

TRACKER SOLAIRE PHOTOVOLTAIQUE



Tuteur du projet : Philippe
LARRAMENDY

Jean-Laurent CABILLON
Jaqueline ORDUNA
Diana Carolina GARCIA
Clément CLAVE

Remerciements

Nous tenons à remercier Monsieur Larramendy, tuteur de notre projet, pour avoir rendu possible la réalisation de ce projet, pour son encadrement et son aide.

Nous souhaitons également remercier l'ensemble des professeurs pour leur aide tout au long du projet et Monsieur LUTHON, directeur de la licence Ecologie Industrielle.

Enfin, que tous ceux ou celles qui ont contribué de près ou de loin à l'accomplissement de ce travail trouvent l'expression de nos remerciements les plus chaleureux.

Résumé

Notre projet implique l'installation d'un « tracker » solaire photovoltaïque à deux axes sur le toit de l'IUT. Celui-ci permettra de charger une batterie de 12V à partir de l'énergie récupérée par un panneau photovoltaïque de 150W de puissance crête. Le système comprend également deux charges, la première à tension continue (12V) et la seconde à tension alternative (230V/50Hz). Ces deux charges pilotables à distance, vont permettre d'étudier le comportement du système lors d'une consommation d'énergie.

Le tracker photovoltaïque est prévu également pour fonctionner suivant deux modes distincts :

- le mode « TP à distance » ; dans ce mode, des étudiants en TP pourront superviser le système à partir d'une interface web,
- le mode « automatique » ; dans ce mode, le tracker solaire photovoltaïque se positionnera automatiquement face au soleil et effectuera des relevés périodiques des différentes grandeurs électriques afin d'alimenter une base de données.

Le projet ayant été entamé il y a deux années, il sera question pour notre groupe de prendre connaissance du travail déjà effectué par les groupes de projet des années précédentes et par la suite :

- finaliser la partie instrumentation déjà en place,
- ajouter un dispositif de mesure de la vitesse du vent,
- finaliser l'interface web de supervision du tracker PV,
- concevoir une interface web pour la visualisation des données enregistrées sur la base de données.

Abstract

This project involve to set up a solar tracker of 150 Watts on the roof of the Montaury scientific pole of the Bayonne IUT. The goal of this project is to recover the electric energy produced by the solar panel in order to stock it in a battery, connected to an electric box, to supply alternative electrical load in 230V / 50Hz.

The aim in the medium term is to set up a practical activity to the DUT GIM students with a web interface to recover the voltage and current density on different points of the electric circuit. We could equally take control of the servomotor to pilot the solar panel and watch the difference of power depending on the solar panel position compared to the position of the sun.

The project was started by last two years students and after work progress our objectives are summarized in:

- Finalize the instrumentation party already in place,
- Add a device for measuring wind speed,
- Finalize the monitoring web interface of tracker PV,
- Design a web interface for viewing data recorded on the database.

Sommaire

Remerciements	1
Résumé.....	2
Abstract.....	3
Introduction.....	6
1. Présentation générale.....	8
1.1 Evolution du projet.....	8
1.2 Objectifs/ Travail à réaliser	10
2. Première partie : Présentation des composants du projet.....	11
2.1 Panneau et armoire :	11
Panneau à 2 axes :	11
Batterie au Gel Victron 110Ah 12V :	12
Régulateur électronique de charge solaire MPPT 15A:	12
Convertisseur/Onduleur Pur Sinus (350VA)	12
Servomoteurs :	12
2.3 Raspberry Pi :	13
3. Deuxième partie : station météo	15
3.1 La station météo	15
3.2 Utilisation de données	16
4. Troisième partie	17
4.1 Langages de programmation et base de données	17
Langage HTML	17
Langage CSS.....	17
Langage de programmation PYTHON	18
Langage de scripts PHP	18
Langage de scripts JavaScript	19
Base de données MySQL	19
4.2 Les programmes réalisés aux seins de la page web visualisés par les utilisateurs :	20
4.3 Les programmes réalisés aux seins de la page web, non visualisés par les utilisateurs :	21
4.2 Communication série avec python.....	21
5. Quatrième partie : Interface web graphique	22
Evolution de pages.....	22
Objectifs et travaux réalisés durant l'année 2013-2014	22

Objectifs et travaux réalisés durant l'année 2014-2015	23
Objectifs et travaux réalisés durant l'année 2015-2016	24
6. Sources.....	28
7. Liste des figures.....	29
8. Annexes	30
8.1 Annexe 1 : Planning prévisionnel du projet	30
8.2 Annexe 2 : Page WEB en html (TrackerPV12.php).....	31
8.3 Annexe 3 : programme CSS de la page WEB (StyleTracker.css)	36
8.4 Annexe 4 : calcul azimut, élévation et pilotage des moteur (Test_pilotage_moteur.py)	44
8.5 Annexe 5 : Programme de récupération des données de la station météo (DonneeMeteoFini.py)	46
8.6 Annexe 6 : Schéma de fonctionnement global	47

Introduction

Les énergies renouvelables s'imposent comme une solution d'avenir pour satisfaire de façon durable les besoins énergétiques mondiaux. Le solaire est une source d'énergie disponible, gratuite et illimitée de façon équitable en tout point de la planète.

Les panneaux solaires, sont donc un moyen d'exploitation d'énergie propre non polluante pour l'environnement d'un point de vue écologique. L'énergie solaire reste fiable et résistante par rapport aux autres énergies renouvelables car elle est produite sans nuisances sonores.

De plus, il préserve le paysage, sa faune et sa flore, ce qui lui permet d'être mis directement sur la toiture de nos bâtiments, que ce soit dans les villes ou dans les campagnes les plus reculées.

Le développement des énergies renouvelables (solaire photovoltaïque, solaire thermique, éolien, hydroélectricité, biomasse) s'impose aujourd'hui comme une évidence et l'énergie solaire photovoltaïque, sans être la solution miracle et unique, joue un rôle très importante.



Figure 1: Panneau photovoltaïque

Notre groupe est composé de 4 membres. Jean Laurent CABILLON est le chef de projet, Jacqueline ORDUNA VAZQUEZ est la secrétaire, Diana Carolina GARCIA ACO est la responsable de communication, Clément CLAVÉ est le responsable technique.

La distribution des rôles et le planning prévisionnel ont été réalisés en début de projet.

1. Présentation générale

1.1 Evolution du projet

Le projet a débuté il y a deux ans, les deux groupes précédents se sont focalisés sur le positionnement du tracker, conçu le système et géré la partie commande du mouvement à l'aide de la carte Arduino.

Ils ont aussi réalisé :

- Le montage de l'ensemble stockage et l'utilisation de l'énergie électrique,
- La programmation de la carte Arduino,
- L'acquisition des prises de mesures (intensité et tension du panneau, de la batterie et de la charge),
- La mise au point de la communication Ethernet entre le logiciel LabView et la carte Arduino,
- Une IHM (Interface Homme Machine) ergonomique sous LabView pour superviser le « tracker ».



Figure 2 : Interface Labview

L'année dernière, le groupe de projet 2014-2015 a remplacé l'ancienne architecture de la carte Arduino par un Raspberry pi équipé d'une carte Arduino nano pour faciliter la prise en main et libérer de la mémoire.

Ce changement a été effectué car ils devaient utiliser le langage C, un langage de programmation informatique système avec une architecture très lourde et complexe à réalisé.

Le travail devait également s'effectuer avec le système d'exploitation Windows.

Celui-ci traitait les commandes mais ne faisait pas leurs exécutions en temps réel, c'est-à-dire, qu'il fallait faire fonctionner tout le système afin d'avoir des informations souhaités. Cela prenait du temps et l'exécution des tâches n'était pas instantanée.

Par contre, un Raspberry pi B+ sous Linux avec une carte Arduino nano peut faire des mesures en temps réel avec des données plus précises et il n'est pas ralenti dans ces exécutions. C'est pour cela que le groupe de l'année dernière a choisi de remplacer la carte Arduino par le Raspberry pi avec une carte Arduino Nano.



Figure 3: Carte Arduino -> Raspberry pi B+ et Carte Arduino nano

Ensuite, ils ont remplacé l'interface Labview par une interface WEB. Cette interface a été remplacée car l'utilisateur devait posséder le logiciel Labview pour pouvoir accéder à la supervision.

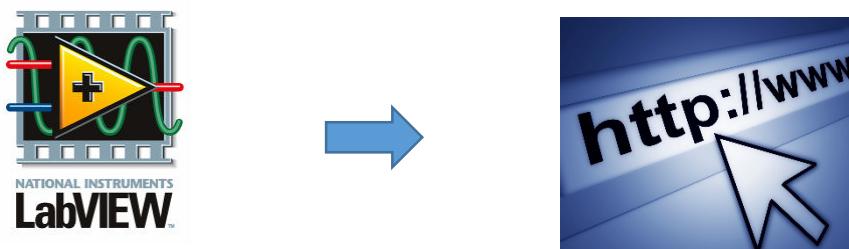


Figure 4: Logo Labview et image de navigateur

Par la suite, afin de déterminer la position du soleil, ils ont calculé une équation de la courbe du soleil en prenant en compte l'azimut pour l'axe vertical et l'élévation pour l'axe horizontal.

Enfin ils ont également :

- Refait la partie instrumentation à partir des études antérieures,
- Mis à jour le dossier technique,
- Débuté la programmation du Raspberry,
- Réalisé une interface web permettant l'accès aux informations répertoriées et aux commandes des lampes à partir du Raspberry.

1.2 Objectifs/ Travail à réaliser

Après avoir découvert les travaux réalisés par les équipes des années précédentes, nos principaux objectifs ont été de :

- Finaliser la partie instrumentation déjà en place,
- Ajouter un dispositif de mesure de la vitesse du vent,
- Finaliser l'interface web de supervision du tracker PV,
- Concevoir une interface web pour la visualisation des données enregistrées sur la base de données.

2. Première partie : Présentation des composants du projet

2.1 Panneau et armoire :

Dans la structure du panneau, nous avons divers composants avec des fonctions spécifiques à chacun :

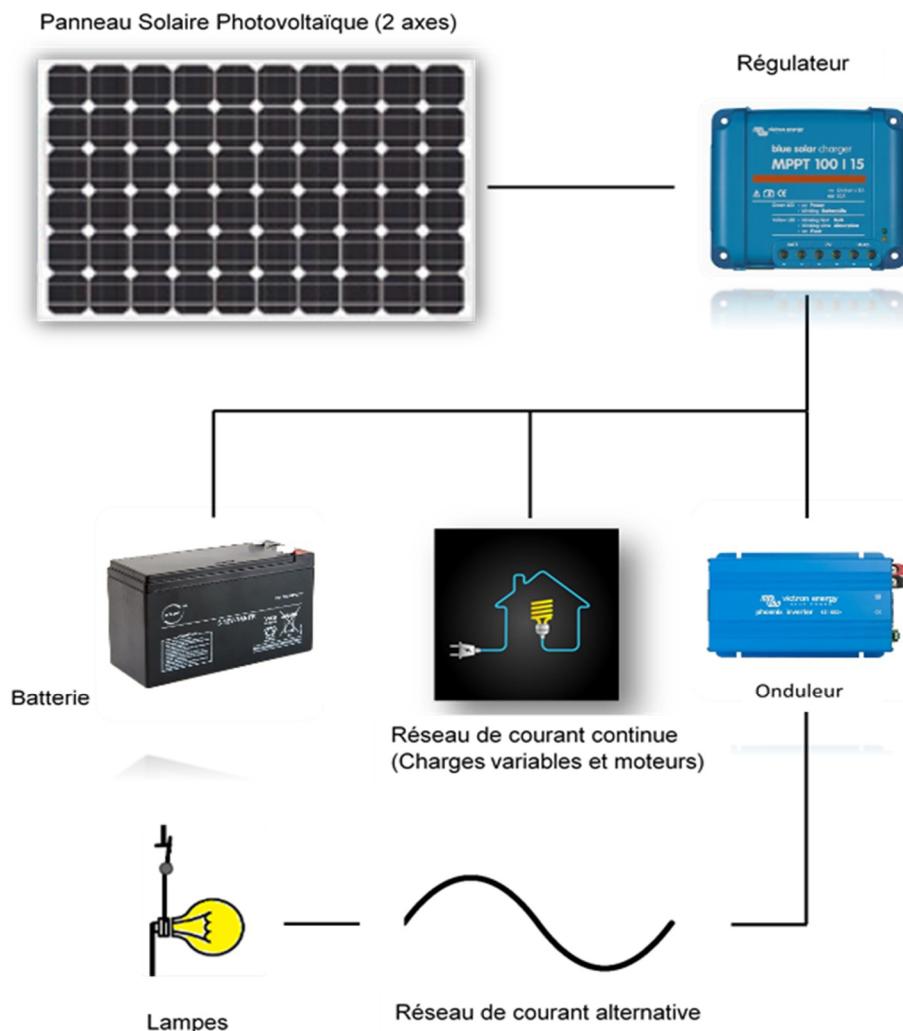


Figure 5 : schéma des composants

Panneau à 2 axes :

Le tracker solaire, nous sert de source d'énergie électrique continue de caractéristique 12V/150W et de surface 2m². Il a pour possibilité d'être monté sur deux axes motorisé permettant de suivre la course du soleil et ainsi optimiser son rendement.



Figure 6 : Tracker support de panneau

Batterie au Gel Victron 110Ah 12V :

Dû à l'intermittence des périodes de production, l'installation de la batterie a été nécessaire afin de réguler l'intensité. Elle a pour objectif de compenser l'énergie non produite par le tracker. La batterie au Gel Victron est composée de matériaux d'une grande pureté et de grilles de plomb/calcium qui garantissent à cette batterie au Gel un très faible taux d'autodécharge.

Régulateur électronique de charge solaire MPPT 15A:

Le régulateur est indispensable afin d'optimiser la charge de la batterie dû aux variations de production énergétique du tracker solaire. L'énergie de sortie sera utilisée pour l'alimentation d'un réseau de courant continu, par exemple, les servomoteurs et la batterie.



Figure 7: Régulateur

Convertisseur/Onduleur Pur Sinus (350VA) :



Figure 8 : Onduleur

Le convertisseur est alimenté par le circuit d'énergie continu, sa fonction principale est de convertir une tension de 12VDC à une tension 230VAC. Ce qui nous permettra d'alimenter des appareils électriques (ici les lampes).

Servomoteurs :

Il s'agit d'un moteur professionnel avec positionneur intégré et fond structural, pour la rotation automatique de cellule solaire dans un intervalle de temps adaptable. Ces appareils tournent la cellule solaire sur les axes horizontal et vertical permettant d'exploiter au maximum le rayonnement du soleil.

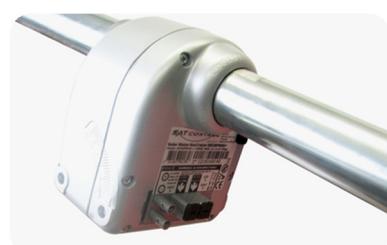


Figure 9: Servomoteur

Les moteurs solaires sont alimentés par une source externe, 12VDC, et peuvent résister à un vent de 130 km/h.

2.3 Raspberry Pi :

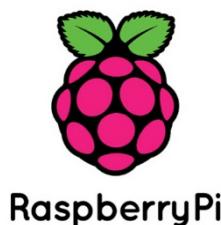


Figure 10 : Logo Raspberry Pi

Le Raspberry Pi est un nano ordinateur mono carte à processeur ARM conçu par le créateur de jeux vidéo David Braben, dans le cadre de sa fondation Raspberry Pi2. Cet ordinateur, à la taille d'une carte de crédit, a été développé dans le but d'encourager l'apprentissage de la programmation informatique. Il permet l'exécution de plusieurs variantes du système d'exploitation libre GNU/Linux et des logiciels compatibles.

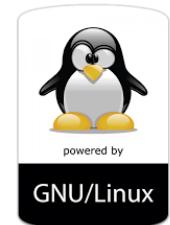


Figure 11: Logo Linux



Figure 12: Raspberry Pi B+

Le modèle utilisé pour le projet est le modèle B+ V1.2. Il possède un processeur ARM à 4 cœurs d'une fréquence d'horloge de 700MHz. Il est capable d'assigner 16 registres généraux (stockage temporaire des données) codés sur 32 bits. C'est-à-dire que les mots (signé ou non) manipulés par le processeur peuvent prendre jusqu'à plus de 4 milliard de valeurs (soit 2^{32}).

Sa mémoire vive (RAM) est de 512 Mo. La mémoire vive est l'espace principal de stockage du microprocesseur, mais le contenu disparaît lors de la mise hors tension de l'ordinateur.

Il a 40 broches GPIO, c'est-à-dire des entrées/sorties. Elles permettent de communiquer avec différents composants électroniques. Les connecteurs GPIO sont généralement alimentés en 3.3Vcc et ne peuvent émettre que des courants de faible capacité, allant de 3mA à 50mA. Cela lui offre d'être à la fois un ordinateur et un contrôleur pour des applications purement électroniques.

Pour permettre la liaison, la communication et la sauvegarde des données il possède également :

- Une prise d'alimentation micro-USB (700mA),
- 1 sortie stéréo jack,
- 4 ports USB2,
- 1 port réseau ETHERNET (de 10 à 100 Mbits/seconde),
- 2 sorties vidéos (HDMI et prise RCA),
- Une micro SD card,

Il a une consommation globale de 3 Watts.

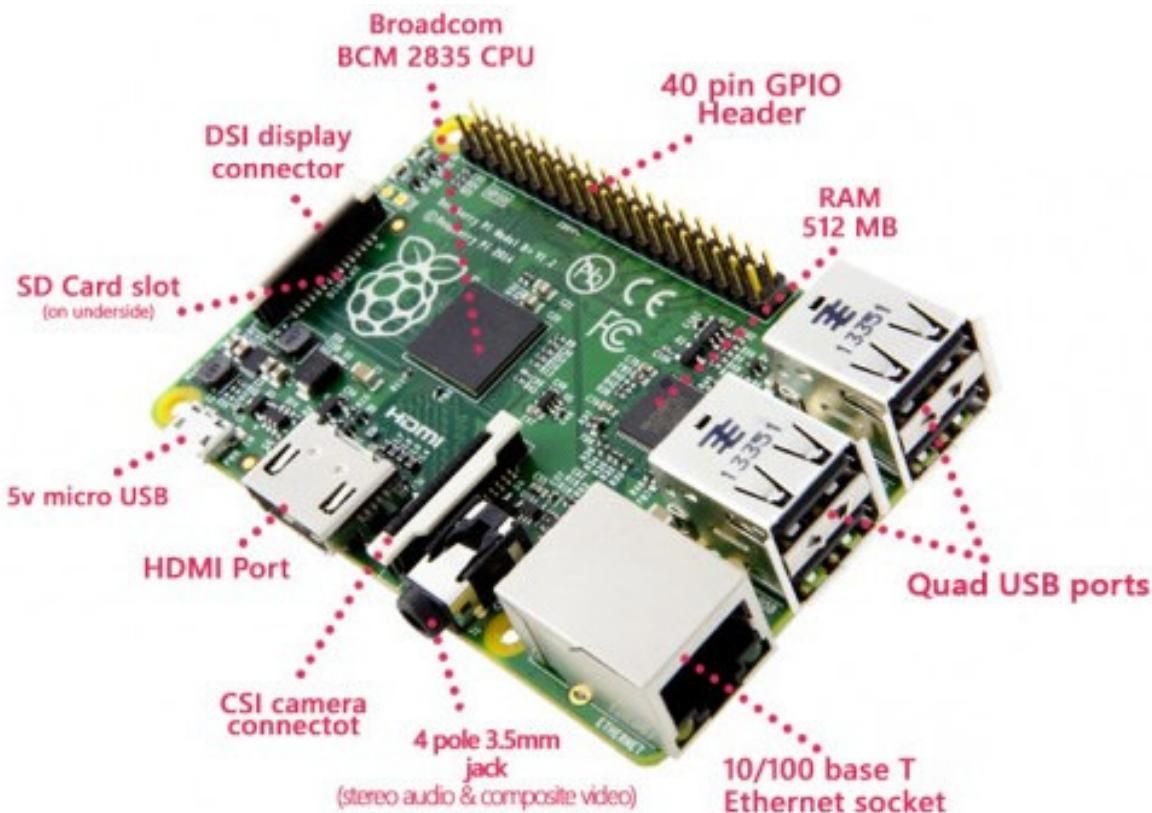


Figure 13: Description des composants d'un Raspberry Pi B+

L'utilisation du Raspberry Pi B+ V1.2 va nous permettre de faire de la programmation et de générer des pages web. En effet, il se comporte comme un micro-ordinateur et il est capable de créer des pages web sous format PHP ou HTML.

Le Raspberry pi va donc nous permettre de :

- Récupérer les données de la carte Arduino en python,
- Calculer la position du soleil avec python,
- Piloter les servomoteurs avec python,
- Gérer la récupération des données météo avec python,
- Rapatrier des données vers un fichier (Json ou .txt),
- Gérer les données de sécurité du système,
- Héberger une interface WEB avec HTML, PHP et CSS.

3. Deuxième partie : station météo

3.1 La station météo



Figure 14 Station météo compacte

Tout d'abord, une station météo sert à observer le temps qu'il fait et à le prévoir. Elle permet grâce à l'intermédiaire d'un ensemble de capteurs d'enregistrer et fournir des mesures physiques et des paramètres météorologiques liés aux variations du climat.

Pour ce projet, elle joue un rôle très important car elle nous permet de réaliser la fonction anémomètre comme demandé dans nos objectifs.

Cette station météo effectue des relevés en temps réels et elle est capable de faire une moyenne de ces relevés sur 10 min. On peut également relever le maximum et le minimum sur ces 10 min.

Pour la récupération de données de la station météo nous avons pris en compte les paramètres suivants:

- Température (actuelle et max)
- Humidité
- Pression atmosphérique
- Vitesse du vent
- Précipitation
- Rayonnement

Pour exploiter ces données on se connecte à la station, adresse IP : 10.3.208.225, puis nous récupérons les données voulu dans des registres, ces registres représentent des paramètres de météo (Température, humidité, pression, ...)



Figure 15 Station météorologique de l'IUT

Selon le manuel de fonctionnement de la station météo (Operating Manual Compact Weather Station), on obtient le numéro des registres pour chaque donnée météo voulue.

Pour les paramètres que l'on a choisis on obtient ce tableau récapitulatif :

Paramètre	Abréviation	Channel
<i>Température</i>		
Maximum	Tmax	33
Actuelle	T	31
<i>Humidité</i>		
Absolue Moyenne	AbsH	74
<i>Pression Atmosphérique</i>		
Relative	Patm	17
<i>Vitesse</i>		
Maximale (km/h)	Windmax	85
Moyenne (km/h)	Windavg	83
<i>Rayonnement</i>		
Global	Glradiation	27
<i>Précipitation</i>		
Intense (mm/h)	Précipitation	50
<i>Boussole</i>		
Compass	Compass	24

Figure 16 Localisation (Channel) des registres de la station météo

3.2 Utilisation de données

Les données rapatriées sur le *Raspberry*, nous permettrons par la suite de réaliser un tableau d'affichage des données météorologiques en temps réel dans notre Interface WEB et de réaliser une sécurité pour le panneau en fonction de la vitesse du vent.

Température (°C)	Température maximum (°C)	Rayonnement (lux)	Précipitation (mm)	Vitesse du vent (km/h)	Rafale de vent (km/h)	Pression Atmosphérique (Pa)	Humidité (%)
0	0	0	0	0	0	0	0

Figure 17: Tableau d'affichage des données météo

4. Troisième partie

4.1 Langages de programmation et base de données

Pour interagir avec le *Raspberry*, nous avons utilisé plusieurs outils informatiques afin de créer une page internet dans le but d'établir un dialogue homme machine (IHM). Tous ces langages de programmation nous permettent d'organiser le fonctionnement général du tracker (voir annexe 6).

Langage HTML

L'Hypertext Markup Language, généralement abrégé HTML, est le format de données conçu pour représenter les pages web. C'est un langage de balisage permettant d'écrire de l'hypertexte, d'où son nom.

HTML permet également de structurer sémantiquement et de mettre en forme le contenu des pages, d'inclure des ressources multimédias dont des images, des formulaires de saisie et des programmes informatiques. Il est souvent utilisé conjointement avec des langages de programmation (JavaScript) et des formats de présentation (feuilles de style en cascade).

Pour notre projet, l'utilisation de HTML a été principalement utilisé pour introduire toutes les informations nécessaires à l'utilisateur (images, données météo, commandes des lampes, ...)



Figure 18: Logo HTML

Langage CSS



Les feuilles de style en cascade, généralement appelées CSS de l'anglais Cascading Style Sheets, est un langage informatique qui décrit la présentation des documents HTML et XML.

On l'utilisera comme outils de décoration des pages HTML afin de rendre la communication simplifié et d'avoir une meilleure visualisation d'ensemble.

Figure 19: Logo CSS

Langage de programmation PYTHON

Python est un langage de programmation de licence libre. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire et fonctionne sur la plupart des plates-formes informatiques, des supercalculateurs ou ordinateurs.



Figure 20: Logo Python

Il est conçu pour optimiser la productivité des programmeurs en offrant des outils de haut niveau et une syntaxe simple à utiliser pouvant procréer à des programmes très simples ou des programmes complets comme :

- des jeux,
- des suites bureautiques,
- des logiciels multimédias,
- des clients de messagerie,
- des progiciels (ensemble de plusieurs logiciels pouvant fonctionner ensemble, principalement utilisés dans le monde professionnel).

Il est, en outre, très facile d'étendre les fonctionnalités existantes. Il existe ce qu'on appelle des bibliothèques qui aident le développeur à travailler sur des projets particuliers.

En effet, cette fonctionnalité nous a permis de faire circuler des informations de la station météo au Raspberry via des bibliothèques.

Langage de scripts PHP

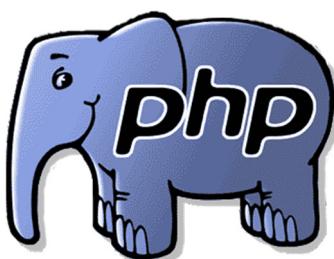


Figure 21: Logo PHP

HyperText Preprocessor, plus connu sous son sigle PHP (acronyme récursif), est un langage de programmation libre et un impératif orienté, principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale.

Il peut générer un dossier contenant les commandes transmises au Raspberry, il est utilisé essentiellement pour les commandes charges du tracker (lampes). Celles-ci se déclenchent lors de l'appui sur les éléments interactifs de l'interface WEB.

Au cours de notre projet, il nous a servi pour la création des fichiers qui commandent le micro serveur Raspberry. Par la suite, il permet de gérer les exécutions dans l'interface Web pour nous permettre, en cliquant sur certains éléments, de piloter les charges.

Langage de scripts JavaScript

JavaScript est un langage de programmation de scripts principalement employé dans les web interactifs mais aussi pour les serveurs, il permet sous forme de code de dicter à l'ordinateur comment exécuter des fonctions.



Figure 22: Logo JavaScript

Lors du projet, le code JavaScript dans la page Web nous sert à dynamiser la page, sans pour autant réactualiser l'ensemble de la page. Cet outil nous permet d'avoir une meilleure fluidité lors de l'exécution des ordres (lampes).

Base de données MySQL



Figure 23: Logo MySQL

MySQL est un Système de Gestion de Bases de Données (SGBD). Il se charge du stockage des informations (liste des messages, des membres,...) de manière simplifiée permettant ainsi l'extraction de ces données plus simplifier.

A l'aide de ce logiciel, nous pouvons enregistrer la liste des informations qui seront par la suite ordonnée sur la page web.

4.2 Les programmes réalisés aux seins de la page web visualisés par les utilisateurs :

- Commande des lampes (nom fichier « bouton-terminé-fct.php » et « lampe(1/2/-1/-2).php ») :
 - o Afin de réaliser cette commande, il nécessite l'intervention d'outils informatique (HTML, CSS, PHP, JavaScript). En effet, le logiciel HTML va créer les boutons, le CSS va les mettre en forme, le JavaScript va rendre dynamique la page et lancer les lignes de codes PHP, lors de l'appui du bouton. Le PHP va inscrire les commandes dans un fichier « CdeWS.txt » qui va être lu et exécuté par le programme « Capteurs_test.py ».

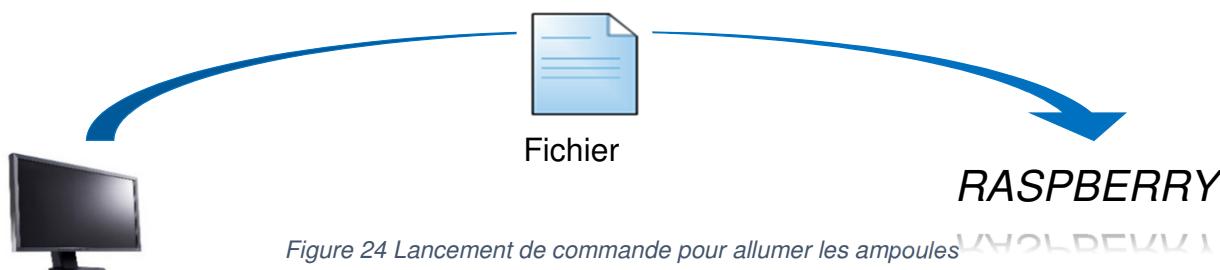


Figure 24 Lancement de commande pour allumer les ampoules

- Commande des servomoteurs

Pour commander les servomoteurs nous avons trois modes de fonctionnement:

- o Le mode automatique :

Grâce au programme « Test_pilotage_moteur.py » nous pouvons piloter les servomoteurs pour qu'ils suivent la courbe du soleil automatiquement. (Voir annexe 4)

- o Le mode manuel :

Les utilisateurs ont à leur disposition des curseurs de positionnement en degrés, qui déclenche un fichier PHP, écrivant les valeurs de positionnement dans un fichier (« CdeWE.txt » pour l'élévation, celui pour l'azimut n'a pas encore été créé). Ce fichier doit être lu et exécuté par les moteurs.

- o Le mode sécurité :

Lorsque la vitesse du vent dépasse une vitesse de 100 km/ h, un programme ordonne le positionnement du panneau du tracker à l'horizontal (élévation à 90°), il reste ainsi tant que les rafales de vent dépassent la vitesse de sécurité.

- Page Web (TrackerPV12.php) :
 - o Cette page regroupe pratique tous les logiciel utilisés (HTML, CSS, PHP, JavaScript), sa programmation permet aux clients de connaitre suivant l'onglet, la météo actuel, commander les moteurs et les lampes, cependant afin d'éviter certain manipulation néfaste, ses capacités sur celui-ci reste restreinte. (Voir Annexe 2 et 3)

4.3 Les programmes réalisés aux seins de la page web, non visualisés par les utilisateurs :

- Récupération des données de la station météo (DonneeMeteoFini.py) :
 - o Le Raspberry va lancer le programme python et se connecter à la station météo (IP : 10.3.208.225) et suivant les valeurs enregistrées va acheminer les valeurs de la station vers un fichier « dataSW.json ». Ces données vont ensuite être récupérées par le code Java Script de la page WEB (« trackerPV12.php »). (Voir annexe 5)
- Convertisseur des valeurs calculées en points (Test_pilotage_moteur.py) :
 - o A l'aide du logiciel Python, nous relevons la date et l'heure du jour, qui va calculer à l'aide d'une équation la position du soleil (élévation, azimut) puis convertir ces valeurs en nombre de points pour permettre le pilotage des moteurs.

4.4 Communication série avec python

La carte Arduino nano a comme principal fonctionnalité de récupérer le message transmis via Python. La carte est connectée par son port USB, détecté comme port série par Python. Cet ainsi qu'elle permet de récupérer les données mesuré et de permettre au Raspberry d'économiser de la mémoire.

5. Quatrième partie : Interface web graphique

Evolution de pages

Objectifs et travaux réalisés durant l'année 2013-2014

Ils avaient pour objectif de programmer l'interface sur le logiciel LabVIEW pour superviser les différents composants de l'installation et d'édifier le panneau solaire.

Les points faibles sur la première supervision sont les suivantes :

- L'utilisateur doit posséder le logiciel LabVIEW pour pouvoir accéder à la supervision,
- Il n'y a pas de commande pour piloter les lampes et les moteurs du tracker.

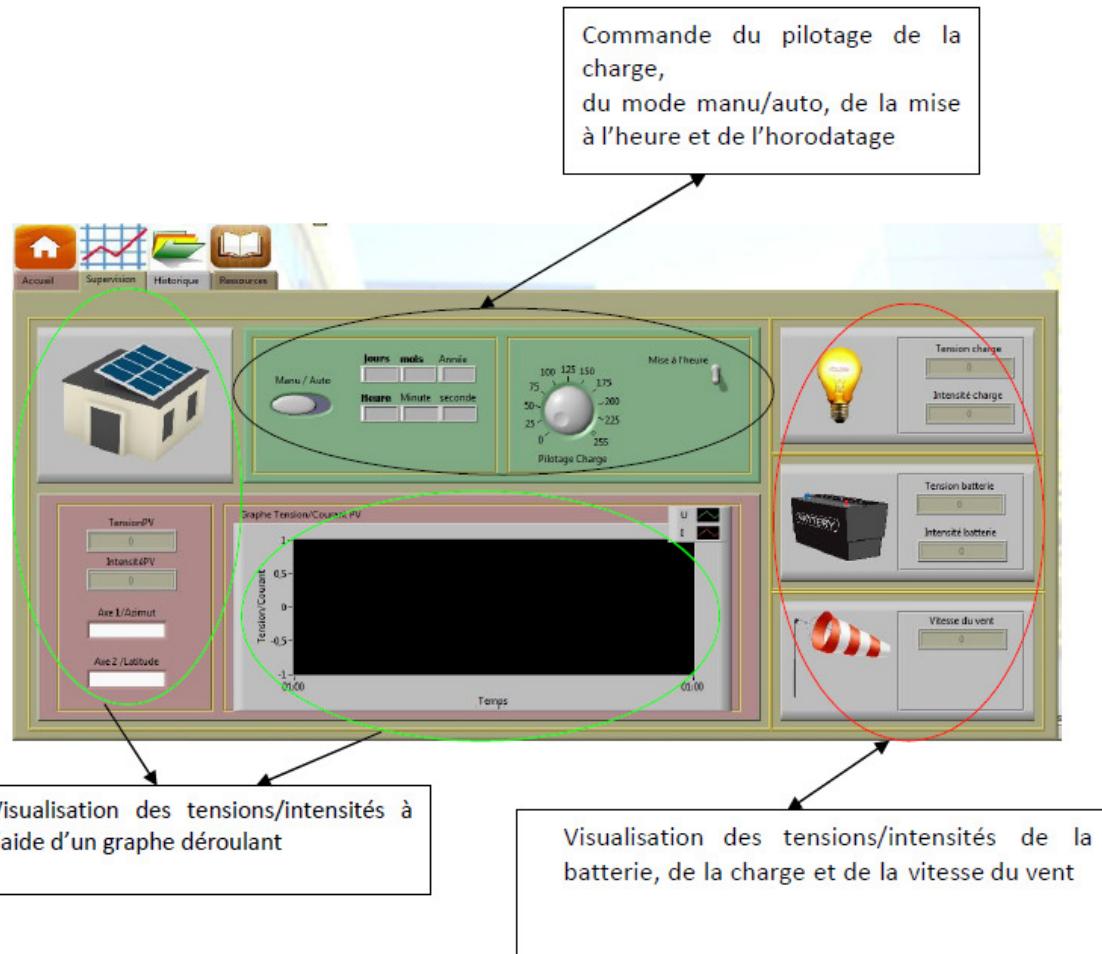


Figure 25: Description interface de supervision Labview

C'est pour cela que l'utilisation du logiciel Labview a été abandonnée.

Objectifs et travaux réalisés durant l'année 2014-2015

Pour remédier au problème de l'interface LabVIEW les étudiants de l'année 2014-2015 ont conçu un prototype de supervision à travers une page web. Ils ont dû assimiler et maîtriser les langages web HTML, CSS et PHP afin de réaliser cette interface. Pour mettre en place des indicateurs de courant, de tension et les boutons de commande des lampes, ils ont fait un schéma de principe de l'installation afin de pouvoir visualiser le fonctionnement.

Cependant une actualisation globale de la page était nécessaire pour permettre l'activation des commandes (lampes). Du coup quand les interrupteurs étaient en action (lampes), une autre page internet apparaissait pour changer l'image du montage.

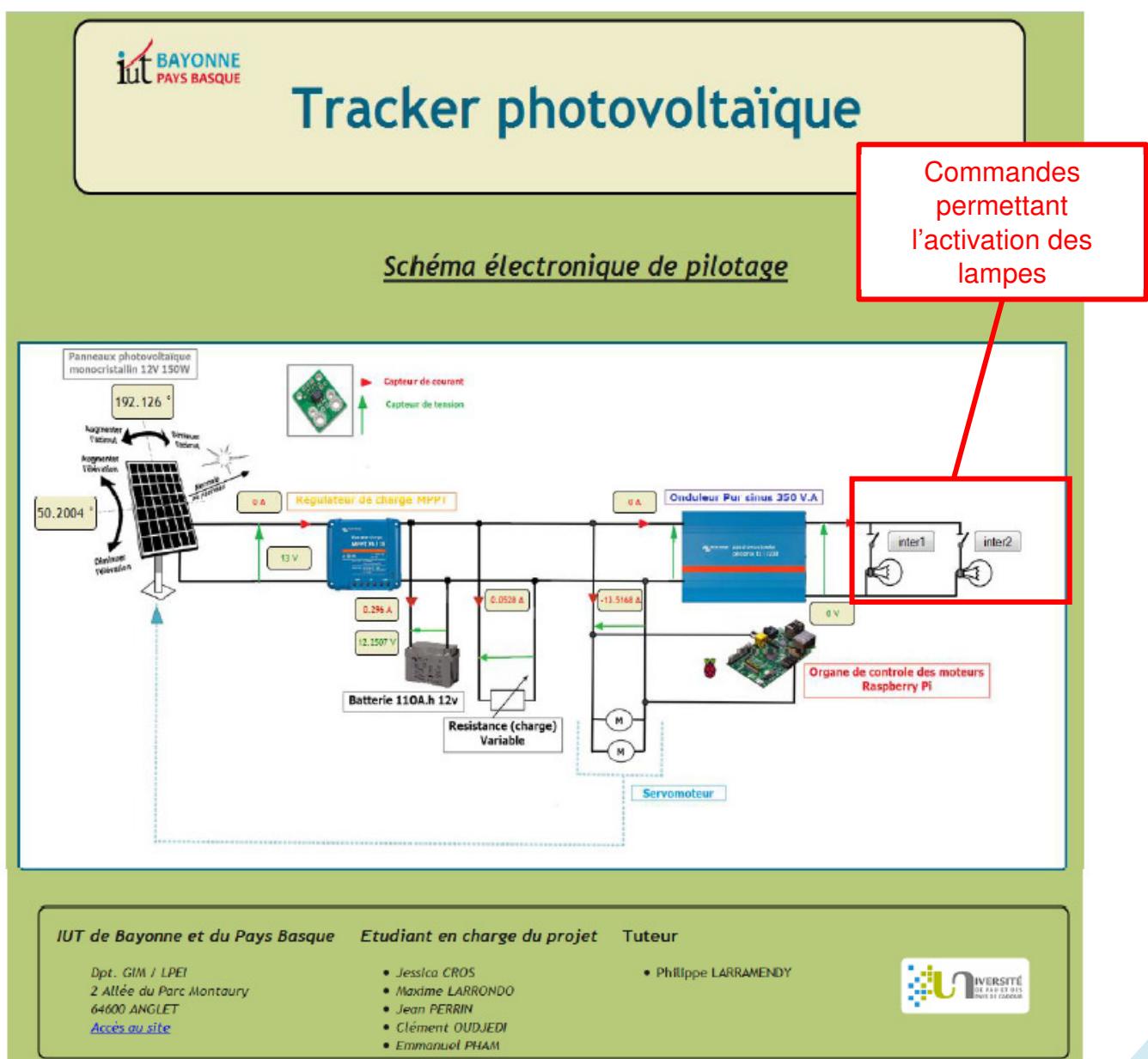


Figure 26 : Interface WEB 2014 -2015

Points faibles

- Pas de visualisation des données météo,
- La page doit être rechargée pour allumer les lampes,
- Erreurs dans le schéma,

Objectifs et travaux réalisés durant l'année 2015-2016

Cette année notre équipe a décidé de faire une page web plus dynamique suivant tout de même le travail réalisé l'année précédent, on a assimilé et maîtrisé les langages web HTML, CSS, java script, python et PHP, comme précisé dans la partie programmation .

Pour nous permettre de conserver une idée fixe, nous avons réalisé un croquis nous permettant de retenir un objectif à accomplir.

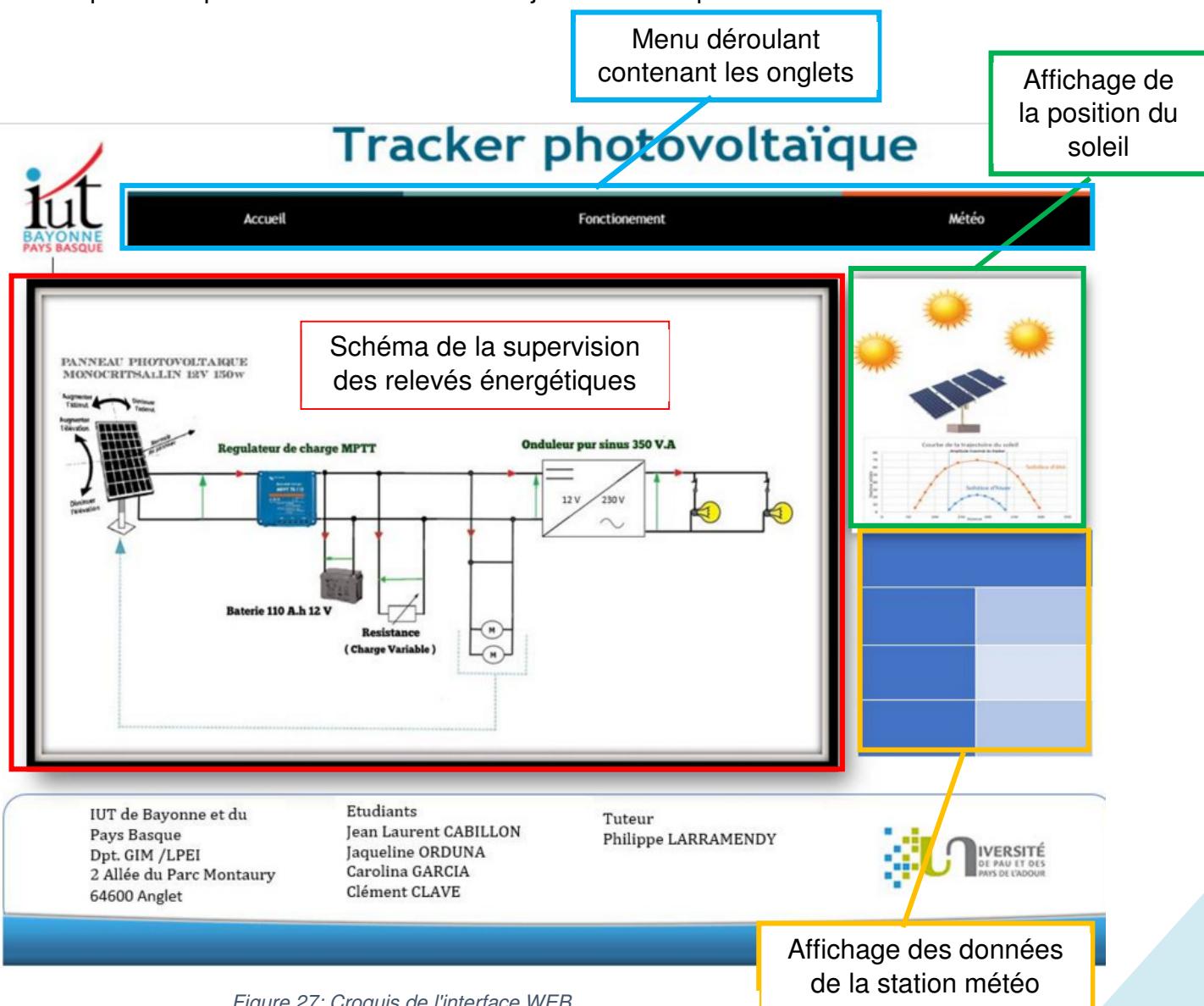


Figure 27: Croquis de l'interface WEB

Dans ce schéma, on souhaitait installer le schéma principal, composé du schéma simplifié du panneau solaire, de son positionnement en fonction du soleil et quelques données recueillies de la station météo.

En pied de page on pensait insérer la présentation des étudiants, du tuteur du projet et les coordonnées de l'IUT.

En se basant sur notre croquis, nous avons créé un menu déroulant permettant d'ouvrir des pages contenant différents paramètres :

- Accueil : comportant la supervision des données récupérées du panneau solaire, la commande des lampes, la position du soleil actuelle et les données récupérées de la station météo actuelle.
- Fonctionnement : comportant des curseurs permettant de bouger manuellement le tracker,
- Météo : comportant les prévisions météo à 5 jours de la ville d'Anglet.

Les résultats obtenus nous ont permis d'obtenir cet affichage pour l'accueil :

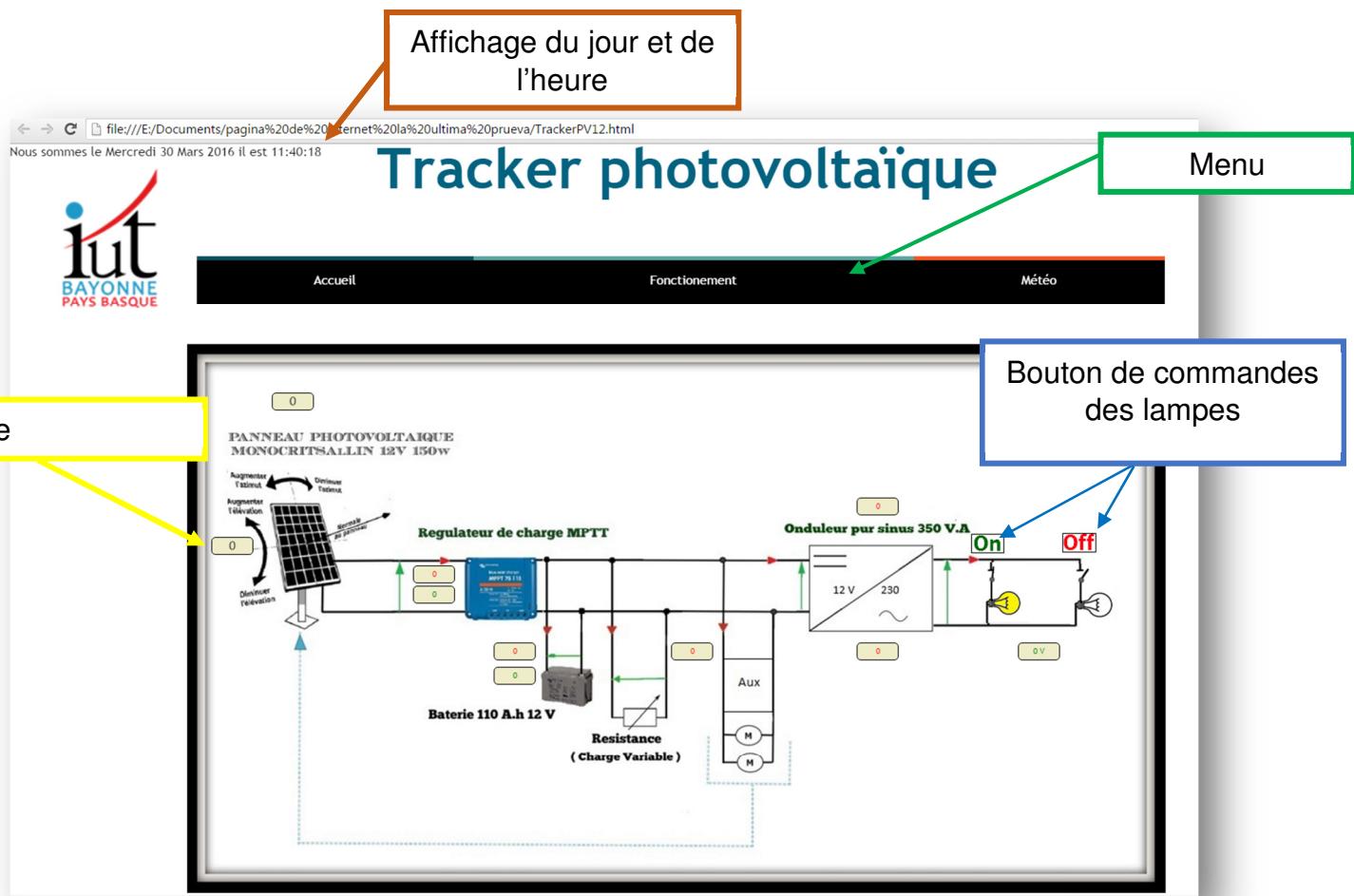


Figure 28 : Page d'accueil

Nous avons installé des boutons pour commander deux lampes, qui ont pour objectif de tester la capacité du panneau solaire en charges, ces boutons sont commandés en Java Script ce qui permet de déclencher un fichier PHP sans actualiser la page.

Dans la deuxième partie de l'accueil, on a créé un tableau qui permet d'afficher à l'utilisateur les données de la station météo, on récupère depuis la station météo les mesures de :

- Température,
- Température maximum,
- Rayonnement,
- Précipitation,
- Vitesse du vent,
- Vitesse des rafales de vent,
- Pression atmosphérique,
- Humidité,

Afin que l'utilisateur puisse connaître la position du soleil au-dessus de l'IUT en temps réel. Nous avons ajouté un affichage satellite, grâce au site Sunearthtools, pour observer le positionnement du soleil à l'IUT de Bayonne en temps réel.

Température (°C)	Température maximum (°C)	Rayonnement (lux)	Précipitation (mm)	Vitesse du vent (km/h)	Rafale de vent (km/h)	Pression Atmosphérique (Pa)	Humidité (%)
0	0	0	0	0	0	0	0

Données de la station météo

recherche SunRise: 06:50:50 * 83.76° | SunSet: 19:30:44 * 276.52° | 43° 28' 40.800" N 1° 30' 39.600" E

Solar Disk Solstice 9 Rue Edouard Barrère, 64600 Anglet, France

Gmail for Work Véhiculez une image plus professionnelle grâce aux e-mails personnalisés de Google Apps

Commencez un essai gratuit

année mois jour heure minute

2016 ▾ 03 ▾ 30 ▾ 11 ▾ 40 ▾

Time zone GMT 0 ▾ DST Default

exécuter

Click here to see the complete map

Plan Satellite

Positionnement du soleil

Coordonnées

IUT de Bayonne et du Pays Basque Etudiant en charge du projet Tuteur

Dpt. GIM / LPEI
2 Allée du Parc Montaury
64600 ANGLET

• Jean Laurent CABILLON
• Jacqueline ORDUNA
• Carolina GARCIA
• Clément CLAVÉ

• Philippe LARRAMENDY

[Accès au site](#)

Figure 29 : Deuxième partie de l'onglet accueil

Ensuite dans la partie fonctionnement, nous avons commencé la création d'un curseur permettant de bouger le panneau manuellement, pour l'instant ce curseur est simplement pour l'élévation.

The screenshot shows a web page titled "Tracker photovoltaïque". At the top, there are three tabs: "Accueil", "Fonctionnement" (which is highlighted in red), and "Météo". Below the tabs, there is a form with the following fields:

- "Vos coordonnées" (Your coordinates) - A large input field containing placeholder text: "Ne remplir qu'une seule commande" (Fill only one command).
- "Elévation:" - A slider with a downward arrow icon, currently set to 90.
- "Envoyer" (Send) - A button at the bottom left of the form.

Figure 30 : Onglet Fonctionnement

Enfin dans l'onglet météo, on a inséré un dispositif de visualisation du climat d'Anglet sur une période de 5 jours s'actualisant à chaque connexion.



Figure 31: Onglet météo

Prévisions météorologiques à 5 jours d'Anglet

Améliorations possibles :

- Créer une visualisation de la position du tracker en fonction du soleil,
- Améliorer l'ergonomie de l'interface grâce au CSS,
- Finaliser le pilotage manuel,
- Améliorer le codage,
- Créer un mot de passe pour protéger le système,
- Insérer une météo en français,

6. Sources

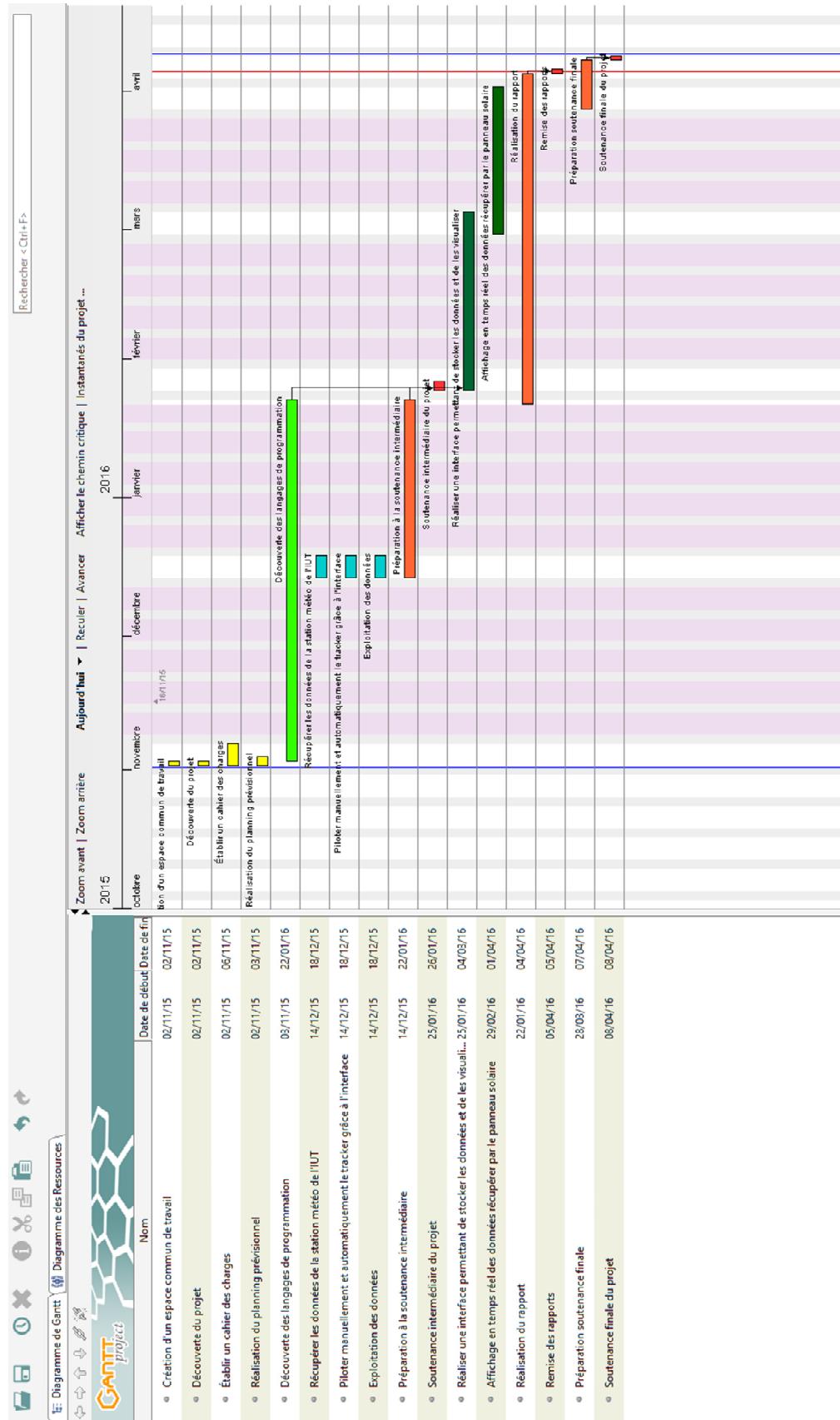
- <https://openclassrooms.com/>
- <https://www.youtube.com/>
- <http://www.i-programmer.info/>
- <http://le-blog-facile.ek.la/mettre-un curseur-personnalisé-p274064/>
- Operating Manual English Compact Weather Station, Document Technique de la Station météo de l'IUT de Bayonne
-

7. Liste des figures

Figure 1: Panneau photovoltaïque.....	6
Figure 2 : Interface Labview	8
Figure 3: Carte Arduino --> Raspberry pi B+ et Carte Arduino nano	9
Figure 4: Logo Labview et image de navigateur	9
Figure 5 : schéma des composants.....	11
Figure 6 : Tracker support de panneau.....	11
Figure 7: Régulateur	12
Figure 8 : Onduleur.....	12
Figure 9: Servomoteur	12
Figure 10 : Logo Raspberry Pi.....	13
Figure 11: Logo Linux	13
Figure 12: Raspberry Pi B+	13
Figure 13: Description des composants d'un Raspberry Pi B+.....	14
Figure 14 Station météo compacte.....	15
Figure 15 Station météorologique de l'IUT	15
Figure 16 Localisation (Channel) des registres de la station météo	16
Figure 17: Tableau d'affichage des données météo	16
Figure 18: Logo HTML.....	17
Figure 19: Logo CSS	17
Figure 20: Logo Python	18
Figure 21: Logo PHP	18
Figure 22: Logo JavaScript.....	19
Figure 23: Logo MySQL.....	19
Figure 24 Lancement de commande pour allumer les ampoules	20
Figure 25: Description interface de supervision Labview.....	22
Figure 26 : Interface WEB 2014 -2015	23
Figure 27: Croquis de l'interface WEB.....	24
Figure 28 : Page d'accueil	25
Figure 29 : Deuxième partie de l'onglet accueil	26
Figure 30 : Onglet Fonctionnement	27
Figure 31: Onglet météo	27

8. Annexes

8.1 Annexe 1 : Planning prévisionnel du projet



8.2 Annexe 2 : Page WEB en html (TrackerPV12.php)

```

1  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"> <!--Declarat
2  <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">
3
4  <head>
5    <title>tracker</title>
6  </head>
7
8  <header>
9    <figure> <!-- Image importante -->
10    <!--logo -->
11   </figure>
12
13   <h1> Tracker photovoltaïque</h1> <!-- Titre important -->
14
15 </header> <!-- Fin de l'en-tête de la page -->
16
17
18   <meta charset="utf-8" /> <!-- Pour prendre en compte presque tous les symboles -->
19   <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
20   <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
21
22   <link rel="stylesheet" href="styletracker.css" /> <!-- Pour introduire css -->
23   <script type="text/javascript" src="date_heure.js"></script> <!-- Pour introduire l'heure -->
24
25 <body>
26   <span id="date_heure"></span>
27   <script type="text/javascript">window.onload = date_heure('date_heure');</script>
28 </body>
29
30
31 <!-- codage javascript -->
32 <script language="javascript">
33   function loadJSON()
34   {
35     var data_file = "/montmpfs/datasW.json";
36     var http_request = new XMLHttpRequest();
37     try{
38       // Opera 8.0+, Firefox, Chrome, Safari
39       http_request = new XMLHttpRequest();
40     }catch (e){
41       // Internet Explorer Browsers
42       try{
43         http_request = new ActiveXObject("Msxml2.XMLHTTP");
44       }catch (e){
45         try{
46           http_request = new ActiveXObject("Microsoft.XMLHTTP");
47         }catch (e){
48           // Something went wrong
49           alert("Your browser broke!");
50           return false;
51         }
52       }
53     }
54     http_request.onreadystatechange = function(){
55       if (http_request.readyState == 4 )
56       {
57         //Javascript function JSON.parse to parse JSON data
58         var jsonObj = JSON.parse(http_request.responseText);
59
60         jsonObj variable now contains the data structure and can
61         document.getElementById("azimut").innerHTML = jsonObj.azimut;
62         document.getElementById("elevation").innerHTML = jsonObj.elevation;
63         document.getElementById("Valt").innerHTML = jsonObj.Valt;
64         document.getElementById("Ialt_pv").innerHTML = jsonObj.Ialt_pv;
65         document.getElementById("Vpv").innerHTML = jsonObj.Vpv;
66         document.getElementById("Vbatt").innerHTML = jsonObj.Vbatt;
67         document.getElementById("Ibatt").innerHTML = jsonObj.Ibatt;
68         document.getElementById("Iaux").innerHTML = jsonObj.Iaux;
69         document.getElementById("Ichg").innerHTML = jsonObj.Ichg;
70         document.getElementById("Iconv").innerHTML = jsonObj.Iconv;
71         document.getElementById("T").innerHTML = jsonObj.T;
72         document.getElementById("Tmax").innerHTML = jsonObj.Tmax;
73         document.getElementById("GLRad").innerHTML = jsonObj.GLRad;
74         document.getElementById("Precipitation").innerHTML = jsonObj.Precipitation;
75         document.getElementById("Windact").innerHTML = jsonObj.Windact;
76         document.getElementById("Patm").innerHTML = jsonObj.Patm;
77         document.getElementById("AbsH").innerHTML = jsonObj.AbsH;
78         document.getElementById("Compass").innerHTML = jsonObj.Compass;
79         document.getElementById("Wind_direction").innerHTML = jsonObj.Wind_direction;
80       }
81       http_request.open("GET", data_file, true);
82       http_request.send();
83       setTimeout('loadJSON();','300');
84     }
85   }

```

Titre page

Déclaration pour que les documents web s'affichent et fonctionnent correctement dans les navigateurs Conformes aux normes, Comme Mozilla, IE5 / Mac, et IE6 /

Affichage

JSON est un sous-ensemble de JavaScript, le compilateur analyse correctement le texte et produit une structure d'objet.

Communication données station

```

85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105 //---codage nouvelle lampe-----
106 //---Commande bouton 1-----
107
108
109 function ImageClick() {
110     var img = document.getElementById('myImage');
111     var xhr = getXhr();
112
113 //xhr.onreadystatechange = function(){
114     if (img.alt=="off") {
115         img.src = "on.png";
116         img.alt = "on";
117         var img = document.getElementById('l1');
118         img.src="lpon.png"
119
120         xhr.open("GET","lampe1.php",true);
121         xhr.send(null);
122
123     }
124 }
125
126
127 else {
128     img.src = "off.png";
129     img.alt = "off";
130     var img = document.getElementById('l1');
131     img.src="lpooff.png"
132
133     xhr.open("GET","lampe-1.php",true);
134     xhr.send(null);
135
136 }
137
138 }
139
140 //---Commande bouton 2-----
141
142 function ImageClick2() {
143     var img = document.getElementById('myImage2');
144     var xhr2 = getXhr();
145
146 //xhr.onreadystatechange = function(){
147     if (img.alt=="off") {
148         img.src = "on.png";
149         img.alt = "on";
150         var img = document.getElementById('l2');
151         img.src="lpon.png"
152
153         xhr2.open("GET","lampe2.php",true);
154         xhr2.send(null);
155
156     }
157 }
158
159
160 else {
161     img.src = "off.png";
162     img.alt = "off";
163     var img = document.getElementById('l2');
164     img.src="lpooff.png"
165
166     xhr2.open("GET","lampe-2.php",true);
167     xhr2.send(null);
168
169 }
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
789
789
790
791
792
793
794
795
796
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
897
898
899
899
900

```

Création bouton

```

166         xhr2.open("GET","lampe-2.php",true);
167         xhr2.send(null);
168     }
169 }
170
171 }
172
173 //-----//
174
175 </script>
176
177
178
179 <!-- codage javascript -->
180
181 <?php
182 // on ouvre le fichier de commandes
183 $monfichier = fopen("montmpfa/CdeWS.txt", 'w+'); // ouvre les fichiers en supprimer le contenu
184 fputs($monfichier, '256.2.2.1.0.0.2.2!'); // activation de K3, K4 et K5. Les autres commandes sont inchangées
185 // quand on a fini de l'utiliser, on ferme le fichier
186 fclose($monfichier);
187 ?>
188
189
190 <!--<![pages]-->
191 <nav>
192 <ul>
193     <li><a href="TrackerPV12.html"><span class="primero"><i class="icon icon-home"></i></span> Accueil</a></li>
194     <li><a href="fonctionnement.html"><span class="segundo"><i class="icon icon-books"></i></span> Fonctionnement</a></li>
195     <li><a href="meteopagmejorada.html"><span class="tercero"><i class="icon icon-sun"></i></span> Météo</a></li>
196
197 </ul>
198 </nav>
199 </header>
200
201
202
203 <!--Donnes station météo-->
204 <body>
205
206     <div id="azimut">
207         <p>0</p>
208         <p>0</p>
209     </div>
210
211     <div id="elevation">
212         <p>0</p>
213     </div>
214
215     <div id="vatt">
216         <p>0</p>
217     </div>
218
219     <div id="Ialt_pv">
220         <p>0</p>
221     </div>
222
223     <div id="Vpx">
224         <p>0 V</p>
225     </div>
226
227     <div id="Vbatt">
228         <p>0</p>
229     </div>
230
231     <div id="Ibatt">
232         <p>0</p>
233     </div>
234
235     <div id="Jaux">
236         <p>0</p>
237     </div>
238
239     <div id="Ichg">
240         <p>0</p>
241     </div>
242
243     <div id="Iconv">
244         <p>0</p>
245     </div>
246
247
248     <table border="1" id="tableau">
249         <tr >
250             <th>Température (*c)</th>

```

Fichier commande

Parties du menu

```

251 <th> Température maximum (°C) </th>
252 <th>Rayonnement (lux) </th>
253 <th>Précipitation (mm)</th>
254 <th>Vitesse du vent (km/h) </th>
255 <th>Rafale de vent (km/h) </th>
256 <th>Pression Atmosphérique (Pa) </th>
257 <th>Humidité (%) </th>
258 </tr>
259 <tr align=middle >
260 <td>
261 <div id="T">
262 <p>0</p>
263 </div>
264 </td>
265 <td>
266 <div id="Tmax">
267 <p>0</p>
268 </div>
269 </td>
270 <td>
271 <div id="GlRad">
272 <p>0</p>
273 </div></td>
274 <td>
275 <div id="Precipitation">
276 <p>0</p>
277 </div></td>
278 <td>
279 <div id="Windact">
280 <p>0</p>
281 </div>
282 </td>
283 <td>
284 <div id="Windmax">
285 <p>0</p>
286 </div>
287 </td>
288 <td>
289 <div id="Ratm">
290 <p>0</p>
291 </div>
292 </td>
293 <td>
294 <div id="AbsH">
295 <p>0</p>
296 </div>
297 </td>
298 </tr>
299
300 <tr align=middle>
301 <td></td>
302 <td></td>
303 <td></td>
304 <td></td>
305 <td></td>
306 <td></td>
307 <td></td>
308 <td></td>
309 </tr>
310 </table>
311

```

Affichage des données de la station météo dans un tableau

Images tableau

```

313   <form action="TrackerPV2.php" method="post">
314     ...
315   </form>
316
317   <form action="TrackerPV1.php" method="post">
318     ...
319   </form>
320
321   <p></p> <!-- Image du montage -->
322
323
324   <!-- Position du soleil -->
325   <div id="position">
326     <iframe frameborder='0' height='750px' width='600px' src='http://www.sunearthtools.com/tools/sun-position-widgetIF.php?point=43.478,-1.511&modeMapH=7'>
327   </div>
328   <!-- Fin position du soleil -->
329
330
331   <!-- codage lampes -->
332
333
334   <section id="lampe1CJL">
335
336     <img src='lpoff.png' id='11' />
337     
338     <span id="resultat"></span>
339   </section>
340
341
342   <section id="lampe2CJL">
343
344     
345     <span id="resultat"></span>
346     <img src='lpoff.png' id='12' />
347   </section>
348
349   </body> <!-- Fin du corps de la page -->
350
351   <footer> <!-- Pied de page -->
352
353     <div id="Info-iut"> <!-- Liste 1 -->
354       <p> IUT de Bayonne et du Pays Basque</p>
355       <ul>
356         <li>Dpt. GIM / LPEI</li>
357         <li>Dpt. GIM / LPEI</li>
358         <li>2 Allée du Parc Montaury</li>
359         <li>64600 ANGLET</li>
360         <li>Pour mettre un lien vers page internet iut -->
361           <li><a href="https://www.iutbayonne.univ-pau.fr">Accès au site</a></li>
362         </ul>
363     </div>
364
365     <div id="Etudiant"> <!-- Liste 2 -->
366       <p> Etudiant en charge du projet </p>
367       <ul>
368         <li> Jean Laurent CABILLON </li>
369         <li> Jacqueline ORDUNA </li>
370         <li> Carolina GARCIA </li>
371         <li> Clément CLAVÉ </li>
372       </ul>
373     </div>
374
375     <div id="Tuteur"> <!-- Liste 3 -->
376       <p> Tuteur</p>
377       <ul>
378         <li>Philippe LARRAMENDY </li>
379       </ul>
380     </div>
381
382     <figure> <!-- Image importante -->
383       
384     </figure>
385
386     </footer> <!-- Fin du pied de page -->
387     <script type="text/javascript">window.onload = loadJSON();</script>
388
389   </html> <!-- Fin du code html -->

```

Lien de la page web pour l'insertion de la position du soleil

Images des lampes et des boutons

Pied de page

8.3 Annexe 3 : programme CSS de la page WEB (StyleTracker.css)

```

1  *{
2      margin:0;
3      padding:0;
4  }
5
6  /* Définition des polices personnalisées */
7  body{
8      background-color: "white"; /* Couleur de fond*/
9      font-family: 'Trebuchet MS', Arial, sans-serif; /* Police*/
10     color: #181818; /* Couleur du texte*/
11 }
12 header{
13     position: absolute;
14     width: 2000px; /* Largeur de la boîte*/
15     /*height: 180px;    Hauteur de la boîte*/
16     background-color: "white";
17     border-radius: 20px; /* Arroondir l'encadrement*/
18     display: inline-block; /* Type d'élément*/
19     margin-left: 75px; /* Marge à gauche*/
20 }
21
22 header h1{
23     position: absolute;
24     font-size: 70px; /* Taille du texte*/
25     font-weight: bold; /* Mettre en gras*/
26     color: #046380;
27     margin-top:-230px; /* Marge du haut*/
28     margin-left:380px;
29 }
30
31 #logo{
32     height:185px;
33     margin-top:30px; /* Marge du haut*/
34     margin-left:-10px;
35 }
36
37 /*menu déroulant*/
38 nav{
39     margin:200px auto;
40     width:90%;
41     max-width:1200px /*longueur du menu */; /* Marge du haut*/
42     margin-top:120px;
43 }
44 nav ul {
45     list-style:none;
46 }
47 nav > ul {
48     display:table;
49     overflow:hidden;
50     width:100%;
51     background:#000;
52     position:relative;
53 }
54 nav > ul li {
55     display:table-cell;
56 }
57 nav > ul > li:hover > ul {
58     display:block;
59     height:100%;
60 }

```

```

60    }
61  nav > ul > li > ul {
62    display: block;
63    position: absolute;
64    background: #000;
65    left: 0;
66    right: 0;
67    overflow: hidden;
68    height: 0;
69    -webkit-transition: all .3s ease;
70    -moz-transition: all .3s ease;
71    -ms-transition: all .3s ease;
72    -o-transition: all .3s ease;
73    transition: all .3s ease;
74  }
75  }
76  nav > ul li a {
77    color: #fff;
78    display: block;
79    line-height: 20px;
80    padding: 20px;
81    position: relative;
82    text-align: center;
83    text-decoration: none;
84    -webkit-transition: all .3s ease;
85    -moz-transition: all .3s ease;
86    -ms-transition: all .3s ease;
87    -o-transition: all .3s ease;
88    transition: all .3s ease;
89  }
90  nav > ul > li > ul > li a:hover {
91    background: #5da5a2;
92  }
93  nav > ul > li > a span{
94    background: #174459;
95    display: block;
96    height: 100%;
96    height: 100%;
97    width: 100%;
98    left: 0;
99    position: absolute;
100   top: -55px;
101   -webkit-transition: all .3s ease;
102   -moz-transition: all .3s ease;
103   -ms-transition: all .3s ease;
104   -o-transition: all .3s ease;
105   transition: all .3s ease;
106  }
107  nav > ul > li > a span.icon{
108    display: block;
109    line-height: 60px
110  }
111  nav > ul > li > a:hover > span {
112    top: 0;
113  }
114  nav ul li a .primero {
115    background: #0e5061;

```

```

115    background:#0e5061;
116  }
117  nav ul li a .segundo {
118    background:#5da5a2;
119  }
120  nav ul li a .tercero{
121    background:#f25724;
122  }
123  nav ul li a .cuarto {
124    background:#174459;
125  }
126  nav ul li a .quinto {
127    background:#37a4d9;
128  }
129  /* Fin menu déroulent*/
130
131
132
133  /*rectangle sur dessin*/
134  #azimut /* bouton principaux*/{
135    position: absolute; /* Positionnement*/
136    line-height: 20px; /* Hauteur de ligne*/
137    width: 50px;
138    background-color: #EFECCA;
139    margin-left:325px;
140    margin-top:-90px;
141    text-align: center;
142    border: 1px solid black;
143    border-radius: 5px;
144    z-index: 2; /* Ordre d'affichage en cas de superposition*/
145  }
146  #elevation{
147    position: absolute;
148    line-height: 20px;
149    width: 50px;
150    background-color: #EFECCA;
151    margin-left:250px;
151    margin-left:250px;
152    margin-top:90px;
153    text-align: center;
154    border: 1px solid black;
155    border-radius: 5px;
156    z-index: 2;
157  }
158  #Valt{
159    position: absolute;
160    line-height: 20px;
161    width: 50px;
162    background-color: #EFECCA;
163    margin-left:500px;
164    margin-top:150px;
165    font-size:10px;
166    text-align: center;
167    color:green;
168    border: 1px solid black;
169    border-radius: 5px;
170    z-index: 2;
171  }
172  #Ialt_pv{
173    position: absolute;
174    line-height: 20px;
175    width: 50px;
175    margin-left: 375px;
175    margin-top: 150px;
175    font-size: 10px;
175    color: green;
175    border: 1px solid black;
175    border-radius: 5px;
175    z-index: 2;
175  }

```

```

175    width: 50px;
176    background-color: #EFECCA;
177    margin-left:500px;
178    margin-top:125px;
179    font-size:10px;
180    text-align: center;
181    color:red;
182    border: 1px solid black;
183    border-radius: 5px;
184    z-index: 2;
185 }
186 #Vpv           /*button lampe vert*/
187 {
188    position: absolute;
189    line-height: 20px;
190    width: 50px;
191    background-color: #EFECCA;
192    margin-left:1250px;
193    margin-top:220px;
194    font-size:10px;
195    text-align: center;
196    color:green;
197    border: 1px solid black;
198    border-radius: 5px;
199    z-index: 2;
200 }
201 #Vbatt {
202    position: absolute; /* button vert baterie*/
203    line-height: 20px;
204    width: 50px;
205    background-color: #EFECCA;
206    margin-left:600px;
207    margin-top:250px;
208    font-size:10px;
209    text-align: center;
210    color: green;
211    border: 1px solid black;
212    border-radius: 5px;
213    z-index: 2;
214 }
215 #Ibatt          /* button rouge baterie*/
216 {
217    position: absolute;
218    line-height: 20px;
219    width: 50px;
220    background-color: #EFECCA;
221    margin-left:600px;
222    margin-top:220px;
223    font-size:10px;
224    text-align: center;
225    color:red;
226    border: 1px solid black;
227    border-radius: 5px;
228    z-index: 2;
229 }
230 #Iaux           /* button rouge resistance*/
231 {
232    position: absolute;
233    line-height: 20px;
234    width: 50px;
235    background-color: #EFECCA;
236    margin-left:820px;
237    margin-top:220px;

```

```
236 margin-left:820px;  
237 margin-top:220px;  
238 font-size:10px;  
239 text-align: center;  
240 color:red;  
241 border: 1px solid black;  
242 border-radius: 5px;  
243 z-index: 2;  
244 }  
245 #Ichg /* button rouge onduleur abajo */  
246 {  
247 position: absolute;  
248 line-height: 20px;  
249 width: 50px;  
250 background-color: #EFECCA;  
251 margin-left:1050px;  
252 margin-top:220px;  
253 font-size:10px;  
254 text-align: center;  
255 color:red;  
256 border: 1px solid black;  
257 border-radius: 5px;  
258 z-index: 2;  
259 }  
260  
261 #Iconv /* button rouge onduleur arriba */  
262 {  
263 position: absolute;  
264 line-height: 20px;  
265 width: 50px;  
266 background-color: #EFECCA;  
267 margin-left:1050px;  
268 margin-top:40px;  
269 font-size:10px;  
270 text-align: center;  
271 border: 1px solid black;  
272 border-radius: 5px;  
273 z-index: 2;  
274 }  
275  
276  
277  
278 /*tableau données météo*/  
279 {  
280 #tableau {  
281 width:65%;  
282 position:absolute;  
283 margin-top:580px;  
284 margin-left:260px;  
285 z-index:5;  
286 }
```

```

287 th , td{
288     text-align:middle;
289     font-weight: bold;
290     font-size: 15px;
291 }
292 /*--FIN données station météo--*/
293
294 /*bouton*/
295
296
297
298 .montage, .montage1{
299     position: absolute;
300     width: 1213px;
301     margin-left: 220px; /*distancia imagen*/
302     margin-top: -150px;
303     z-index: 1;
304 }
305
306
307
308 #Info-iut, #Etudiant, #Tuteur, figure{
309     display: inline-block;
310     vertical-align: top;
311     padding-top: 0px; /* Marge intérieure en haut*/
312 }
313 #Info-iut{
314     font-style: italic;
315     margin-left: 20px;
316     width: 350px;
317 }
318 #Info-iut p{
319     font-weight: bold;
320     font-size: 20px;
321 }
322 #Info-iut li{
323     list-style-type: none; /* Change le style des puces d'une liste*/
324 }
325 #Etudiant{
326     font-style: italic;
327     width: 300px;
328 }
329 #Etudiant p{
330     font-weight: bold;
331     font-size: 20px;
332 }
333 #Etudiant li{
334     list-style-type: disc;
335 }

```

```

335     }
336     #Tuteur{
337         width: 280px;
338     }
339     #Tuteur p{
340         font-weight: bold;
341         font-size: 20px;
342     }
343
344     #position /*position du soleil */
345     {
346         position:relative;
347         margin-left:500px;
348         margin-top:950px;
349         z-index:2;
350         width:1200px;
351         height:550px;
352     }
353
354     /*Bande Meteo 3*/
355     #TT_RyJEE11kkIEQE1KAKFzjjDzDjWnAMdEkuzDzj66GD25oA6nIg
356     {
357         position:absolute;
358         margin-left:500px;
359         margin-top:200px;
360         z-index:2;
361         width:1200px;
362         height:200px;
363     }
364
365     /*fin pag Meteo*/
366
367     /*bouton*/
368
369     #lampelCJL{ /*Déplacement de la section (tout)*/
370         Margin-top:-1214px;
371         margin-left:1192px;
372         z-index:9;
373         position: absolute;
374     }
375
376     #11 /* Dans la section 1(que la lampe)*/
377     {
378         position: absolute;
379         margin-left:15px;
380         margin-top:35px;

```

```
380     margin-top:35px;  
381     width: 55px;  
382     height:75px;  
383     z-index:6;  
384 }  
385  
386 #lampe2CJL /*toute la section 2*/  
387 {  
388     margin-left:1304.5px;  
389     margin-top:-1215px;  
390     z-index:6;  
391     position: absolute;  
392 }  
393  
394 #12  
395 {  
396     position:absolute;  
397     margin-left:10px;  
398     margin-top:10px;  
399     width: 55px;  
400     height:75px;  
401     z-index:6;  
402 }  
403  
404  
405 footer {  
406     position: absolute;  
407     border: 2px solid black;  
408     border-radius: 10px;  
409     margin-left: 200px;  
410     margin-top:650px;  
411     margin-bottom:10px;  
412     width: 1250px;  
413     height: 180px;  
414 }  
415  
416 footer figure{  
417     margin-top: 60px;  
418 }
```

8.4 Annexe 4 : calcul azimut, élévation et pilotage des moteur (Test_pilotage_moteur.py)

```

1  #!/usr/bin/python3.2
2  # -*- coding: utf8 -*-
3
4  import math      # Importer la bibliothèque Math
5  import time       # Importer la bibliothèque Temps
6  import MySQLdb    # Importer la bibliothèque base de données
7  #from Capteur import * # Importer les données du programme Capteur.py (ordre Capteur.py s'exécute avant Test_pilotage_moteur)
8  import serial
9
10 #-----
11 # Déclaration de la fonction de calcul de la hauteur solaire et de l'azimut:
12 #           CalPosSol()
13 #-----
14
15 def CalPosSol(heure,min,jour,mois,annee):  # Définition de la fonction
16     # debut du code de la fonction
17
18     E= math.asin(0.687 * math.sin(Dec(Jour)) + 0.725 * math.cos(Dec(Jour)) * math.cos(Ah_r(Tsv)))  # angle de l'élévation en degrés
19
20     if Ah_r(Tsv) >= (-1 * math.pi) and Ah_r(Tsv) <= (-1 * math.pi)/3:
21         A = math.asin((math.cos(Dec(Jour)) * math.sin(Ah_r(Tsv)))/ math.cos(E)) * -1      # angle de l'azimut en degrés
22     elif Ah_r(Tsv) >= (-1 * math.pi)/3 and Ah_r(Tsv) <= (math.pi)/3:
23         A = math.asin((math.cos(Dec(Jour)) * math.sin(Ah_r(Tsv)))/ math.cos(E)) + math.pi
24     else:
25         A = math.asin((math.cos(Dec(Jour)) * math.sin(Ah_r(Tsv)))/ math.cos(E)) * -1 + ( 2* math.pi)
26     A = math.degrees(A)
27     E = math.degrees(E)
28
29     # Afficher date et heure
30     print "Horodatage du calcul: ",time.strftime('%H'),":",time.strftime('%M'),":",time.strftime('%d'),"/",time.strftime('%m'),"/",time.strftime('%Y')
31     return (A,E)
32     # fin du code de la fonction
33
34 #-----
35 # Déclaration de la fonction de calcul du jour d l'année:
36 #
37 #-----
38
39 def Jour():
40     a = time.strftime('%Y')
41     a= int(a)
42     m = time.strftime('%m')
43     m = int(m)
44     j = time.strftime('%d')
45     j = int(j)
46     if((a%4 == 0 and a%100 != 0) or a%400 == 0):   # Calcul si année bissextile
47         return(0,31,60,91,121,152,182,213,244,274,305,335,366)[m-1]+j
48
49 #-----
50 # Déclaration de la fonction de la déclinaison:
51 #
52 #-----
53
54
55 def Dec(Jour):
56     Dec = math.radians(23.45 * math.sin((6.28 * (Jour() + 284))/ 365))
57     return Dec
58
59
60 #-----
61 # Déclaration de la fonction du temps solaire vrai:
62 #
63 #-----
64
65 def Tsv():
66     heure = time.strftime('%H')
67     heure = int(heure)
68     min = time.strftime('%M')
69     min = float(min)
70     if Jour() > 90 and Jour() < 300:  # Si heure d'été
71         Tsv = heure+1 + min/60
72     else:
73         Tsv = heure + min/60
74     return Tsv
75
76
77 #-----
78 # Déclaration de la fonction de l'amplitude horaire:
79 #
80 #-----
81
82 def Ah_r(Tsv):
83     Ah_r = math.radians((Tsv()-12)*(360/24))
84     return Ah_r
85
86
87 # Test du programme CalPosSol:
88 #
89
90 resultats = CalPosSol(time.strftime('%H'),time.strftime('%M'),time.strftime('%d'),time.strftime('%m'),time.strftime('%Y'))
91 print "azimut:",resultats[0]      # affichage de l'azimut
92 print "élévation:",resultats[1]   # affichage de l'élévation solaire
93
94
95 #-----
96 # Déclaration de la conversion des degrés en point pour l'azimut
97 #-----
98
99 def Conv_Deg_PtA(Pos_Deg1):

```

```

99     def Conv_Deg_PtA(Pos_Deg1):
100         Conv_Deg_PtA = (-584.69*(Pos_Deg1)+152019)
101         return Conv_Deg_PtA
102
103     #-----
104     # Déclaration de la conversion des degrés en point pour l'élévation
105     #-----
106
107     def Conv_Deg_PtE(Pos_Deg2):
108         Conv_Deg_PtE = (989.61*(Pos_Deg2)-16494)
109         return Conv_Deg_PtE
110
111     #-----
112     # Affichage de la conversion des degrés en point
113     #-----
114
115     azimut_pt = Conv_Deg_PtA(resultats[0])           # conversion de l'azimut des degrés en point
116     print "azimut en point:",azimut_pt ,resultats[0]  # affichage de l'azimut en point
117     elevation_pt = Conv_Deg_PtE(resultats[1])         # conversion de l'élévation des degrés en point
118     print "elevation en pt:",elevation_pt ,resultats[1] # affichage de l'élévation en point
119
120     #int(azimut_pt)
121     #int(round(azimut_pt,0))                         # Suppression des décimales et de la virgule, arrondi du résultat de l'azimut
122     #int(round(elevation_pt,0))                       # Suppression des décimales et de la virgule, arrondi du résultat de l'élévation
123
124     #-----
125     # Calcul permettant d'affiner le résultat en point pour l'azimut
126     #-----
127
128     if int(azimut_pt) >= 96000:
129         val_azimut = 96000
130         print "val_azimut = ",val_azimut
131     elif int(azimut_pt) <= 0:
132         val_azimut = 0
133         print "val_azimut = ",val_azimut
134     else:
135         val_azimut = int(azimut_pt)
136         print "val_azimut = ",val_azimut
137
138     #-----
139     # Calcul permettant d'affiner le résultat en point pour l'élévation
140     #-----
141
142     if int(elevation_pt) >= 96000:
143         val_elevation = 96000
144         print "val_elevation = ",val_elevation
145     elif int(elevation_pt) <= 0:
146         val_elevation = 0
147         print "val_elevation = ",val_elevation
148     else:
149         val_elevation = int(elevation_pt)
150         print "val_elevation = ",val_elevation
151
152
153     #-----
154     # Pilotage des moteurs
155     #-----
156
157
158     ser=serial.Serial('/dev/ttyACM0',38400,timeout=1)
159     ser.write("$."+str(val_elevation)+"\n")
160     print val_azimut
161     ser=serial.Serial('/dev/ttyACM1',38400,timeout=1)
162     ser.write("$."+str(val_azimut)+"\n")
163     print val_elevation
164
165     ser.close()
166
167     #ser=serial.Serial('/dev/ttyACM1',38400,timeout=1)
168     #ser.write("$."+str(val_azimut)+"\n")
169
170     #print "azimut en point:",(-989.61 * resultats[0] + 250700)
171     #conversion = (-989.61 * resultats[0] + 250700)
172
173     #-----
174     # Pilotage des moteurs
175     #-----
176
177
178     #import serial
179
180     #ser=serial.Serial('/dev/ttyACM0',38400,timeout=1)
181     #ser.write('$.'+str(val_azimut+'\n')
182     #ser=serial.Serial('/dev/ttyACM1',38400,timeout=1)
183     #ser.write('$.'+str(int(round(elevation_pt,0))+'\n')
184     #ser.close()
185

```

8.5 Annexe 5 : Programme de récupération des données de la station météo (DonneeMeteoFini.py)

```

1  #!/usr/bin/python3.2
2  # -*- coding: utf-8 -*-
3
4  import time, threading # importation des bibliothèques de temps
5  import demjson # importation de la bibliothèque de serialisation des données Json
6  from pymodbus.client.sync import ModbusTcpClient # importer les fonctions de dialogues
7  DonneeMeteo={} # création du dictionnaire pour les données capteur
8
9  #
10 # Lecture des données des capteurs
11 #
12
13 def mesures_ls():
14     from pymodbus.client.sync import ModbusTcpClient # importer les fonctions de dialogues
15     client = ModbusTcpClient('10.3.208.225',9000) # dialogue avec la station météo
16
17     # Affichage de l'humidité
18     result = client.read_input_registers(74,1) # Récupérer les données à partir de la sortie n°74 et combien de ligne qui suit
19     data = result.registers[0]/10.0 # Diviser result par 10
20     DonneeMeteo["AbsH"] = data # Ecrire la valeur de l'humidité dans le dictionnaire DonneeMeteo
21
22     # Affichage du rayonnement solaire
23     result = client.read_input_registers(27,1) # Récupérer les données à partir de la sortie n°27 et combien de ligne qui suit
24     data = result.registers[0]/10.0 # Diviser result par 10
25     DonneeMeteo["GIRad"] = data # Ecrire la valeur de rayonnement solaire dans le dictionnaire DonneeMeteo
26
27     # Affichage de la vitesse actuelle du vent en km/h
28     result = client.read_input_registers(83,1) # Récupérer les données à partir de la sortie n°83 et combien de ligne qui suit
29     data = result.registers[0]/10.0 # Diviser result par 10
30     DonneeMeteo["Windact"] = data # Ecrire la valeur dans le dictionnaire DonneeMeteo
31
32     # Affichage de la vitesse maximum du vent en km/h
33     result = client.read_input_registers(85,1) # Récupérer les données à partir de la sortie n°85 et combien de ligne qui suit
34     data = result.registers[0]/10.0 # Diviser result par 10
35     DonneeMeteo["Windmax"] = data # Ecrire la valeur dans le dictionnaire DonneeMeteo
36
37     # Affichage de la pression atmosphérique
38     result = client.read_input_registers(17,1) # Récupérer les données à partir de la sortie n°17 et combien de ligne qui suit
39     data = result.registers[0]/10.0 # Diviser result par 10
40     DonneeMeteo["Pattm"] = data # Ecrire la valeur dans le dictionnaire DonneeMeteo
41
42     # Affichage de la température actuelle
43     result = client.read_input_registers(31,1) # Récupérer les données à partir de la sortie n°31 et combien de ligne qui suit
44     data = result.registers[0]/10.0 # Diviser result par 10
45     DonneeMeteo["T"] = data # Ecrire la valeur dans le dictionnaire DonneeMeteo
46
47     # Affichage de la température min
48     # result = client.read_input_registers(32,1)
49     # data = result.registers[0]/10.0
50     # DonneeMeteo["Tmin"] = data
51
52     Affichage de la température max
53     result = client.read_input_registers(33,1) # Récupérer les données à partir de la sortie n°33 et combien de ligne qui suit
54     data = result.registers[0]/10.0 # Diviser result par 10
55     DonneeMeteo["Tmax"] = data # Ecrire la valeur dans le dictionnaire DonneeMeteo
56
57     # Affichage de la température moy
58     result = client.read_input_registers(34,1) # Récupérer les données à partir de la sortie n°34 et combien de ligne qui suit
59     data = result.registers[0]/10.0 # Diviser result par 10
60     DonneeMeteo["Tmoy"] = data # Ecrire la valeur dans le dictionnaire DonneeMeteo
61
62     # Affichage de la précipitation
63     result = client.read_input_registers(50,1) # Récupérer les données à partir de la sortie n°50 et combien de ligne qui suit
64     data = result.registers[0]/10.0 # Diviser result par 10
65     DonneeMeteo["Precipitation"] = data # Ecrire la valeur dans le dictionnaire DonneeMeteo
66
67     # Affichage de la boussole
68     result = client.read_input_registers(24,1) # Récupérer les données à partir de la sortie n°24 et combien de ligne qui suit
69     data = result.registers[0]/10.0 # Diviser result par 10
70     DonneeMeteo["Compass"] = data # Ecrire la valeur dans le dictionnaire DonneeMeteo
71
72     # Affichage de la direction du vent
73     result = client.read_input_registers(97,1) # Récupérer les données à partir de la sortie n°97 et nbre de ligne qui suit
74     data = result.registers[0]/10.0 # Diviser result par 10
75     DonneeMeteo["Wind_direction"] = data # Ecrire la valeur dans le dictionnaire DonneeMeteo
76
77     print DonneeMeteo # lire le dictionnaire DonneeMeteo
78     client.close() # Fermer le dialogue avec la station météo
79
80     json = demjson.encode(DonneeMeteo) # serialisation au format Json du dictionnaire DonneeMeteo
81     print json # Ecrire de la serialisation au format Json
82
83     # script lecture.py
84     NomFichier = '/var/www/montmpfs/dataSW.json' # chemin et nom du fichier d'échange système>>web
85     # creation et ouverture du fichier data.json en mode write 'w' (écriture)
86     # si le fichier test.txt existe déjà, il est écrasé
87     Fichier = open(NomFichier,'w') # instantiation de l'objet Fichier de la classe file
88     Fichier.write(json) # écriture dans le fichier avec la méthode write()
89     Fichier.close() # fermeture du fichier avec la méthode close()
90     threading.Timer(3, mesures_ls).start() # répéter la boucle mesures_ls toutes les 3s
91
92     mesures_ls() #démarrer le programme de mesures
93

```

8.6 Annexe 6 : Schéma de fonctionnement global

