

POSITIONNEMENT DU TRACKER PHOTOVOLTAÏQUE

DUCOUT David, HIRIBARNE Andoni, BIRABENT Cédric

*Prise en main du
panneau
photovoltaïque,
du programme
python et des
sujets
antécédents.*

Table des matières

Remerciement.....	2
Résumé	3
Abstract.....	4
Introduction	5
I/ Découverte des projets précédents.....	6
1/ Évolution.....	6
2/ Objectifs	6
II/ Amélioration de l'équation solaire.....	7
1/ Analyse et solutions	7
2/ Résultat.....	11
III/ Programmation	13
1/ Découverte des composant du tracker solaire	13
2/ Notre programme	15
3/ Conversion et Cron tab	18
IV/ Mécanique	21
1/ Conception du boîtier d'étanchéité	21
2/ Maintenance du moteur	23
V/ Conclusion	25
VI/ Liste des figures.....	26
VII/ Annexes	27
ANNEXE 1 : Tableau Excel	27
ANNEXE 2 : Programme Python.....	28
ANNEXE 3 : Bons de commande	29
ANNEXE 4 : SolidWorks des pièces en plastique	30

Remerciement

Nous exprimons d'abord toute notre gratitude à toutes les personnes qui ont contribué au succès de ce projet et qui nous ont aidés lors de la rédaction de ce rapport.

Nous adressons nos remerciements à l'IUT de Bayonne et du Pays Basque, qui nous ont permis de réaliser ce projet dans de bonnes conditions, et fournis le matériel nécessaire.

Enfin, nous tenons à remercier Philippe LARRAMENDY, notre tuteur de projet, pour l'aide et les conseils de réalisation apportée pour réussir ce projet à temps, et commander les pièces manquantes.

Résumé

Au cours de notre cursus professionnel, nous avons été en charge d'un projet d'octobre à avril. C'est un projet de groupe indépendant. Cela nous permet d'apprendre à travailler de manière autonome et de mobiliser les connaissances acquises pendant la course.

Notre projet se concentre spécifiquement sur un tracker solaire. C'est un système qui absorbe la lumière du soleil comme source d'énergie pour produire de l'électricité. Un module photovoltaïque (PV) est un ensemble de cellules solaires photovoltaïques connectées entre elles. Les modules photovoltaïques constituent la matrice photovoltaïque du système, qui génère et fournit de l'électricité solaire pour des applications commerciales et résidentielles.

Nous avons dû améliorer un programme existant pour que le panneau solaire fonctionne à pleine capacité, et il était nécessaire de refaire la boîte comprenant les moteurs du panneau. Dans le programme, il était nécessaire de mettre en place un mode de sécurité lorsque le vent dépasse 100 km/h, et un mode automatique et manuel afin que les élèves puissent prendre le contrôle du panneau.

Notre groupe a bien fonctionné, ce qui a été très utile tant pour la recherche de l'amélioration du programme que pour la mise en place de la boîte finale.

Abstract

During our professional degree, we are in charge of a project from October to April. This is an independent group project. This allows us to learn to work autonomously and to mobilize the knowledge acquired during the course.

Our project focuses specifically on a solar tracker. It is a system that absorbs sunlight as a source of energy to generate electricity. A photovoltaic (PV) module is a packaged, connect assembly photovoltaic solar cells. Photovoltaic modules constitute the photovoltaic array of a photovoltaic system that generates and supplies solar electricity in commercial and residential applications.

We had to improve an existing program for the solar panel to operate at full capacity, and it was necessary to redo the box including the two panel motor. In the program, it was necessary to set up a safety mode when the wind exceeds 100Km / h, and an auto and manual mode so that students can take control of the panel.

Our group worked well, which was very useful both for the research of the improvement of the program and the implementation of the final box.

Introduction

Il est pour projet l'installation d'un « tracker » solaire photovoltaïque sur le toit de l'IUT. Ce tracker se base sur deux moteurs, un couvrant l'axe verticale et l'autre l'axe horizontale. Ces deux moteurs seront programmés pour suivre la trajectoire du soleil quelques soit la saison et quelques soit l'heure par l'intermédiaire d'un raspberry (micro-ordinateur). Le fait de suivre la trajectoire du soleil permet un rendement maximal au panneau.

Ce projet a déjà été entamé par des groupes de projet les années précédentes. Cependant, plusieurs tâches nous restent à faire.

Premièrement, le programme utilisé par les groupes précédents était perfectible donc une première recherche pour l'amélioration sera nécessaire surtout au lever et au coucher du soleil.

Deuxièmement, une fois notre programme élaboré il faut le transcrire en langage python pour pouvoir l'insérer dans le raspberry.

Finalement, une dernière partie mécanique devra être faite avec l'élaboration d'un boîtier étanche pour permettre aux moteurs de rester à l'abri des intempéries. De plus, un moteur étant endommager il est nécessaire de le réparer.

Notre équipe est composée de trois membres. David Ducout est chef de projet, Andoni Hiribarne est responsable de communication et Cédric Birabent s'occupe de la partie technique.

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

I/ Découverte des projets précédents

1/ Évolution

Le projet porte sur l'implantation d'un tracker solaire autonome sur le toit d'un futur bâtiment du site de Montaury.

Après analyse des anciens rapports nous avons constaté qu'ils avaient réalisé une bonne partie du projet :

- toute la partie électrique est faite avec l'armoire électrique
- le branchement des servomoteurs.
- la mise en place du raspberry.
- une programmation sur python a été réalisé.

2/ Objectifs

Nos objectifs pour l'amélioration du tracker sont :

- l'amélioration de l'équation solaire réalisée par le groupe d'étudiants de la promotion de 2015.
- finaliser la programmation du raspberry.
- la maintenance du servomoteur qui permet l'orientation azimuth du panneau.
- la conception d'un boîtier d'étanchéité des deux servomoteurs

Pour réaliser dans de bonnes conditions nos objectifs, nous nous sommes organisés en réalisant un planning. Mettre en place un planning prévisionnel et définir les tâches respectives à chacun nous a permis d'obtenir une méthode de travail.

II/ Amélioration de l'équation solaire

1/ Analyse et solutions

Tout d'abord, pour pouvoir diriger le tracker photovoltaïque nous devons valider les équations qui permettent de connaître la position du soleil.

Avant de les vérifier, nous devons d'abord effectuer des recherches pour comprendre à quoi correspondent ces équations. Puis nous avons cherché des sites internet qui permettent de calculer la position du soleil en fonction des coordonnées géographiques afin de pouvoir les comparer à nos résultats.

Nous allons nous servir des sites « Sunearthtools » et « Solartopo ».

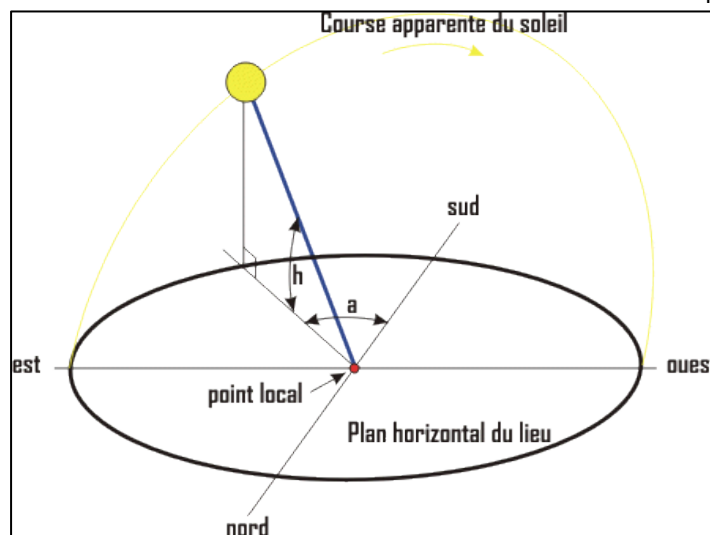


Figure 1 : Course apparente du soleil

Hauteur ou Élévation, h est la hauteur angulaire du Soleil sur l'horizon (plan vertical).

Azimut a , l'angle entre le sud et la projection au sol de la droite issue de la direction terre-soleil (plan horizontal).

Pour récupérer le plus d'énergie du soleil, il faut que les rayons soient perpendiculaires au panneau, pour cela nous avons besoin de connaître l'azimut et l'élévation afin d'adapter la position du panneau.

En effet, un tracker solaire sur deux axes permet d'améliorer de 35% le rendement de production par rapport à un panneau fixe.

L'axe horizontal sert à régler l'azimut, ce qui nous permet de suivre l'avancée du soleil dans la journée, tandis que l'axe vertical, pour l'élévation, permet d'être plus précis et de pouvoir régler l'orientation en fonction des saisons.

Pour obtenir la position exacte du soleil nous nous basons sur l'azimut et l'élévation. Ces deux caractéristiques nous permettent d'obtenir la position avec exactitude. Pour avoir ces deux indications nous nous appuyons sur de nombreux paramètres.

Pour calculer ces deux angles, nous devons effectuer des calculs intermédiaires et pour pouvoir les réaliser il faut connaître les coordonnées GPS du lieu, la date et l'heure.

Pour montrer les interactions entre les différentes équations vous trouverez un schéma ci-dessous.

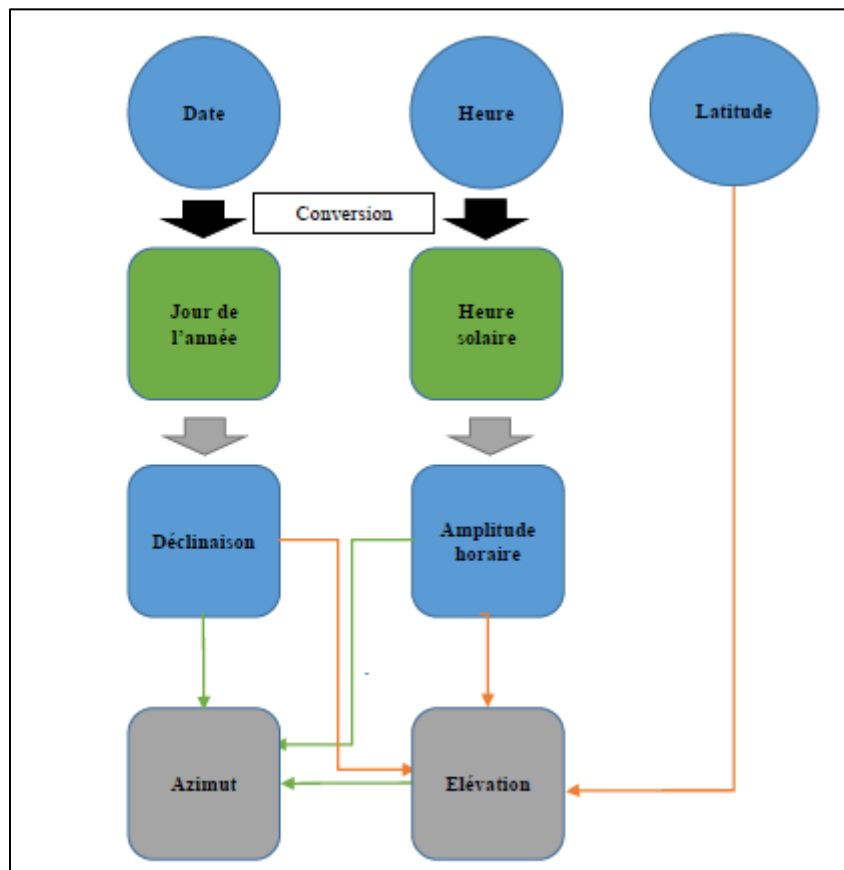


Figure 2 : Synoptique du fonctionnement des équations

En comparant les courbes représentant la trajectoire du soleil obtenu les années précédentes par rapport aux valeurs réelles, on remarque un manque de précision au coucher et au lever du soleil.

De ce fait, il nous a été demandé de revoir les équations qui ont permis d'obtenir ces courbes soit l'azimut et l'élévation, les deux éléments permettant de caractériser la trajectoire du soleil.

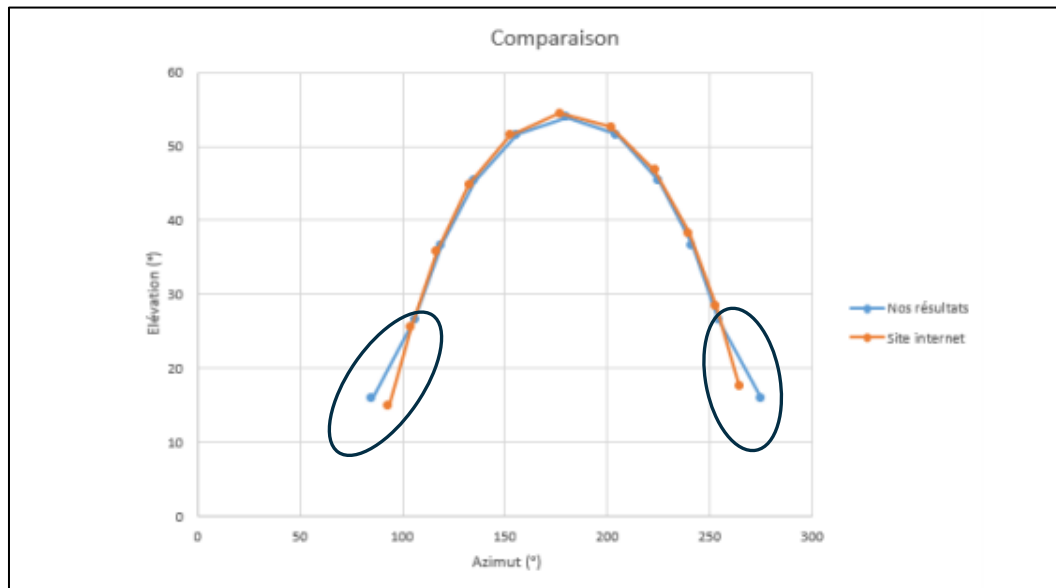


Figure 3 : Courbe obtenue les années précédentes

Après des recherches assidues, nous avons réussi à obtenir des équations plus précises car elles prenaient en compte de nombreux facteurs qui pouvaient influencer sur la position du soleil. Ci-dessous nous allons expliquer les équations qui nous permettent d'obtenir l'azimut et l'élévation ainsi que celles qui en découlent.

- **Explication de l'équation de l'élévation**

Les résultats que nous utilisons de l'équation finale dépendent de deux facteurs. L'angle solaire zénithal correspondant à l'angle entre le zénith et le centre du disque solaire mais également de la réfraction atmosphérique qui est une correction qu'on ajoute par rapport aux conditions atmosphériques.

Élévation final= $90 - \text{angle solaire du zénith (en deg)} + \text{réfraction atmosphérique}$

Ensuite, l'angle solaire correspondant à une équation qui dépend de la latitude soit un facteur fixe qu'on prédéfinit au départ sur l'Excel, dans notre cas nous prenons la latitude de l'IUT GIM à Montauray. Cette équation prend également en compte la déclinaison du soleil et l'angle horaire.

Angle solaire zénithale= $\text{Acos} [\sin(\text{latitude}) \cdot \sin (\text{déclinaison du soleil}) + \cos(\text{latitude}) \cdot \cos (\text{déclinaison du soleil}) \cdot \cos (\text{angle horaire})] \cdot 360 / 2\pi$

Tous les angles dans l'équation sont en radians et nous multiplions le tout par 360 et divisons par 2π pour obtenir l'angle solaire zénithale en degrés.

La déclinaison solaire est l'angle formé par la droite reliant la terre au soleil et le plan équatorial.

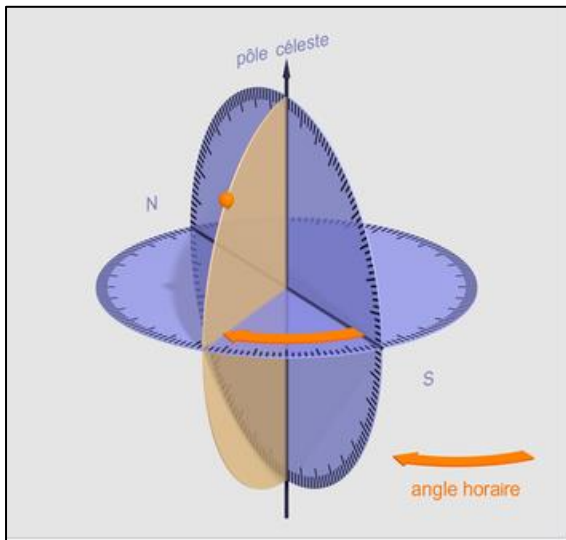


Figure 4 : Angle horaire

Angle horaire

L'observateur est au centre. Le disque horizontal correspond à l'Equateur. Le disque vertical est formé par le cercle qui relie le pôle Nord céleste, le zénith (local) et le pôle Sud céleste. La petite sphère orange représente l'étoile. L'angle horaire est donc l'angle entre le méridien et l'ascension droite de l'étoile.

La déclinaison solaire et l'angle horaire sont également eux-mêmes établis à partir d'équation utilisant plusieurs paramètres eux-mêmes basés sur des équations.

Déclinaison solaire = $\sin^{-1}(\sin(\text{mean obliq ecliptic} + 0.0256 \times \cos(125.04 - 1934.136 \times \text{Julian Century})) \times \sin(\text{sun true long} - 0.00569 - 0.00478 \times \sin(125.04 - 1934.136 \times \text{Julian Century})))$

La déclinaison solaire est en degré. On constate que de nouveau paramètres interviennent.

Ainsi, pour parvenir à l'équation finale de l'élévation nous faisons appel à une quinzaine d'équations imbriquées les unes dans les autres.

- **Explication de l'équation de l'azimut**

L'équation finale qui permet de calculer l'azimut est une équation très complexe. Celle-ci diffère suivant l'angle horaire c'est-à-dire si l'angle horaire est supérieur à 0, une certaine équation sera applicable pour calculer l'azimut sinon on doit utiliser une autre équation. En plus de l'angle horaire, la latitude, la déclinaison du soleil et l'angle solaire zénithale interviennent dans l'équation.

Equation de l'azimut :

Si angle horaire > 0 : $\text{Acos}[(\sin(\text{latitude}) * \cos(\text{angle solaire zénithale}) - \sin(\text{déclinaison du soleil})) / (\cos(\text{latitude}) * \sin(\text{angle solaire zénithale})) + 188] * 2\pi / 360$;

Sinon, $540 - \text{Acos}(\sin(\text{latitude}) * \cos(\text{angle solaire zénithale}) - \sin(\text{déclinaison})) / (\cos(\text{latitude}) * \sin(\text{angle solaire zénithale})) * 2\pi / 360$

Tous les angles qui interviennent dans les équations sont en radian et nous avons multiplié 2π puis divisé par 360 pour obtenir des résultats en degré

Ainsi, grâce à ces équations que nous avons détaillées précédemment nous avons obtenu des résultats précis au dixième de degré près.

2/ Résultat

Pour avoir des résultats les plus précis possible nous étions obligés de retravailler l'équation du soleil.

Pour cela nous avons tout d'abord essayé d'être plus rigoureux sur les résultats des anciens étudiants. En vain, nous retombions sur de même résultats.

Après de multitudes recherches, nous avons trouvé un excel. Les calculs dans l'excel « calculateurs NOAA (National Oceanic and Atmospheric Administration) Sunrise / Sunset and Solar Position » sont basés sur les équations d'algorithmes astronomiques, de Jean Meeus. Les résultats du lever et du coucher du soleil sont théoriquement exacts à une minute près pour les emplacements situés entre +/- 72 ° de latitude et 10 minutes au-delà de ces latitudes. Cependant, en raison des variations de la composition de l'atmosphère, de la température, de la pression et des conditions, les valeurs observées peuvent différer des calculs.

Les feuilles de calcul suivantes peuvent être utilisées pour calculer les données solaires d'un jour ou d'une année sur un site donné. Ils sont disponibles en format Microsoft Excel

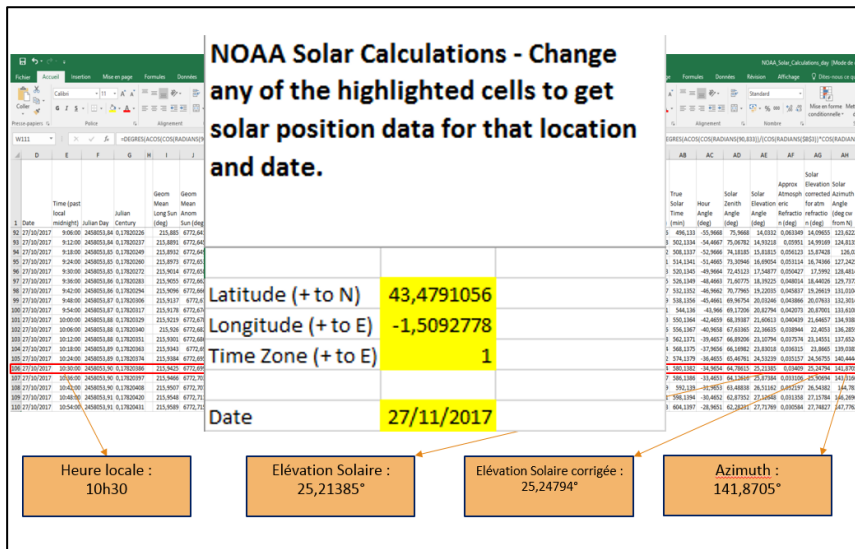


Figure 5 : Valeurs sur Excel

Date	27/10/2017 1 GMT1	
coordonnées	43.4800424, -1.5093202	
emplacement	1 Allée du Parc Montaury, 64600 Anglet, France	
heure	Elevation	Azimuth
9:00:00	13.11°	122.44°
9:05:00	13.87°	123.42°
9:10:00	14.62°	124.4°
9:15:00	15.36°	125.4°
9:20:00	16.1°	126.42°
9:25:00	16.82°	127.44°
9:30:00	17.54°	128.47°
9:35:00	18.24°	129.52°
9:40:00	18.93°	130.57°
9:45:00	19.62°	131.64°
9:50:00	20.29°	132.72°
9:55:00	20.95°	133.82°
10:00:00	21.59°	134.93°
10:05:00	22.23°	136.05°
10:10:00	22.85°	137.18°
10:15:00	23.46°	138.33°
10:20:00	24.05°	139.49°
10:25:00	24.64°	140.67°
10:30:00	25.2°	141.86°
10:35:00	25.75°	143.06°
10:40:00	26.29°	144.28°
10:45:00	26.81°	145.51°
10:50:00	27.32°	146.76°
10:55:00	27.8°	148.02°
11:00:00	28.27°	149.29°
11:05:00	28.73°	150.58°
11:10:00	29.16°	151.88°
11:15:00	29.58°	153.19°
11:20:00	29.98°	154.52°
11:25:00	30.36°	155.86°
11:30:00	30.72°	157.21°

Figure 6 : Résultats solaires exactes

Sur l'Excel il suffit de rentrer la position du site Montaury (figure2) ainsi que la date pour qu'il nous affiche l'azimute ainsi que l'élévation solaire toutes les 10 minutes. D'après la figure 5 et 6 on constate que nos résultats sont précis et exacts au millième de degrés près.

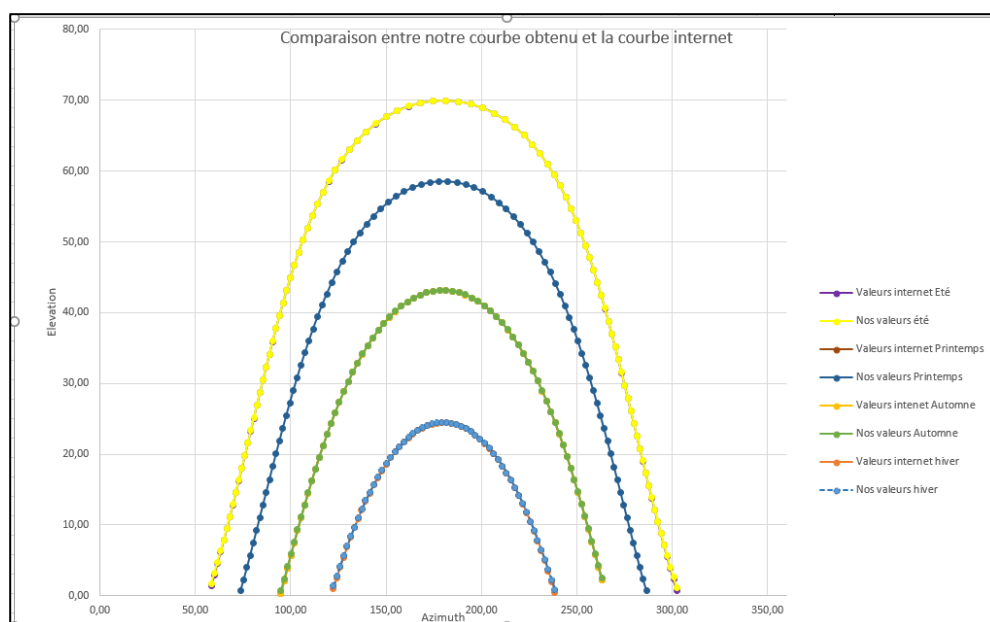


Figure 7 : Comparaison des résultats et de ceux d'internet en fonction des saisons

D'après la figure 7 on constate qu'il n'y a plus de soucis de précision au levé et couché du soleil. Notre premier objectif est donc rempli.

Nous allons donc nous servir de ça pour la suite de notre projet

III/ Programmation

1/ Découverte des composant du tracker solaire

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 motorisés permettant de suivre la course du soleil et ainsi optimiser son rendement.

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 horizontaux et verticaux permettant d'exploiter au maximum le rayonnement du soleil. Les moteurs solaires sont alimentés par une source externe, 12VDC, et peuvent résister à un vent de 130 km/h.

Raspberry Pi :

Le Raspberry Pi est un nano ordinateur monocarte 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

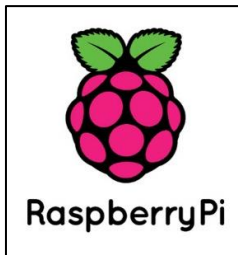


Figure 8 : Logo raspberry

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.

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 milliards de valeurs (soit 2³²).

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 faibles capacités, 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 (HMDI et prise RCA),
- Une micro SD card,

Il a une consommation globale de 3 Watts.

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 microordinateur 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,
- Rassembler des données vers un fichier (Json ou .txt),

Figure 9 : Raspberry Pi B+



2/ Notre programme

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. 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.

Nous avons donc utilisé le programme Excel pour réaliser le programme python. Il a fallu apprendre ce langage qui est très facile d'utilisation. De nombreux tutoriels sont présents sur internet pour l'apprentissage de celui-ci.

Nous avons créé un diagramme (voir ci-dessous) expliquant les différentes étapes de notre programme nommé « Pyhonpoint.py ». Lorsque le programme s'exécute, on calcule tout d'abord la position réelle de l'azimut et de l'élévation du soleil en degrés. Une fois les angles calculés, on va questionner la station météo



Figure 10 : Logo Python

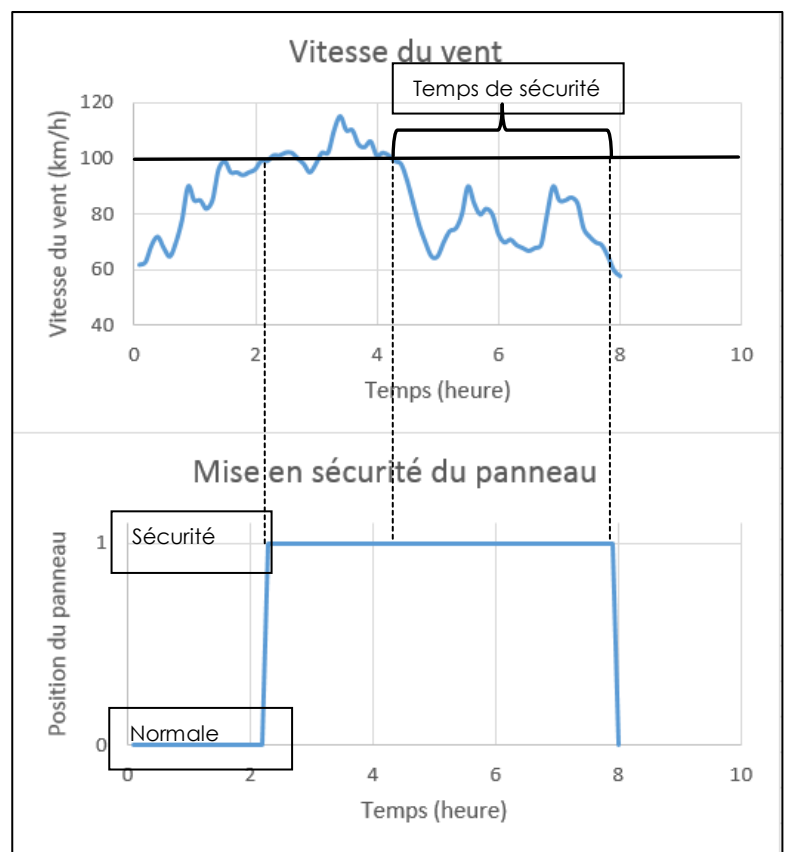
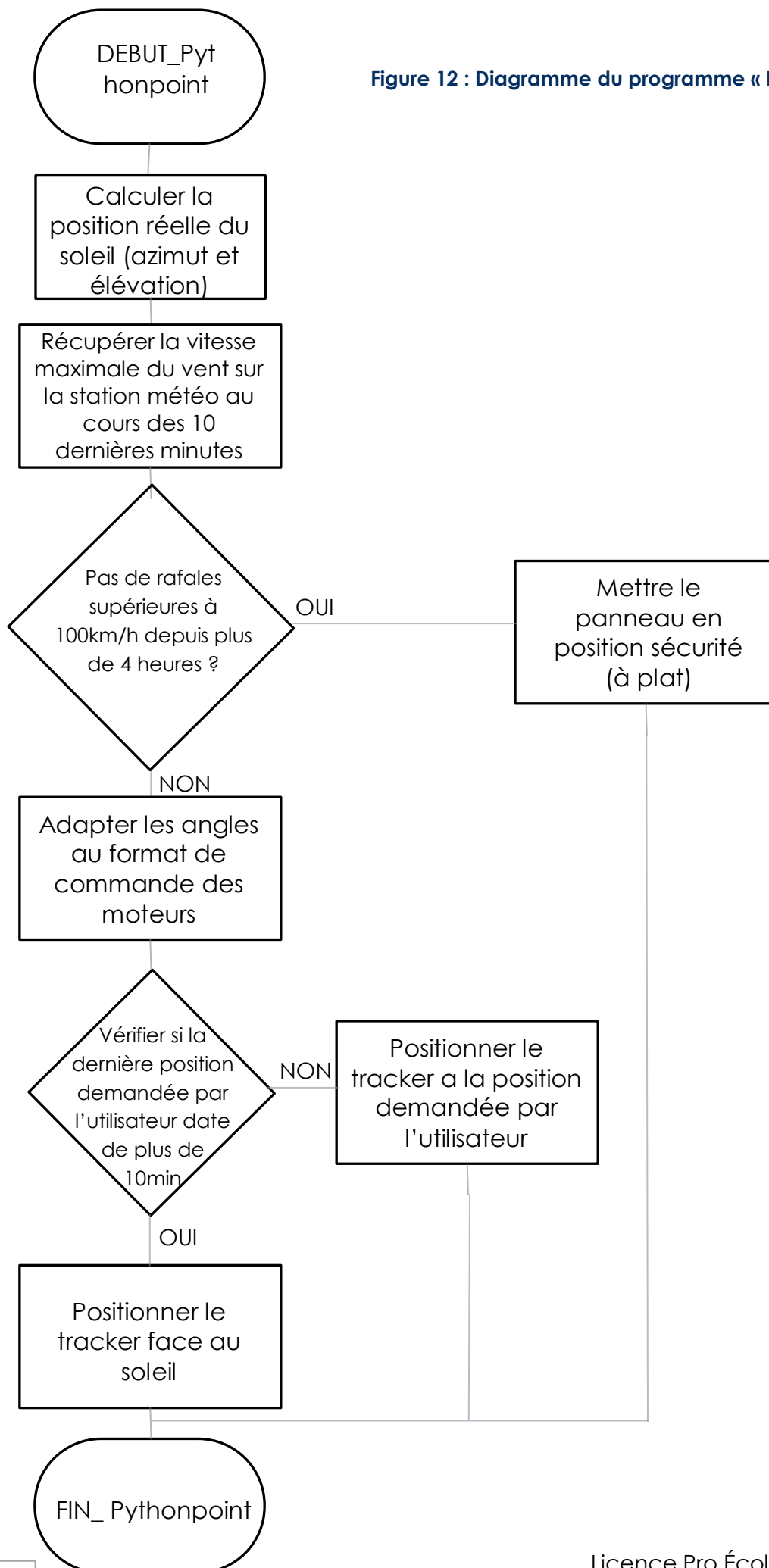


Figure 11 : Graphiques montrant la fonction "Sécurité" du panneau solaire

de l'IUT pour récupérer les données de la vitesse du vent sur les 10 dernières minutes. Ensuite, si le vent a une vitesse supérieure à 100 km/h, on met le panneau en position sécurité. La position sécurité du panneau consiste à le mettre à l'horizontale (90°) pour éviter une prise au vent du panneau. Une fois la position de sécurité activée, on attend de vérifier qu'il n'y a pas eu de nouvelle rafales dans les 4 dernières heures. Enfin, on relance le programme pour de nouveau calculer la position actuelle du soleil.

Sinon, si la vitesse du vent maximale est inférieure à 100 km/h, on peut continuer la suite du programme. Une fois la vitesse du vent vérifiée, on adapte les angles de la position du soleil au format de la commande des moteurs. On refait un nouveau test pour savoir s'il y a un utilisateur en train de modifier les positions du soleil. On regarde dans un fichier texte, si les valeurs rentrées par l'utilisateur datent de plus de 10 minutes. Si ces valeurs sont entrées dans ces 10 minutes, on prend en compte la position rentrée par l'utilisateur et on l'indique aux deux moteurs. Si elles ne sont pas modifiées pendant 10 minutes, alors on envoie aux moteurs les positions du soleil calculé par le programme. Enfin, on termine le programme.

Figure 12 : Diagramme du programme « Pythonpoint.py »



On obtient au final des résultats similaires avec les résultats trouvés sur le tableau Excel.

Valeur du programme

```
>>>
('Julian Day :', 2458088.541666667)
('Julian Century :', 0.17915240702715884)
('Geom Mean Long Sun (deg) :', 250.09103960660104)
('Geom Mean Anom sun (deg) :', 6806.84561521215)
('Eccent Earth Orbit :', 0.0167010990781305)
('Sun Eq of Ctr', -1.065194209638753)
('Sun True Long (deg):', 249.0258453969623)
('Sun True Anom (deg) :', 6805.7804210025115)
('Sun Rad Vector (AUs) :', 0.9861040473968202)
('Sun App Long (deg):', 249.01699025312845)
('Mean Obliq Ecliptic (deg) :', 23.43696138098705)
('Obliq Corr (deg):', 23.435043022032307)
('Sun Rt Ascen (deg):', -112.68514093695846)
('Sun Declin (deg) :', -21.798017389641554)
('Var y :', 0.04301848816447057)
('Eq of Time (minutes):', 11.058338392615203)
('HA Sunrise (deg):', 69.04310687220443)
('Solar Noon (LST) :', 0.5312094872273506)
('Sunrise Time (LST) :', 0.33942307924900494)
('Sunset Time (LST) :', 0.7229958952056963)
('Sunlight Duration (minutes) :', 552.3448549776355)
('True Solar Time (minutes) :', 5.021227672615197)
('Hour Angle (deg):', -178.7446930818462)
('Solar Zenith Angle (deg) :', 158.29384834538547)
('Solar Elevation Angle (deg) :', -68.29384834538547)
('Approx Atmospheric Refraction (deg) :', 0.002296879519805397)
('Solar Elevation Corrected for atm refraction (deg) :', -68.291551)
('Solar Azimuth Angle (deg cw from N) :', 3.152783343523481)
>>> |
```

Valeur de l'Excel

```
Julian Day : 2458088.50
Julian Century : 0.17915127
Geom Mean Long Sun : 250.05
Geom Mean Anom Sun : 6806.8045
Eccent Earth Orbit : 0.016701
Sun Eq of Ctr : -1.06635
Sun True Long : 248.9836
Sun True Anom : 6805.738
Sun Rad Vector : 0.98611
Sun app Long : 248.9748
Mean Obliq Ecliptic : 23.43696
obliq Corr : 23.43504
Sun Rt Ascen : -112.73
Sun Decline : -21.7915
Var Y : 0.043018
Eq of Time : 11.073907
HA Sunrise : 69.05068
Solar Noon : 0.53817
Sunrise Time : 0.34636144
Sunset Time : 0.72997
Sunlight Duration : 552.40546
True Solar Time : 5.036796
Hour Angle : -178.741
Solar zenith Angle : 158.2872
Solar Elevation Anglet : -68.2872
Approx Atmospheric Refraction : 0.002298
Solar Elevation Corrected for atm refraction : -68.2849
Solar Azimuth Angle : 3.16179
```

Figure 13 : Comparaison des valeurs obtenues

3/ Conversion et Cron tab

On sait que les moteurs ont une plage de fonctionnement comprise entre 0 et 96 000 points.

On aura donc le Sud situé à 48 000 points soit 180 degrés. De plus les moteurs peuvent varier de 46 degrés maximum, donc :

- 96 000 → 180+46=226 degrés

- 0 → 180-46=134 degrés

$$\text{Coef directeur} = \frac{48000}{48} = 1043,5$$

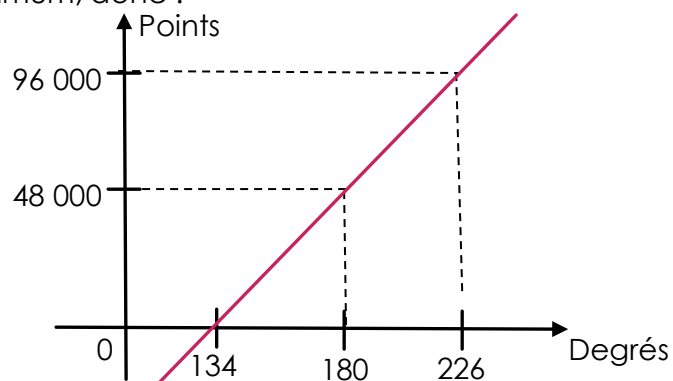
donc Points = -1043,5xdegrés+B

$$-1043,5 \times 226 + B = 96000$$

$$B = 235\,831$$

$$\text{Point azimute} = -1043,5 \times \text{degrés} + 235\,831$$

Il faut maintenant que le programme s'exécute toutes les 10 minutes. On va donc utiliser la fonction « crontab » de putty.



La façon la plus simple de créer un fichier `crontab` consiste à utiliser la commande `crontab -e`. Cette commande ouvre l'éditeur de texte qui a été défini pour votre environnement système.



```
GNU nano 2.7.4 File: /tmp/crontab.Gm8qCT/crontab
* * * * * /usr/bin/python /home/pi/Pythonpoint.py
[ Read 3 lines ]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

Figure 14 : Programmation de la Cron tab

Pour utiliser cron, ajoutez simplement les entrées à votre fichier `crontab` (situé dans le répertoire `/etc`). Une entrée dans cron. Les champs, dans l'ordre:

- *minute,*
- *heure,*
- *jours dans le mois,*
- *mois.*
- *jour de la semaine*
- La commande à lancer.

Ici « `* * * * *` » car pour les tests nous exécutons le programme toutes les minutes. Pour l'exécuter toutes les 10 minutes il suffit de remplacer la première étoile par « `10` »

Puis s'ensuit par le chemin de notre programme.

Il faut maintenant vérifier la fonctionnalité du `crontab`.

Quel que soit le type de programme écrit, script ou autres, il est forcément intéressant à un moment donné de tracer ce qui se passe durant l'exécution. Cela permet de pouvoir analyser une exécution ayant entraîné une erreur par exemple.

Par principe ce genre d'informations est placé dans un fichier de log, extension « .log », également appelé journal en français.

Les logs sont répartis en différents niveaux. Chaque niveau correspond à une importance donnée. Par défaut cinq niveaux sont prédéfinis. S'il vous est possible de définir vos propres niveaux, chose qui ne sera pas vue dans ce tutoriel, les cinq niveaux prédéfinis seront cependant suffisants dans la plupart des cas.

Niveau défini	Niveau pris en compte					Description
	C	E	W	I	D	
CRITICAL (C)	✓					Désigne en général une erreur ayant entraîné le plantage de l'application
ERROR (E)	✓	✓				Désigne une erreur sévère, mais n'ayant pas entraîné le crash du programme.
WARNING (W)	✓	✓	✓			Désigne un élément/événement qui pourrait avoir son importance : détection d'instabilité, seuil atteint...
INFO (I)	✓	✓	✓	✓		Niveau d'information standard. Permet de tracer les événements classiques. Par exemple, le lancement d'une fonction, transfert d'un fichier...
DEBUG (D)	✓	✓	✓	✓	✓	Ne sert que pour le débogage. Sert à tracer des informations détaillées : valeur de variables, trame de communication...

Figure 15 : Tableau des niveaux

On importe ensuite notre module.

Nous définissons notre fichier de log. Nous passons tout d'abord le filename, en chemin relatif (ici au même niveau que notre script) ou absolu. Ensuite, nous définissons le niveau de sensibilité, puis enfin le format de sortie.

```
import logging
logging.basicConfig(filename='/var/log/tracker/example.log',level=logging.DEBUG)
```

Figure 16 : Fichier de log

Il faut par la suite transmettre les messages de logs.

```
azimut_pt = Conv_Deg_PtA
logging.debug("azimut en point : %s", azimut_pt)

elevation_pt=Conv_Deg_PtE
logging.debug("elevation en pt : %s", elevation_pt)
```

Figure 17 : Transmission des messages log

Une fois terminé on peut visualiser les résultats à l'aide de la fonction « tail », tail affiche la dernière partie (par défaut : 10 lignes) de chacun des fichiers indiqués.

```
pi@raspberrypi:~ $ tail -f /var/log/tracker/example.log
DEBUG:root:L'heure actuelle est : 15:43
DEBUG:root:Solar Elevation Corrected for atm refraction (deg) : 37.515784567
DEBUG:root:Solar Azimut Corrected for atm refraction (deg) : 230.770509471
DEBUG:root:azimut en point : 17089.7908171
DEBUG:root:elevation en pt : 20631.995566
```

Figure 18 : Résultat de la Cron tab

Ici on y retrouve l'heure, l'élévation, l'azimut ainsi que leur conversion en points. Le tout s'exécute toutes les minutes.

IV/ Mécanique

1/ Conception du boîtier d'étanchéité

Le panneau solaire est fixé à un moteur, permettant de changer l'inclinaison de celui-ci. Ce moteur est lui-même fixé sur un boîtier en aluminium, lui-même fixé au deuxième moteur permettant de gérer la rotation du panneau. Ce moteur est fixé à un tube support.

Le but de ce boîtier est de protéger les deux moteurs de l'humidité et de la pluie, et aussi de renforcer la structure qui soutient le panneau. Le problème du précédent boîtier est surtout l'étanchéité de celui-ci. En effet, les différentes plaques étaient vissées entre elles, et de nombreux trous permettaient l'infiltration de l'eau dans le boîtier. Pour rigidifier la structure, un câble a été mis en place entre le boîtier et le panneau. Dernier point à modifier pour le futur boîtier, les câbles d'alimentation des moteurs et celui du panneau n'avaient pas assez de place dans le boîtier.

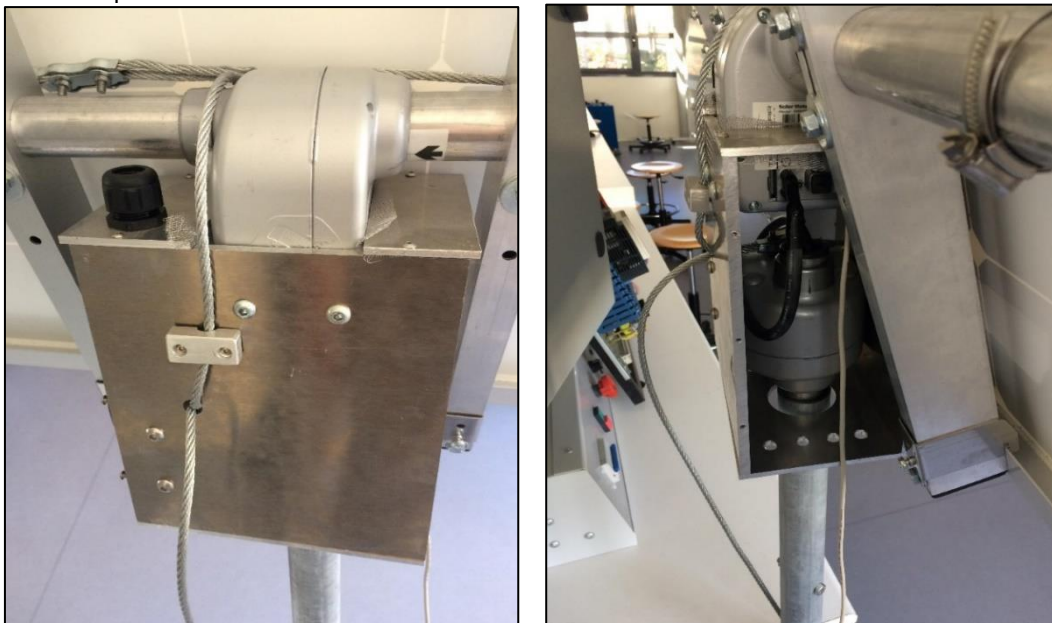


Figure 19 : Boîtier du groupe précédent

La première étape a été de récupérer les pièces sur SolidWorks du précédent assemblage panneau/moteur/boîtier. En récupérant ces différentes pièces, on pouvait récupérer les différentes parties qu'on allait garder et améliorer. Nous

avons donc allongé le boîtier pour donner plus de place aux câbles à l'intérieur de celui-ci. Pour pouvoir accéder au câble, nous avons supprimé une des plaques du haut du boîtier. Nous avons pu garder l'emplacement des trous prévue pour fixer les moteurs et les plaques entre elles. Le grand changement sera de souder toutes les plaques entre elles pour renforcer l'étanchéité et la solidité du boîtier. Une seule pièce en aluminium sera juste fixée avec des vis, la pièce en vert sur l'assemblage, pour pouvoir positionner le boîtier sur le moteur vertical.

Pour assurer cette étanchéité, nous avons pensé à rajouter un cache en aluminium sur le boîtier et sur le moteur, pour éviter que la pluie entre dans celui-ci

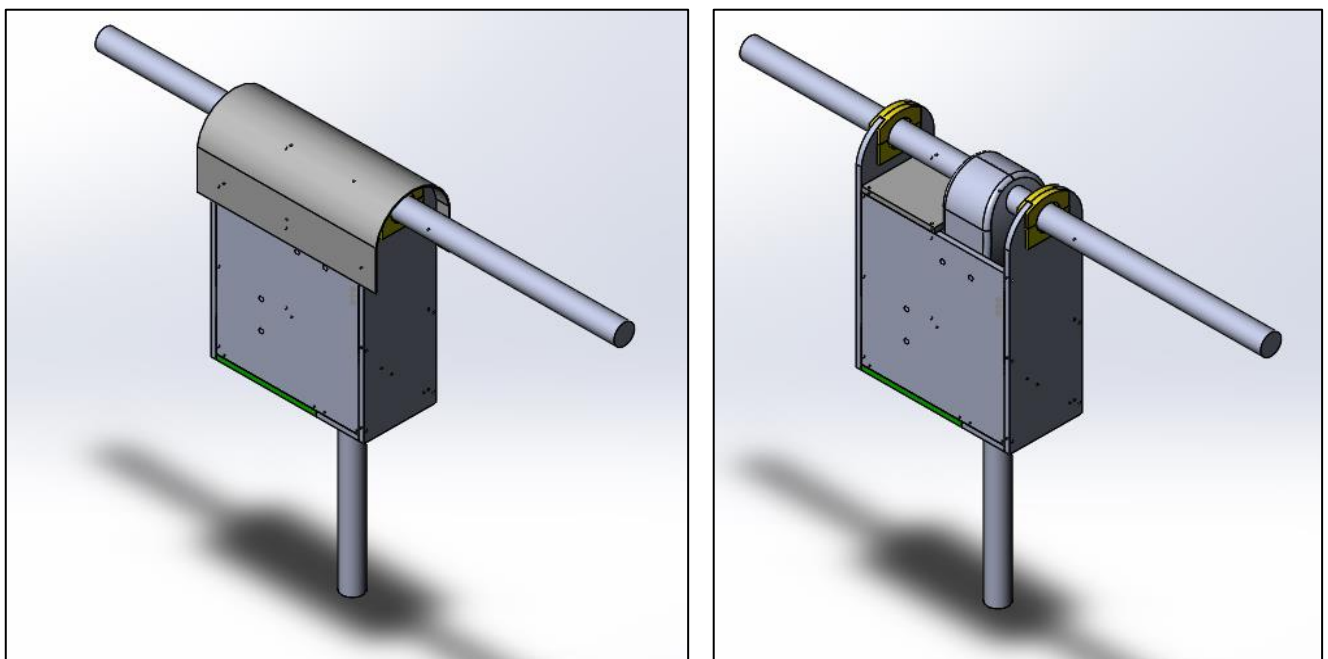


Figure 20 : Boîtier final sur Solidworks

En plus de la partie métallique, il était nécessaire de fabriquer des pièces en plastiques supplémentaire pour le passage du tube moteur supportant le panneau horizontalement et le tube moteur vertical. Ces pièces seront ensuite imprimées en ABS, une matière qui résiste mieux aux conditions climatiques extérieures, et qui est plus robuste que le PLA, une matière plastique plus couramment utilisées. Ces pièces sont en jaunes sur l'assemblage final du boîtier sur SolidWorks. Enfin, nous avons choisi de mettre un roulement à la sortie des câbles dans le tube vertical, pour éviter que ceux-ci ne frottent contre le métal et se sectionnent. Pour fixer ce roulement au reste du boîtier, nous avons créé une pièce sur SolidWorks où l'on encastre le roulement dedans et on fixe la partie basse de la pièce sur le tube vertical.

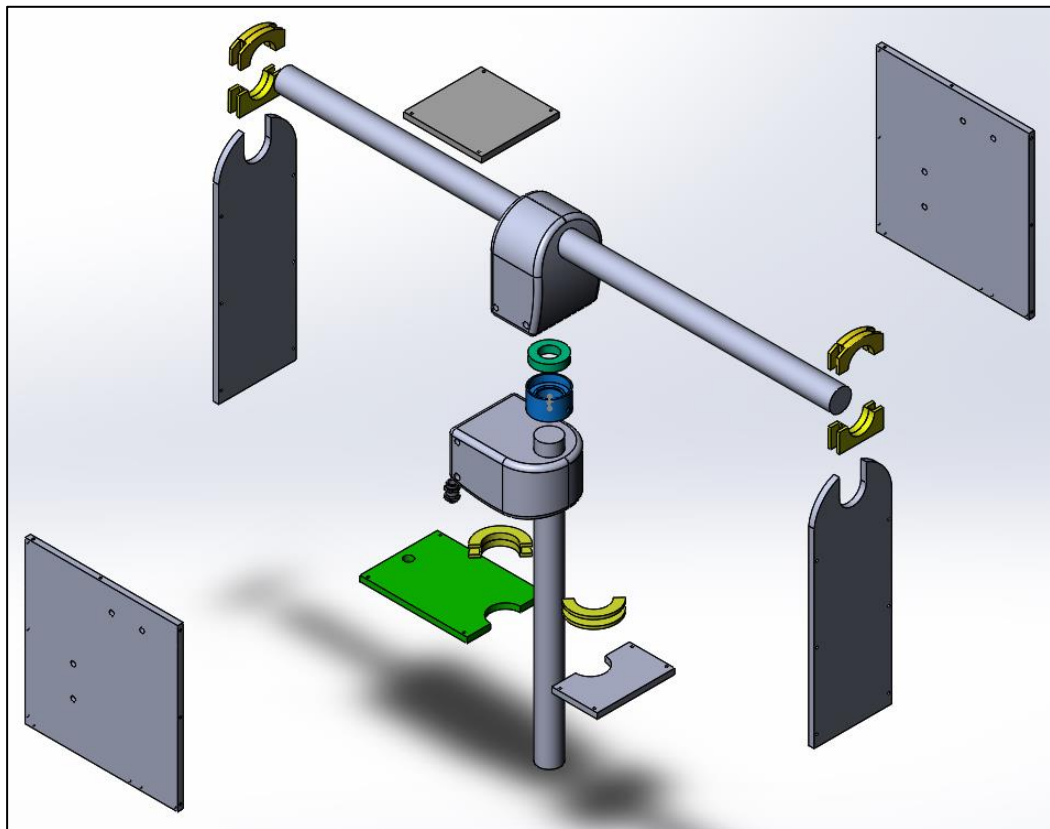


Figure 21 : Vue éclatée de l'assemblage final

Après vérification du boîtier sur SolidWorks, nous avons commandé les sept plaques en aluminium chez ADD. Une fois les plaques réceptionnées, nous avons commencé l'usinage des différents trous, le taraudage et il fallait faire les chanfreins pour fixer les différentes parties avant d'envoyer le boîtier à souder. Une fois le boîtier soudé, nous avons installé celui-ci avec les moteurs et le panneau solaire.

2/ Maintenance du moteur

Sur les deux moteurs à notre disposition dans le boîtier, un seul ne fonctionner pas. On entendait bien le moteur tourner, mais dans le vide. Après avoir hésité à en racheter un nouveau, nous avons remarqué qu'il était possible de commander des pièces détachées sur le site du fabricant.

Il a donc fallu démonter le carter du moteur pour accéder à l'intérieur et trouver d'où venait le problème. Une fois le carter enlevé, nous avons remarqué que deux pignons étaient cassés au niveau des dents.



Figure 22 : Pignons du moteur "azimut"

Une fois les deux pignons repérés, il fallait donc les remplacer par les nouveaux. Nous avons donc installé les nouveaux pignons, puis huiler de nouveau le tout et remonter le carter.

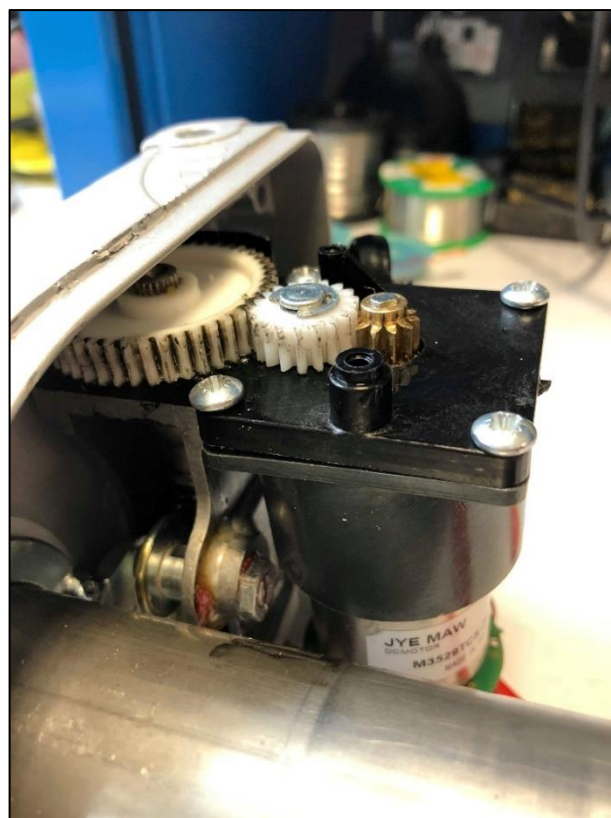
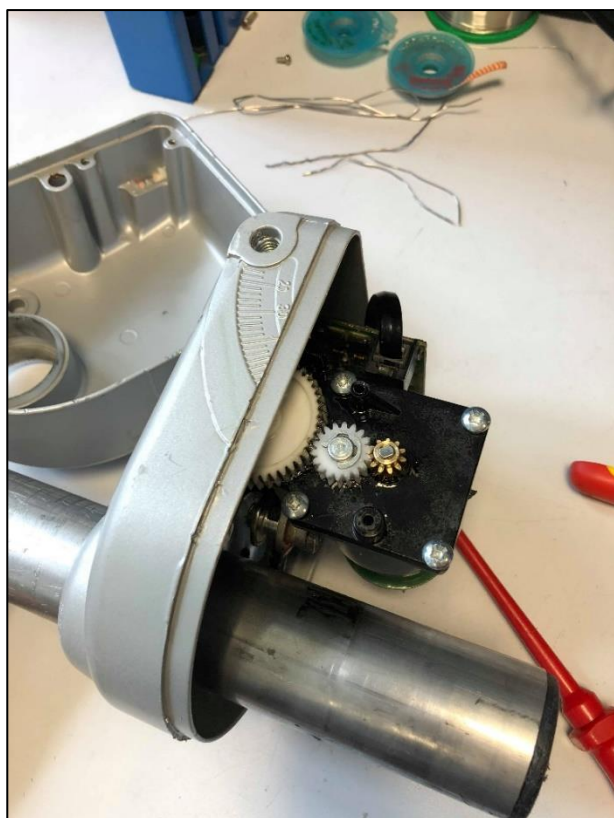


Figure 23 : Démontage du moteur à réparer

Une fois le moteur remonté, nous avons effectué des tests pour voir si la réparation à bien fonctionner. Le moteur a bien tourné, il a donc fallu le synchroniser de nouveau sur un PC (cf. annexe 5)

V/ Conclusion

La réalisation de notre projet a été une expérience très enrichissante de par sa variété de conception.

Nous avons acquis des compétences dans la programmation en langage python, en usinage ainsi que le décryptage de l'équation solaire.

Grâce à ce projet nous pouvons dire que les technologies liées aux énergies solaires ont une nécessité d'évoluer de par leur faible rendement. Le trackeur solaire est une solution qui améliore l'efficacité du panneau photovoltaïque. De plus, la mise en place d'un système intelligent couplé à une batterie permet d'augmenter la pénétration de cette énergie intermittente.

L'amélioration du programme (levé et couché du soleil), nous permet d'avoir un rendement maximal et donc utiliser aux mieux toutes les capacités du tracker solaire.

Le boîtier quant à lui permet une amélioration de sa qualité ainsi qu'une durée de vie plus élevée.

Pour la prochaine promotion, il serait envisageable de travailler sur l'implantation du panneau solaire sur le toit ainsi que la création d'une page internet avec toutes les données météo mais aussi les caractéristiques du tracker solaire à l'instant T.

VI/ Liste des figures

Figure 1 : Course apparente du soleil	7
Figure 2 : Synoptique du fonctionnement des équations.....	8
Figure 3 : Courbe obtenue les années précédentes	9
Figure 4 : Angle horaire	10
Figure 5 : Valeurs sur Excel	12
Figure 6 : Résultats solaires exactes	12
Figure 7 : Comparaison des résultats et de ceux d'internet en fonction des saisons..	12
Figure 8 : Logo raspberry.....	13
Figure 9 : Raspberry Pi B+	14
Figure 10 : Logo Python.....	15
Figure 11 : Graphiques montrant la fonction "Sécurité" du panneau solaire.....	15
Figure 12 : Diagramme du programme « Pythonpoint.py ».....	17
Figure 13 : Comparaison des valeurs obtenues	18
Figure 14 : Programmation de la Cron tab	19
Figure 15 : Tableau des niveaux	20
Figure 16 : Fichier de log	20
Figure 17 : Transmission des messages log	20
Figure 18 : Résultat de la Cron tab	20
Figure 19 : Boitier du groupe précédent	21
Figure 20 : Boitier final sur Solidworks	22
Figure 21 : Vue éclatée de l'assemblage final	23
Figure 22 : Pignons du moteur "azimut"	24
Figure 23 : Démontage du moteur à réparer	24

ANNEXE 1 : Tableau Excel

[illegible]

ANNEXE 2 : Programme Python

```
• from datetime import date
• import time
• import datetime
• from math import sin
• from math import radians
• from math import cos
• from math import atan2
• from math import degrees
• from math import asin
• from math import tan
• from math import acos
• import math
• import time
• import serial

• import logging
• logging.basicConfig(filename='/var/log/tracker/example.log', level=logging.DEBUG)

• d0 = date(1900, 1, 1)
• today = datetime.datetime.now()
• y = today.year
• m = today.month
• j = today.day
• d1 = date(y,m,j)
• delta = d1 - d0
• d=delta.days
• h = today.hour
• m = today.minute
• lm=time.strftime("%H:%M")
• logging.debug("L'heure actuelle est : %s", lm)
• e = ((h*60)+m)/1.44
• e = e/1000
• f=d+2415018.5+e-1/24
• g=(f-2451545)/36525
• l=(280.46646+g*(36000.76983+g*0.0003032))%360
• j=357.52911+g*(35999.05029-0.0001537*g)
• k=0.016708634-g*(0.000042037+0.000000121267*g)
• l=sin(radians(j))*(1.914602-g*(0.004817+0.000014*g))+sin(radians(2*j))*(0.019993-0.000101*g)+sin(radians(3*j))*0.000289
• m=1+l
• n=j+l
• o=(1.000001018*(1-k*k))/(1+k*cos(radians(n)))
• p=-0.00569-0.00478*sin(radians(125.04-1934.136*g))
• q=23+26+((21.448-g*(46.815+g*(0.00059-g*0.001813)))/60)/60
• r=q+0.00256*cos(radians(125.04-1934.136*g))
• s=degrees(atan2(cos(radians(r))*sin(radians(p)),cos(radians(p))))
• t=degrees(asin(sin(radians(r))*sin(radians(p))))
• u=sin(radians(t/2))/cos(radians(t/2))
• v=4*degrees(u*sin(2*radians(l))-2*k*sin(radians(j))+4*k*u*sin(radians(j))*cos(2*radians(l))-0.5*u*u*sin(4*radians(l))-1.25*k*k*sin(2*radians(j)))
• w=degrees(acos(cos(radians(90.833))/(cos(radians(43.4791056))*cos(radians(t)))-tan(radians(43.4791056))*tan(radians(t))))
• x=(720-4*1-v+1*60)/1440
• y=x-w*4/1440
• z=x+w*4/1440
• aa=0*w
• ab=(e**1440+vv+4*(-1.5092778)-60*(1))%1440
• if (ab/4) < 0:
•     ac=ab/4+180
• else:
•     ac=ab/4-180
• ad=degrees(acos(sin(radians(43.4791056))*sin(radians(t))+cos(radians(43.4791056))*cos(radians(t))*cos(radians(ac))))
• ae=90-ad
• if ae > 85:
•     af=0
• elif ae > 5:
•     af=58.1/tan(radians(ae))-0.07/pow(tan(radians(ae)),3)+0.000086/pow(tan(radians(ae)),5)
• elif ae > -0.575:
•     af=1735+ae*(-518.2+ae*(103.4+ae*-12.79+ae*0.711))
• else:
•     af=-20.772/tan(radians(ae))
• ag=af/3600
• ag=ae+af
• logging.debug("Solar Elevation Corrected for atm refraction (deg) : %s", ag)
• if ac > 0:
•     ah=(degrees(acos(((sin(radians(43.4791056))*cos(radians(ad)))-sin(radians(t)))/(cos(radians(43.4791056))*sin(radians(ad)))+180))%360
• else:
•     ah=(540-degrees(acos(((sin(radians(43.4791056))*cos(radians(ad)))-sin(radians(t)))/(cos(radians(43.4791056))*sin(radians(ad)))+180))%360
• logging.debug("Solar Azimut Corrected for atm refraction (deg) : %s", ah)

• Conv_Deg_PtA = (-584.69*ah+152019) #azimut
• Conv_Deg_PtE = (989.61*ag-16494) #?levation

• azimut_pt = Conv_Deg_PtA
• logging.debug("azimut en point : %s", azimut_pt)

• elevation_pt=Conv_Deg_PtE
• logging.debug("elevation en pt : %s", elevation_pt)

• if azimut_pt >= 96000:
•     azimut_pt = 96000

• if azimut_pt <= 0:
•     azimut_pt = 0
• if elevation_pt <= 0:
•     elevation_pt = 0
• if elevation_pt <= 0:
•     elevation_pt = 0

• if int(azimut_pt) >=96000:
•     val_azimut = 96000
•     print "val_azimut = ",val_azimut
• elif int(azimut_pt) <= 0:
•     cal_azimut = 0
•     print "val_elevation = ",val_elevation
• else:
•     val_azimut = int(azimut_pt)
•     print "val_azimut = ",val_azimut


• if int(elevation_pt) >=96000:
•     val_elevation = 96000
•     print "val_elevation = ",val_elevation
• elif int(elevation_pt) <= 0:
•     cal_elevation = 0
•     print "val_elevation = ",val_elevation
• else:
•     val_elevation = int(elevation_pt)
•     print "val_elevation = ",val_elevation

• try:
•     ser=serial.Serial('/dev/ttyACM0',38400,timeout=1)
•     ser.write("$."+str(val_azimut)+"\n")
•     logging.debug("azimut en point : %s", azimut_pt)
•     ser.close()
• except:
•     print("pb com moteur 1")
• try:
•     ser=serial.Serial('/dev/ttyACM1',38400,timeout=1)
•     ser.write("$."+str(val_elevation)+"\n")
•     logging.debug("elevation en pt : %s", elevation_pt)
•     ser.close()
• except:
•     print("pb com moteur 2")
```

ANNEXE 3 : Bons de commande

Chez RS :


✓ 2 En stock pour livraison dès le lendemain

 **Disque à lamelles Aluminium oxyde de zirconium 3M Dia. 125mm**
Code commande RS : 153-555
Marque : 3M
Référence fabricant : 3M55MD 861793
Statut RS: Non applicable

2 Mettre à jour | Supprimer

2,91 € la pièce 5,82 €


✓ 1 En stock pour livraison dès le lendemain

 **Bois-Cloche RS Pro Dents M3 HSS - corps en alliage d'aluminium 44mm Bc**
Métal : profondeur de coupe 32mm
Code commande RS : 215-956
Marque : RS Pro
Statut RS: Non applicable

1 Mettre à jour | Supprimer

9,26 € la pièce 9,26 €

✓ 1 En stock pour livraison dès le lendemain

 **Bois-Cloche RS Pro Dents M3 HSS - corps en alliage d'aluminium 44mm Bc**
Métal : profondeur de coupe 32mm
Code commande RS : 215-934
Marque : RS Pro
Statut RS: Non applicable

1 Mettre à jour | Supprimer

10,60 € la pièce 10,60 €


[Afficher les accessoires](#)

[Tout effacer](#) [Mettre à jour](#)

Livraison

Le jeu 22 mars pour les commandes passées avant 19h00 (s'applique uniquement aux produits en stock).

✓ 1 En stock pour livraison dès le lendemain


 **Vis à six pans creux M4 x 16mm M6 Vis bombées à tête creuse hexagonales en Acier inoxydable A4 316 Bout**
Code commande RS : 232-8215
Marque : RS Pro
Statut RS: Conforme

1 Mettre à jour | Supprimer

10,66 € la boîte de 50 10,66 €

[Afficher les accessoires](#)

✓ 1 En stock pour livraison dès le lendemain

 **Roulement à billes RS Pro 6006-2RS Rainure croisée Flanche 30mmx55mm**
Code commande RS : 619-0480
Marque : RS Pro
Statut RS: Non applicable

1 Mettre à jour | Supprimer

5,30 € la pièce 5,30 €

[Afficher les accessoires](#)

[Tout effacer](#) [Mettre à jour](#)

Chez ADD Acier :

Produit	Description	Prix unitaire	Quantité	Total	
	ALUMINIUM 2017A (AU4G) Réf. : 2017ATL10NALC L:277.5mmxH:293mm	29,27 €	2	58,54 €	
	ALUMINIUM 2017A (AU4G) Réf. : 2017ATL10NALC L:141mmxH:405mm	29,27 €	2	41,12 €	
	ALUMINIUM 2017A (AU4G) Réf. : 2017ATL10NALC L:141mmxH:195mm	9,90 €	1	9,90 €	
	ALUMINIUM 2017A (AU4G) Réf. : 2017ATL10NALC L:82.5mmxH:141mm	4,19 €	1	4,19 €	
	ALUMINIUM 2017A (AU4G) Réf. : 2017ATL10NALC L:141mmxH:142mm	7,21 €	1	7,21 €	
				Total produits TTC :	120,95 €
				Frais de port :	11,67 €
				Total HT :	112,46 €
				Total TTC :	20,16 €
				TOTAL	132,62 €

BONS DE RÉDUCTION

[ok](#)

Chez SatControl :

SAT CONTROL

Want to get more?™

HOME PAGE | CART | CONTACT | GENERAL BUSINESS TERMS | SECURITY | SUPPORT | MONITORING PROGRAMS | REFERENCES | NEWS | REQUEST FOR TECHNICAL SUPPORT | WORLDWIDE DEALERS | RP

SOLAR TRACKERS

DUAL AXIS SUN TRACKERS

SINGLE AXIS SUN TRACKERS

SPARE PARTS & ACCESSORIES

SECOND HAND - REFURBISHED

LINEAR MOTORS / ACTUATORS

LINEAR MOTORS

ACTUATORS

ACCESSORIES

SPARE PARTS

SLEWING DRIVES

SLEWING DRIVES

SOLAR POSITIONERS

EXTERNAL DIN RAIL

Home page > Cart

[Continue with shopping](#)

Empty cart content

Item name	Price excl. tax	VAT	Quantity	Total
Set of gears for GM45-R214 - spare parts for SM3-OG+ / S041 / SETZCBGM45-R214-SM3	8,00	0%	1	8,00

Calculate values

Tax base 8,00

[Checkout](#)

Search

[Find](#)

Log in

E-mail

Password

[Log in](#)

Are you a new user?

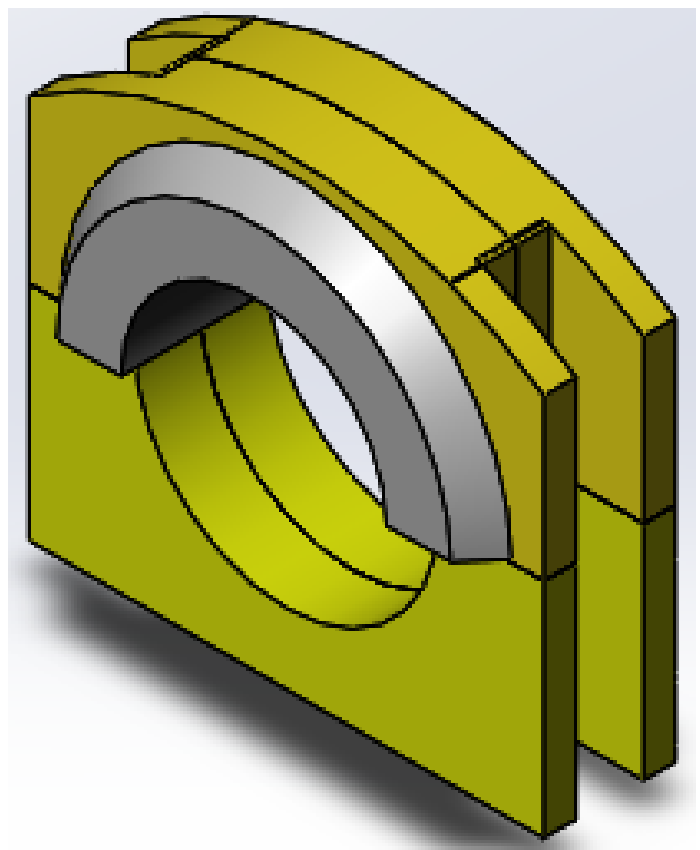
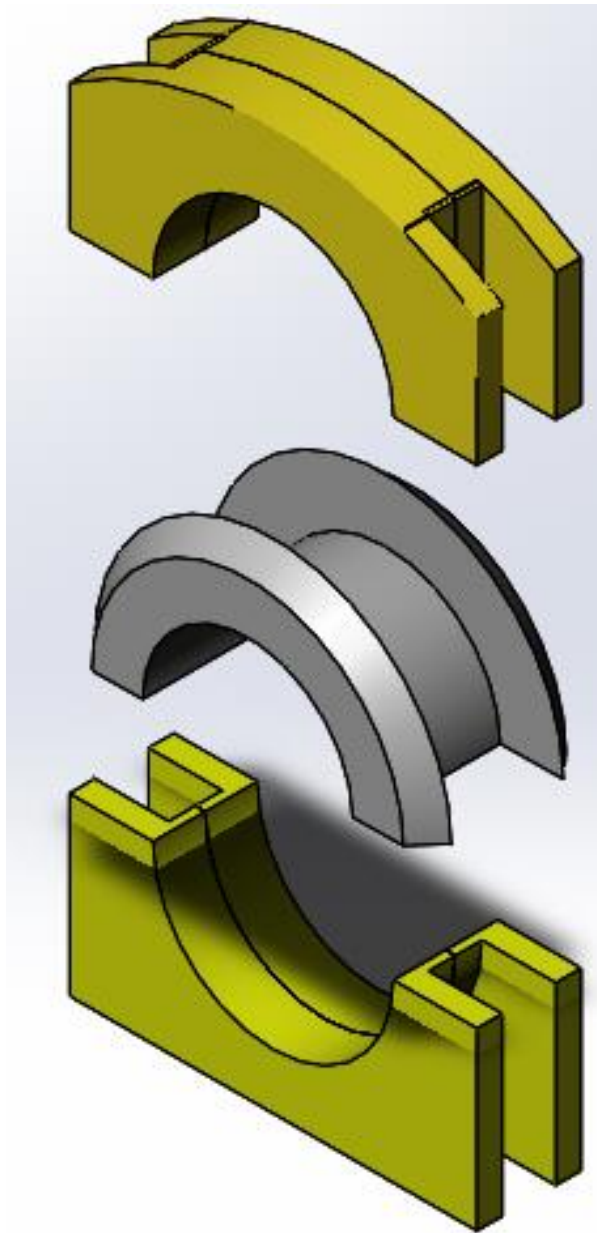
Did you forget your password?

Mailing list

E-mail

[Login](#) [Logout](#)

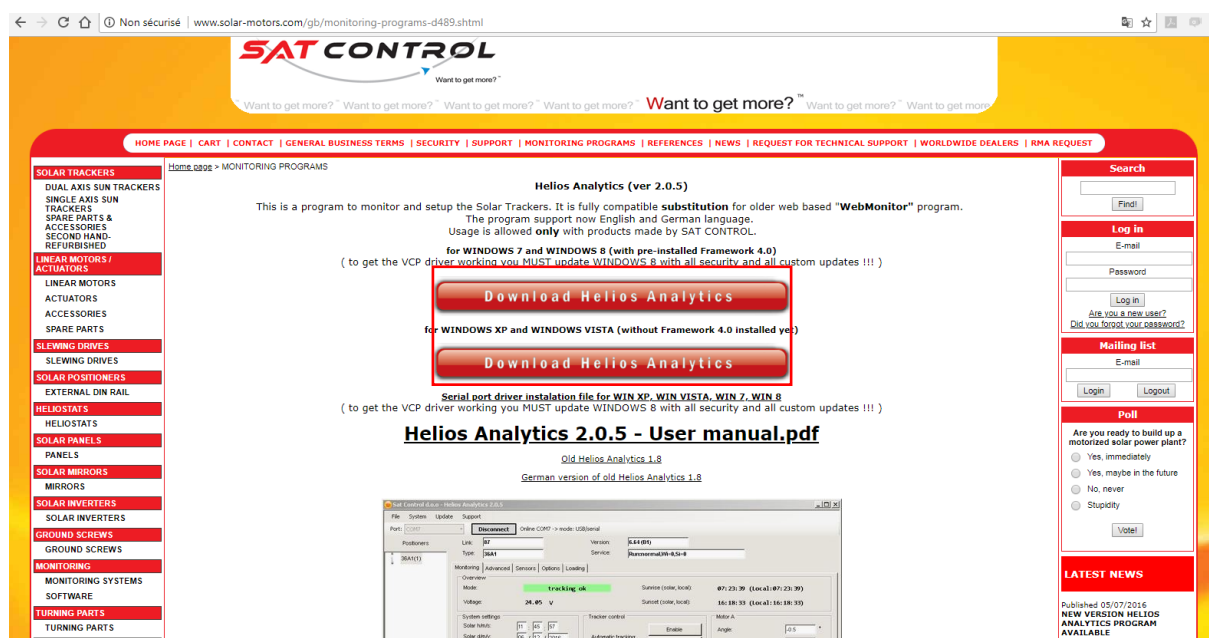
ANNEXE 4 : SolidWorks des pièces en plastique



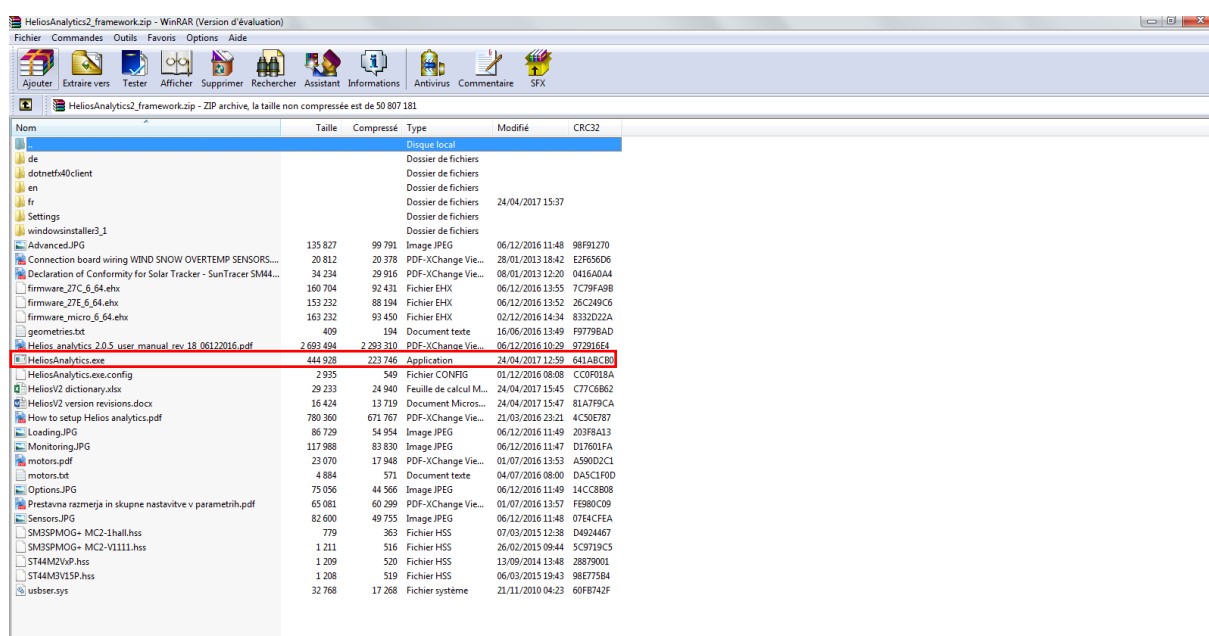
ANNEXE 5 : Tutoriel pour synchroniser le moteur sur PC

Tout d'abord rendez-vous sur le site <http://www.solar-motors.com/gb/monitoring-programs-d489.shtml>

Il faut ensuite choisir la bonne version du logiciel adapté à votre Windows

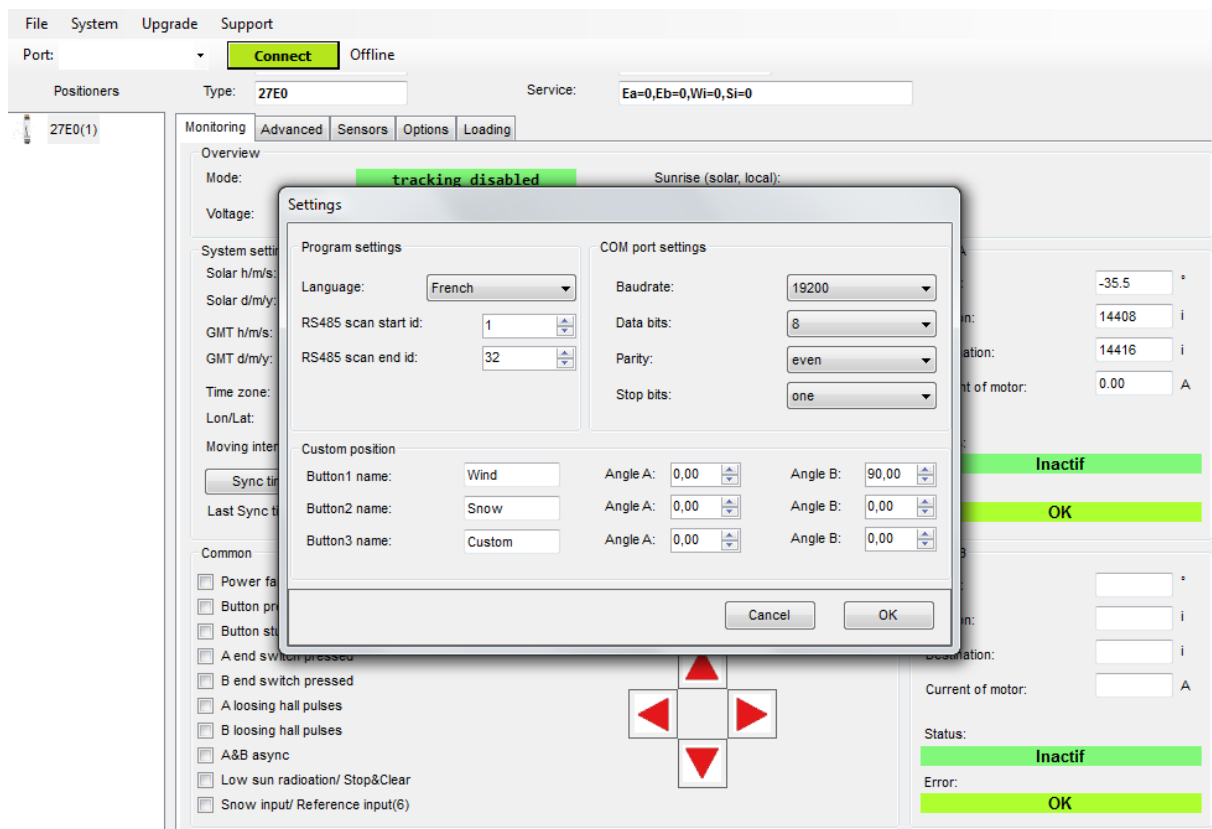
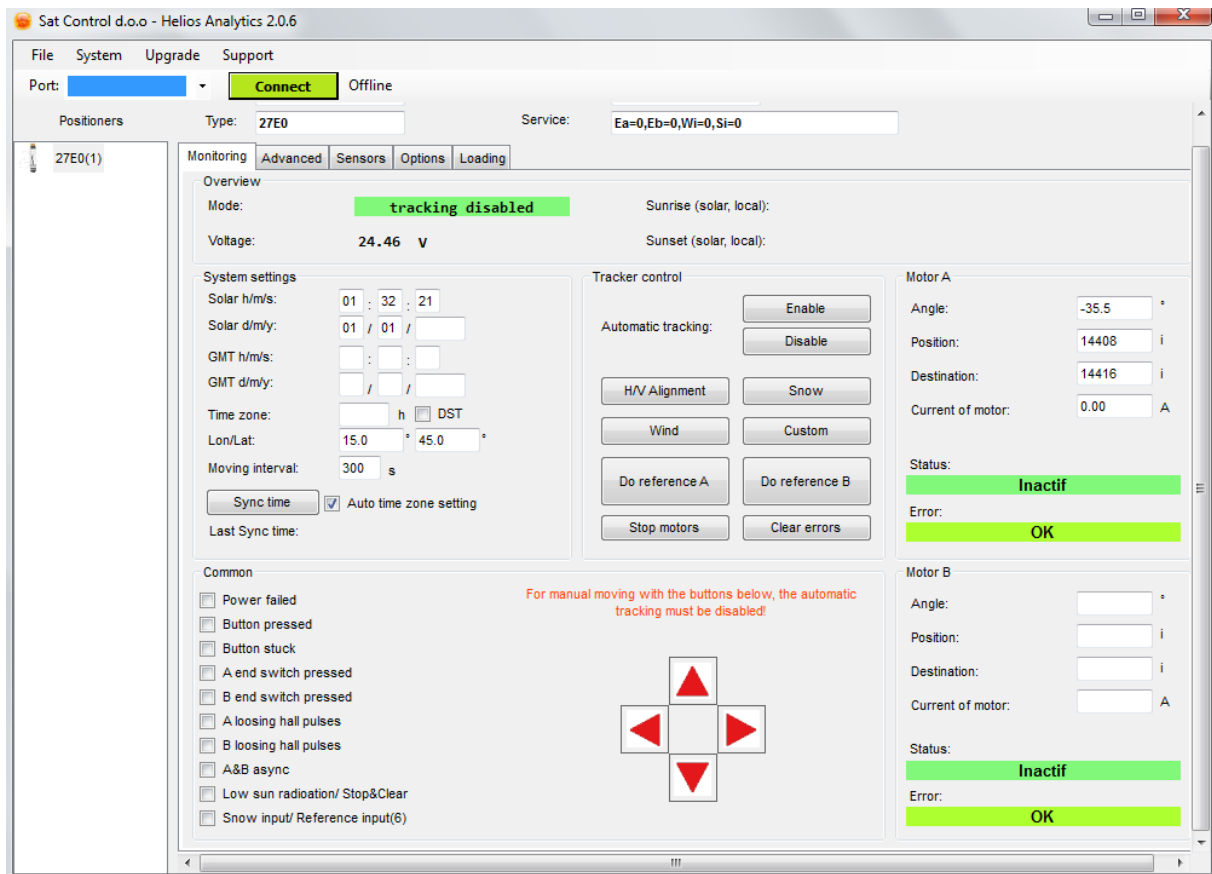


Une fois le téléchargement terminé, double cliquer sur « HeliosAnalytics.exe », le logiciel s'installe.



Nom	Taille	Compressé	Type	Modifié	CRC32
de			Dossier de fichiers		
dotnetfx40client			Dossier de fichiers		
en			Dossier de fichiers		
fr			Dossier de fichiers	24/04/2017 15:37	
Settings			Dossier de fichiers		
windowsinstaller_1			Dossier de fichiers		
Advanced.JPG	135 827	99 791	Image JPEG	06/12/2016 11:48	98F91270
Connection board wiring WIND SNOW OVERTEMP SENSORS...	20 812	20 378	PDF-XChange Vie...	28/01/2013 18:42	E2F650D6
Declaration of Conformity for Solar Tracker - SunTracer SM4...	34 234	29 916	PDF-XChange Vie...	08/01/2013 12:20	0416A0A4
firmware_27C_6_64.ehx	160 704	92 431	Fichier EHX	06/12/2016 13:55	7C79FA98
firmware_27E_6_64.ehx	153 232	88 194	Fichier EHX	06/12/2016 13:52	26C249C6
firmware_micro_6_64.ehx	163 232	93 450	Fichier EHX	02/12/2016 14:34	8332022A
geomtries.txt	409	194	Document texte	18/06/2016 13:49	F97799AD
Helios_analytics_2.0.5_user_manual_rev_18_06122016.pdf	2 693 404	2 293 310	PDF-XChange Vie...	06/12/2016 10:29	972916E4
HeliosAnalytics.exe	444 928	223 746	Application	24/04/2017 12:59	641ABC80
HeliosAnalytics.exe.config	2 935	549	Fichier CONFIG	01/12/2016 08:08	C0F018A
HeliosV2 dictionary.xlsx	29 233	24 940	Feuille de calcul M...	24/04/2017 15:45	C77C6862
HeliosV2 version revisions.docx	16 424	13 719	Document Micros...	24/04/2017 15:47	81A7F9CA
How to setup Helios analytics.pdf	780 360	671 767	PDF-XChange Vie...	21/03/2016 23:21	4C50E787
Loading.JPG	86 729	54 954	Image JPEG	06/12/2016 11:49	203F8A13
Monitoring.JPG	117 988	83 830	Image JPEG	06/12/2016 11:47	D17601FA
motors.pdf	23 070	17 948	PDF-XChange Vie...	01/07/2016 13:53	A58002C1
motors.txt	4 884	571	Document texte	04/07/2016 08:00	D45C1F0D
Options.JPG	75 056	44 566	Image JPEG	06/12/2016 11:49	14C83808
Predstavna razmera in skupne nastavitve v parametih.pdf	65 081	60 299	PDF-XChange Vie...	01/07/2016 13:57	F580C0C9
Sensors.JPG	82 600	49 755	Image JPEG	06/12/2016 11:48	07E4CFEA
SM3SPMOG+ MC2-1hall.hss	779	363	Fichier HSS	07/03/2015 12:38	D4924467
SM3SPMOG+ MC2-V1111.hss	1 211	516	Fichier HSS	26/02/2015 09:44	5C9719C5
ST44M2Vp.hss	1 209	520	Fichier HSS	13/09/2014 13:48	28879001
ST44M3V1SP.hss	1 208	519	Fichier HSS	06/03/2015 19:43	98E77584
usbser.sys	32 768	17 268	Fichier système	21/11/2010 04:23	60F8742F

Voici l'interface du logiciel. Il faut maintenant mettre la langue en français en allant dans Système→Settings→language et sélectionner « French ».



Relier le moteur à l'ordinateur à l'aide du câble USB.

Attendre (5min) que les pilotes s'installent sur l'ordinateur

Une fois les périphériques installés, sélectionner « COM3 » puis cliquer sur « connecter ».

The screenshot shows the software interface with the 'Port' dropdown set to 'COM3' and the 'Connecter' button highlighted with a red rectangle. The interface includes a menu bar (Fichier, Système, Mettre à niveau, Support), a status bar (Port: COM3, Type: 27E0, Service: Ea=0, Eb=0, Wi=0, Si=0), and a main panel with tabs for Surveillance, Paramètres avancés, Capteurs, Options, and Chargement. The 'Surveillance' tab is active, displaying system parameters like Mode (tracking disabled), Tension (24.46 V), and various control buttons for the sun tracker and motors. The 'Connecter' button is highlighted in red.

Dernière étape, il ne reste plus qu'à cliquer sur « Synchronisation du moteur A ».

The screenshot shows the software interface with the 'Synchronisation du moteur A' button highlighted with a red rectangle. The interface is similar to the previous one, but the 'Connecter' button is now 'Déconnecter'. The 'Synchronisation du moteur A' button is highlighted in red, indicating the next step in the process. The status bar shows 'En ligne COM3 -> Mode: USB/serial'.

Le logiciel reconnaît le moteur, il ne reste plus qu'à cliquer sur « synchronisation du moteur A ». Le moteur va se réinitialiser à sa position initiale.

Ce logiciel va vous permettre de piloter manuellement les moteurs. D'autres options sont disponibles, il suffit de lire le manuel téléchargeable sur le site <http://www.solar-motors.com/gb/monitoring-programs-d489.shtml>

The screenshot shows the SAT CONTROL website's monitoring programs page. The main content area is titled "Helios Analytics (ver 2.0.5)" and provides instructions for downloading the software. It includes two prominent red buttons labeled "Download Helios Analytics" for different operating systems. Below these, there are links to a user manual and a serial port driver installation file. A small screenshot of the Helios Analytics software interface is also visible. The website has a yellow header with the SAT CONTROL logo and a navigation bar with various links. A sidebar on the left lists product categories, and a sidebar on the right contains a search bar, login fields, a mailing list sign-up, and a poll.