد بيئتي تطوير تطبيقات أب أنفوننتور و نظام البرمجة بطاقة الأردوينو على تبسيط البرمجة لحل مشكلتنا.

الكلمات المفتاحية : النظام المضمن ، أردوينو ميغا ، الهواتف الذكية ، البلوتوث، أب أفننتور ،