

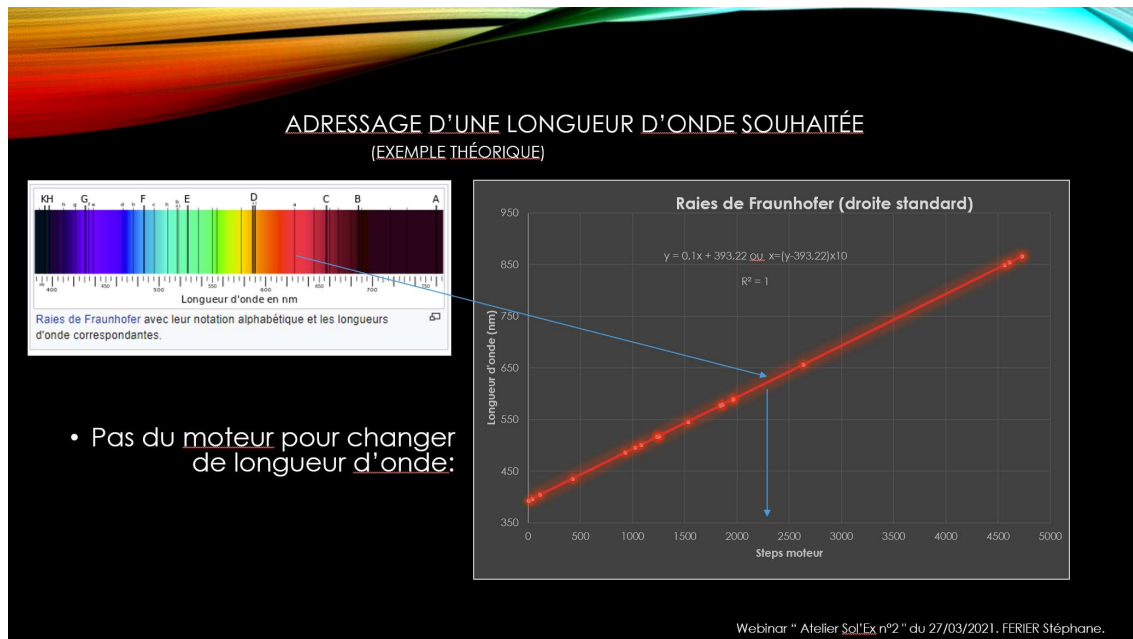
Motorisation du Sol'Ex (par Jean Brunet et Stéphane Ferier)

Bref historique :

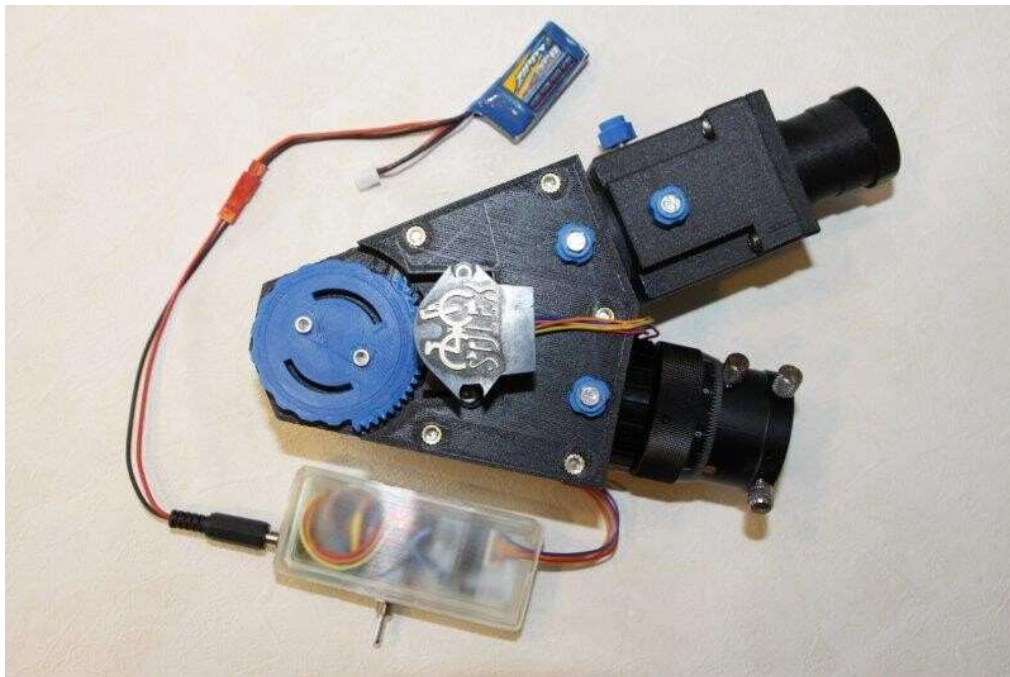
Au début de l'aventure Sol'Ex, j'ai rapidement souhaité motoriser le spectroscopie, afin de faciliter à l'utilisateur la recherche des bandes d'intérêt.

En effet, la multitude de bandes du spectre solaire ne permet pas de trouver son chemin de manière aisée pour le débutant en spectroscopie. Le simple fait de rechercher la raie $H\alpha$ peut prendre un temps précieux, surtout lorsque le matériel est à remettre en station si l'on ne dispose pas d'un abri astro poste fixe. Poser la monture, l'orienter correctement pour pouvoir réaliser le suivi solaire (mise en station de jour) assemblage de la lunette et du Sol'Ex ; recherche de la raie d'intérêt ... Tout cela peut prendre beaucoup de temps et décourager à sortir le matériel si le ciel est nuageux et offre peu de « fenêtres de tir » !

L'idée était alors de motoriser la rotation du réseau, et de pouvoir enregistrer l'écart entre l'ordre 0 et la raie $H\alpha$. Réaliser une droite standard sur les raies d'intérêt majeures. Par la suite, calculer le nombre de pas moteur à réaliser pour pouvoir atteindre d'autres raies d'intérêt, en prenant en compte une approximation de linéarité dans la distribution des raies d'intérêt et des pas moteur pour les atteindre.



Une application de base (ordre de déplacement Gauche et Droite) basée sur l'ESP8266 gérée par Wifi avait été développée, ainsi qu'un prototype de support pour embarquer la motorisation.



MATÉRIEL

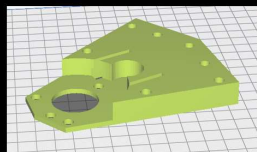
- ESP8266



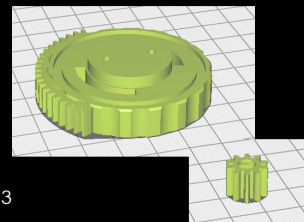
- HC-05 (BT)



- Socle adapté



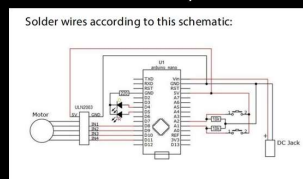
- Manette adaptée



- Moteur et contrôleur



- Schéma de pilotage actuel (interrupteur 3 positions & Arduino nano)

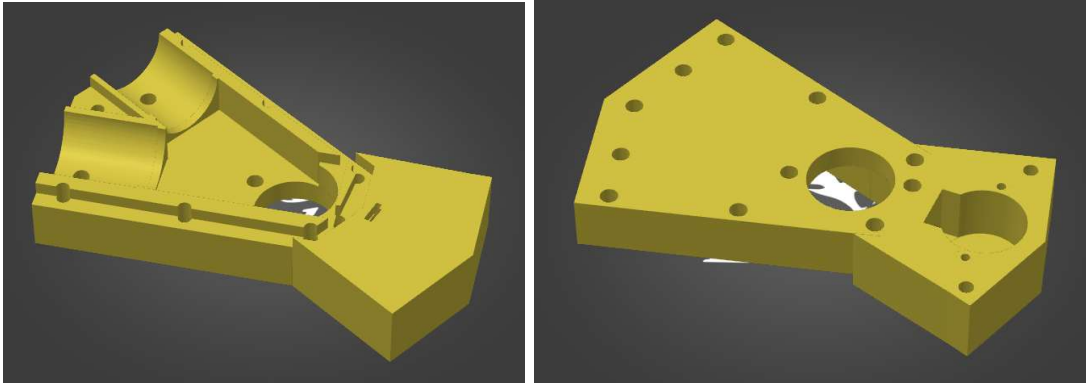


Webinar "Atelier Sol'Ex n°2" du 27/03/2021, FERRIER Stéphane,

Malheureusement, le déplacement moteur ne parvenait pas à entraîner la rotation du réseau. Je n'ai compris que récemment en discutant avec Jean que le problème provenait très sûrement du driver utilisé qui n'est pas optimisé pour donner les impulsions qui conviennent à ce type de moteur pas à pas.

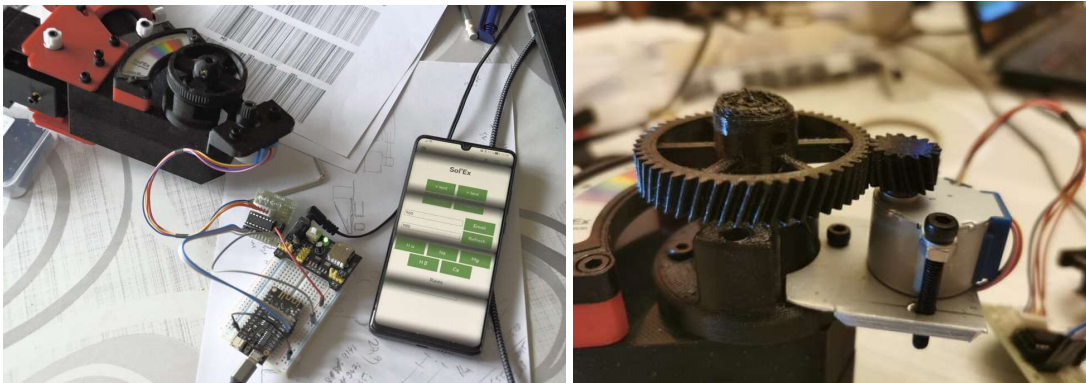
Un second prototype est alors envisagé, utilisant une courroie plutôt qu'un entraînement dent contre dent utilisé pour ce premier essai.

Dans le but de rendre le système le plus compact possible, j'ai alors redessiné la partie supérieure du corps principal du Sol'Ex pour y intégrer le moteur.

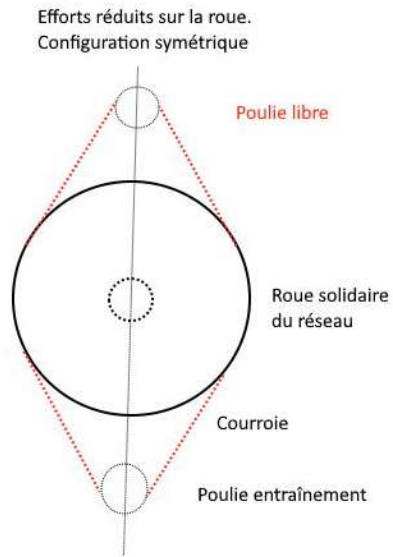


Cet essai, bien que plus « propre » intégrant le moteur dans l'axe du Sol'Ex n'a pas permis à ce moment d'améliorer la problématique de l'entraînement. (On sait maintenant pourquoi !!). Je désespérais de trouver une solution fonctionnelle.

De son côté, l'idée de la motorisation avait aussi germé dans l'esprit de Jean Brunet ! Son premier prototype nous était partagé pour la Saint Nicolas (... et oui, je suis Belge, et chez nous, c'est le jour des cadeaux 😊)



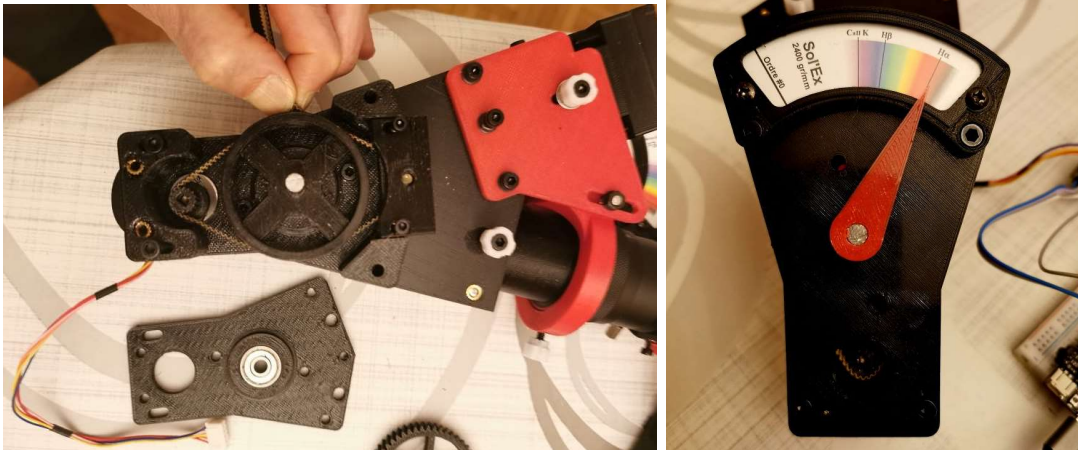
C'est alors que nous avons commencé à échanger sur le sujet offline, car dans nos conversations, il venait avec une idée géniale à mes yeux, revenir à un boîtier additionnel, utilisant des roulements à billes et un axe d'entraînement.



Les contraintes physiques de tension sur l'axe du porte réseau en ont profité pour disparaître ... De plus, Jean avait aussi réalisé un software parfaitement fonctionnel et apportant directement des « touches préprogrammées » pour aligner les raies majeures ! Par la suite, la poulie libre du schéma ci-dessus n'apportant rien de particulier a été enlevée.

Mon bonheur était fait ...

Voici son boîtier en version de test :



Il ne restait plus qu'à optimiser les pièces pour une réalisation plus aisée, utilisant deux poulies et une courroie, facilement trouvables par internet.

Pendant ce temps, Jean peaufinait la partie électronique en réalisant un PCB de test, et un boîtier adapté à son montage électronique. De mon côté, j'adaptais le programme à une roue dentée commerciale de 60 dents.

Nous vous proposons en annexes le fruit de notre travail commun, comprenant un montage mécanique simple, et deux options possibles pour la partie électronique, basées sur soit un PCB à graver, soit une plaque de prototypage comme illustré sur cette image.



La programmation de la motorisation est aussi expliquée en détails plus bas.

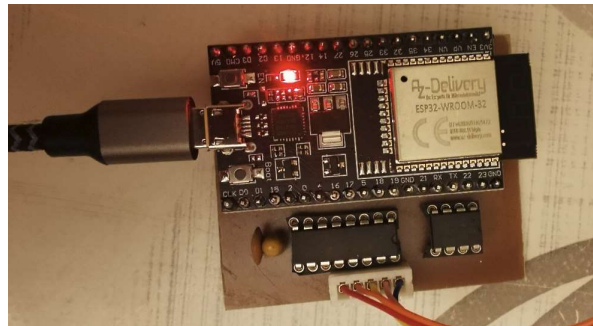
Concernant le matériel à vous procurer, tout se trouve aussi plus bas dans ce document, y compris des liens proposés pour trouver les pièces à commander.

Impressions 3D, montage électronique : Préambule.

Comme dit un peu plus haut, annexés à ce document, vous trouverez les fichiers .STL nécessaires qui vous permettront d'imprimer les bâtis inférieur et supérieur, ainsi que le nouveau support réseau adapté au montage.

Point d'attention suivant : il n'est pas donné à tout un chacun de savoir imprimer un PCB (carte de support pour les composants électroniques).

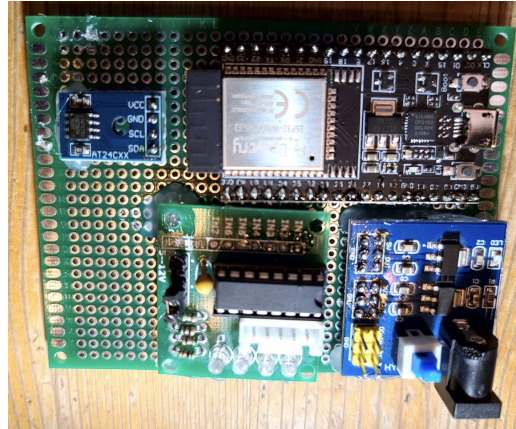
Vous trouverez donc deux options possibles : soit le plan pour imprimer votre PCB et y monter les composants électroniques, ainsi qu'un boîtier, soit le schéma pour monter les composants directement sur une plaque de prototypage. L'alimentation se fait par le port USB de l'ESP. C'est lui seul qui alimente aussi le moteur.



Dans le kit « prototypage », la partie électronique sera disposée sur une plaque de prototypage. Comme vous pourrez y disposer les composants à votre goût, il n'y a pas ici de fichiers STL pour le boîtier.

Un fichier PPT reprend les connections à réaliser entre les composants. L'avantage ici, c'est que sans les compétences pour réaliser le PCB, vous pouvez quand même réaliser la motorisation de votre Sol'Ex.

Autre avantage de l'alimentation proposée ici, c'est que vous pouvez alimenter idéalement en 12V (de 8 à 12V en réalité, mais je conseille le 12V), et que le module dispose d'un bouton « on/off » intégré. La réalisation d'un boîtier pourra dès lors utiliser cette fonction supplémentaire. Attention que pour bien fonctionner, l'ESP doit impérativement être alimenté par l'USB (dans les deux options).



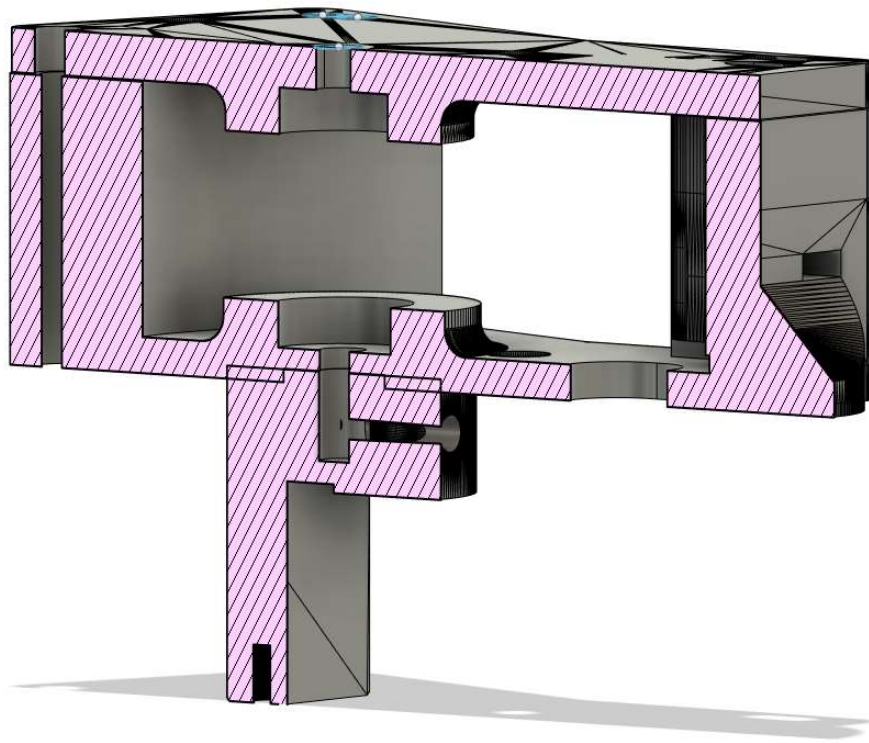
Pour l'impression des pièces 3D, il y a toujours le PETG, mais j'ai préféré utiliser le PLA+ ou PLA + Carbone de chez Sunlu. Ce matériau est plus dur une fois refroidi. Le PLA+ s'imprime à 225°C, sur un lit à 80°C. Afin d'améliorer l'adhérence lors de l'impression, utilisez de la laque 3D sur votre vitre d'impression. Vous pouvez aussi tester avec de la laque pour cheveux, mais celle-ci adhère moins que celle prévue pour la 3D.

Les pièces à imprimer en 3D :

La partie mécanique est composée :

- D'un nouveau support réseau adapté pour y insérer un axe 5mm long de 40mm (ou 50mm si vous souhaitez utiliser le vernier de Xavier au-dessus de la motorisation), qui servira à l'entraînement. Afin de fixer l'axe dans le porte réseau, on utilise un écrou Ruthex M3 et une vis sans tête. Il est aussi possible de coller l'axe dans le porte réseau, mais attention que l'axe métallique doit impérativement être bien vertical !!
- Le bâti inférieur qui recevra un roulement à billes et les poulies/courroie d'entraînement ;
- Le bâti supérieur qui recevra lui aussi un roulement à billes.

Coupe 3D des différentes pièces à imprimer :



Montage du kit de motorisation :

Comme nous avons déjà réalisé notre Sol'Ex, le plus difficile est loin derrière nous ! Il faudra donc ici remplacer précautionneusement le porte réseau initial par celui du kit.

La première chose à faire, avant de mettre le réseau en place, c'est de fixer l'axe dans le porte réseau. Vous pouvez le coller avec une colle deux composants, ou utiliser un filet Ruthex 3mm et une vis sans tête, à insérer dans l'orifice perpendiculaire à l'axe de rotation du porte réseau.

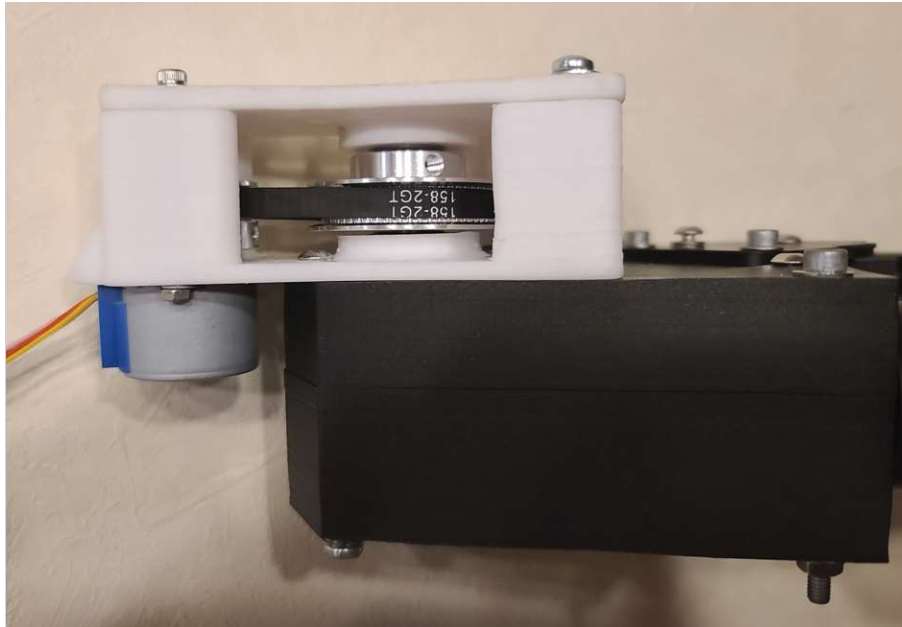


Le réseau sera ensuite disposé sur le porte réseau, puis le tout inséré dans le corps du Sol'Ex.

Ensuite, on vient disposer le bâti inférieur autour de l'axe et on le solidarise au Sol'Ex par les vis qui étaient utilisées pour fixer la manette réseau de Christian.

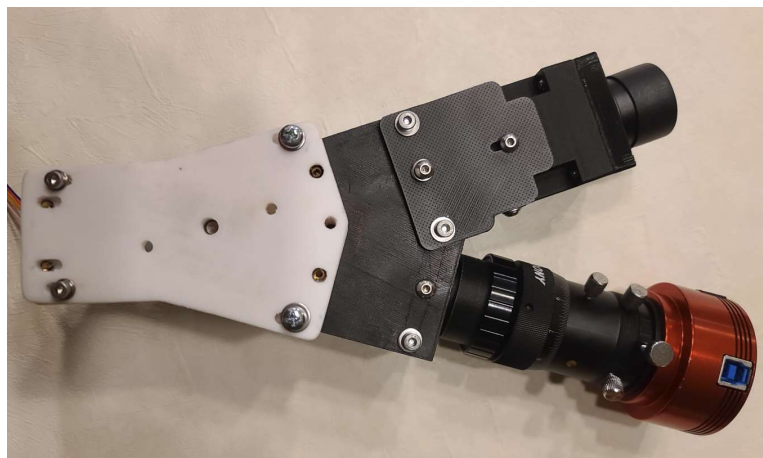


On dispose ensuite la poulie principale sur l'axe. Ici aussi, elle dispose de vis sans tête pour rendre l'axe et la poulie solidaires. Pensez à repérer l'orientation de l'ordre zéro du réseau par un trait sur le bâti et la poulie. On dispose ensuite la poulie qui s'adapte au moteur, ainsi que la courroie avant de fixer et tendre légèrement la courroie en fixant le moteur sur le bâti inférieur. Il se fixe à l'extérieur du montage.

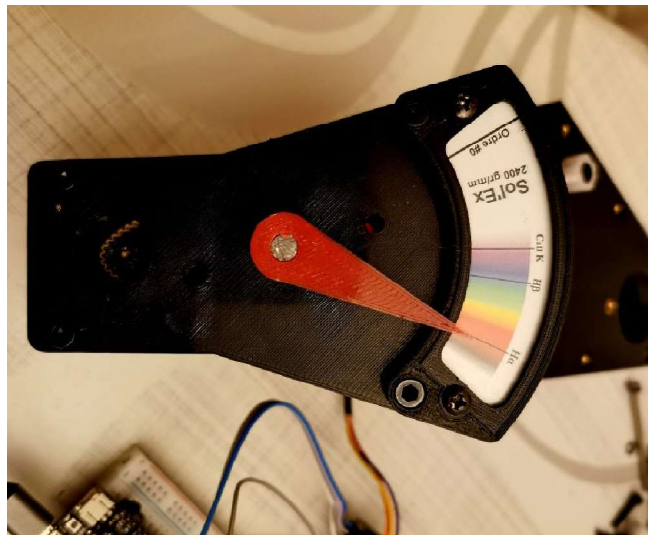


Comme on peut le voir sur cette image, les vis de maintien du moteur passent au travers des deux bâtis. On voit aussi ici les logements de vis sans tête de la poulie.

Le montage refermé ressemble à ceci :

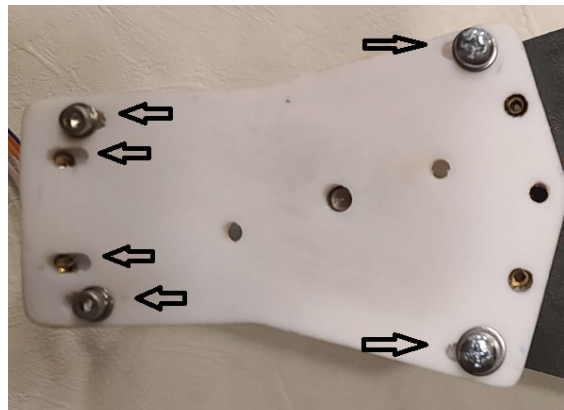


Deux longues vis M4 complètent le montage et solidarisent encore un peu plus l'ensemble. Le vernier de Xavier peut être aussi ajouté, afin de garder un repère visuel sur la position du réseau. Dans ce cas, l'axe de rotation devra dépasser du montage, afin de pouvoir y fixer une aiguille qui est à disposition dans le kit RAR.



Réglage de l'inclinaison de l'axe :

Dans certains cas, il se peut que l'image renvoyée par le réseau vers la caméra ne soit pas bien centrée ... Ce défaut d'alignement peut être corrigé en ajoutant une épaisseur de papier collant derrière le réseau pour l'incliner légèrement... Mais rassurez-vous, je n'en ai pas eu besoin.



La partie mécanique est terminée !!

Chargement du software dans l'ESP :

Cette partie peut paraître compliquée, pourtant il n'en est rien !

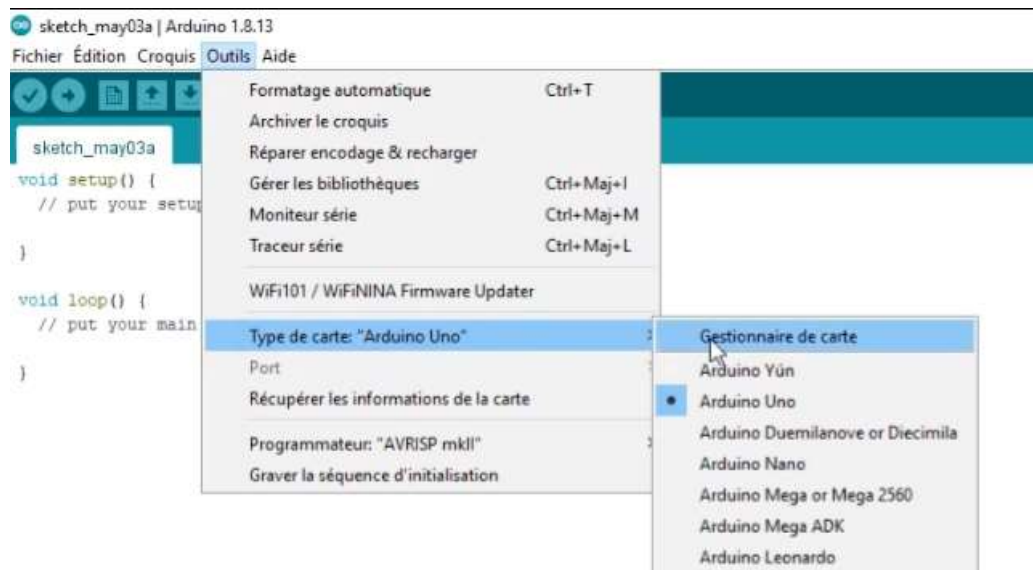
Nous partirons du principe que vous avez à disposition un PC, mais un MAC peut aussi faire l'affaire, car le software ARDUINO est disponible pour les deux plateformes. Il se télécharge gratuitement ici : <https://www.arduino.cc/en/software>



The screenshot shows the Arduino IDE 2.0.3 download page. On the left, there's a section with the Arduino logo and the text "Arduino IDE 2.0.3". Below this, it says "The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger." It also mentions "For more details, please refer to the [Arduino IDE 2.0 documentation](#)." and "Nightly builds with the latest bugfixes are available through the section below." There's a "SOURCE CODE" section stating "The Arduino IDE 2.0 is open source and its source code is hosted on [GitHub](#)." On the right, there's a "DOWNLOAD OPTIONS" section with links for Windows (Win 10 and newer, 64 bits; MSI installer; ZIP file), Linux (Appliance 64 bits (X86-64); ZIP file 64 bits (X86-64)), and macOS (Intel, 10.14: "Mojave" or newer, 64 bits; Apple Silicon, 11: "Big Sur" or newer, 64 bits). There's also a link for "Release Notes".

Une fois l'installation du programme faite, vous pouvez lancer son exécution. Il nous faut maintenant ajouter le module de gestion de la carte ESP qui nous intéresse comme ceci :

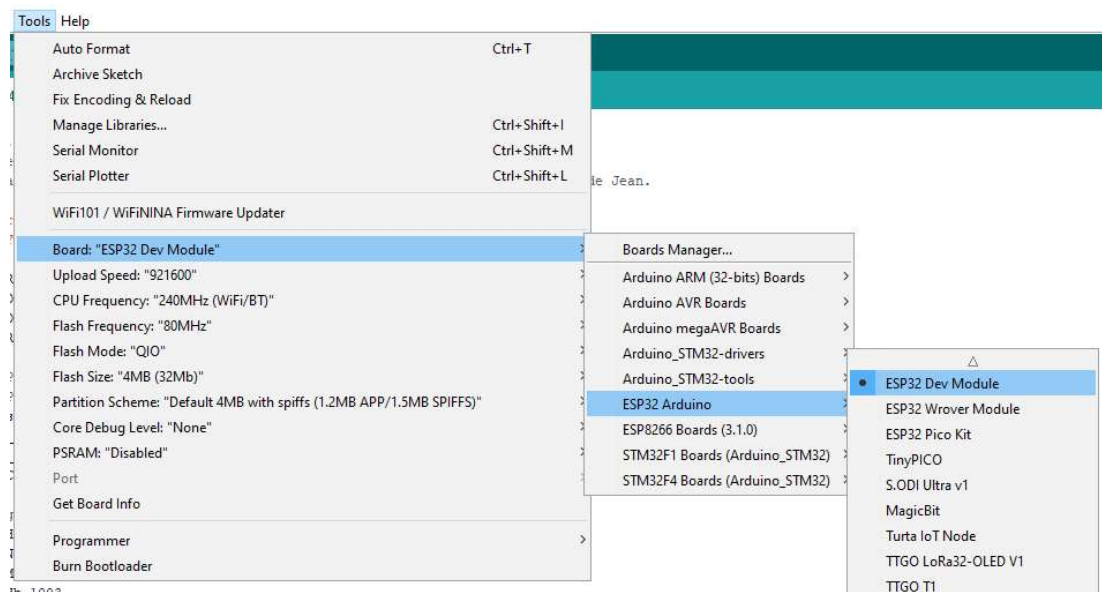
Ouvrez « Outil/Type de carte/Gestionnaire de carte »



Tapez « ESP32 », vous pouvez installer la version 1.0.6

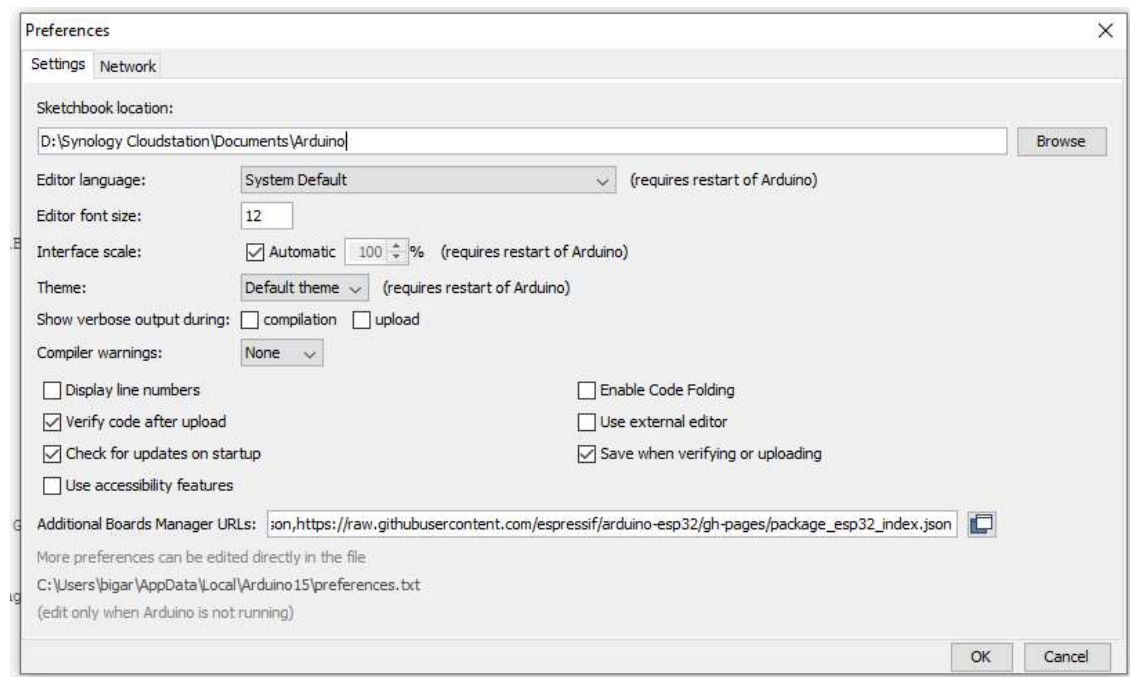


Sélectionnez ensuite ESP32 Dev Module :

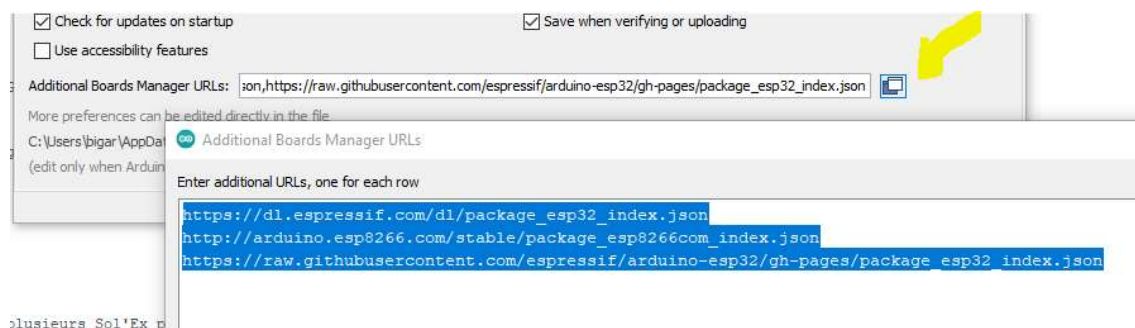


Vous déposez le dossier « Arduino » qui se trouve dans votre fichier RAR dans le dossier « Documents ».

Dans le menu « Fichier/Préférences » de l'application Arduino, indiquez le chemin du dossier où vous souhaitez déposer les fichiers de travail. Dans mon cas, j'ai choisi « Documents/Arduino » :



Tant qu'on est dans cette fenêtre, profitez de l'occasion pour ajouter dans « Gestionnaire URL de carte additionnelles » les lignes suivantes, qui pourront vous servir pour d'autres applications Arduino :



https://dl.espressif.com/dl/package_esp32_index.json

http://arduino.esp8266.com/stable/package_esp8266com_index.json

https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json

Vous pouvez refermer cette fenêtre.

Chargez maintenant le fichier « ESP_SolEx .INO » situé dans le dossier « Documents/Arduino/ »

```
ESP_solEx2_4b | Arduino 1.8.14 Hourly Build 2020/09/23 10:33
File Edit Sketch Tools Help

ESP_solEx2_4b

// programme de commande du réseau de Sol'Ex
// version 1.2b
// Jean Brunet le 17/01/2023
// Modifié par Stéphane Ferrier pour poulie 60 dents GT2 le 18/01/2023 avec l'accord de Jean.

#include <Wire.h>
#include <WiFi.h>

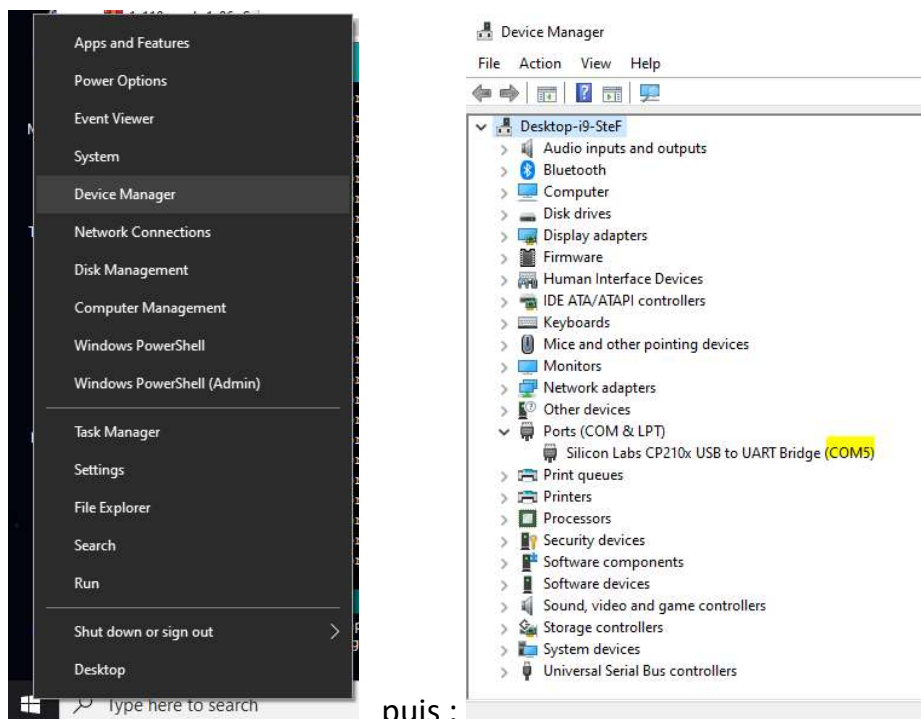
#define MOTOR_PIN_1 19 // Blue - 28BYJ48 pin 1
#define MOTOR_PIN_3 18 // Pink - 28BYJ48 pin 2
#define MOTOR_PIN_2 5 // Yellow - 28BYJ48 pin 3
#define MOTOR_PIN_4 17 // Orange - 28BYJ48 pin 4

//#define EEPROM_SDA 23
//#define EEPROM_SCL 19
#define adresse EEPROM 0x50 // eeprom externe 24LC01B
```

Connectez l'ESP32 par un câble USB à votre ordinateur, une led rouge doit s'allumer en continu sur l'ESP.

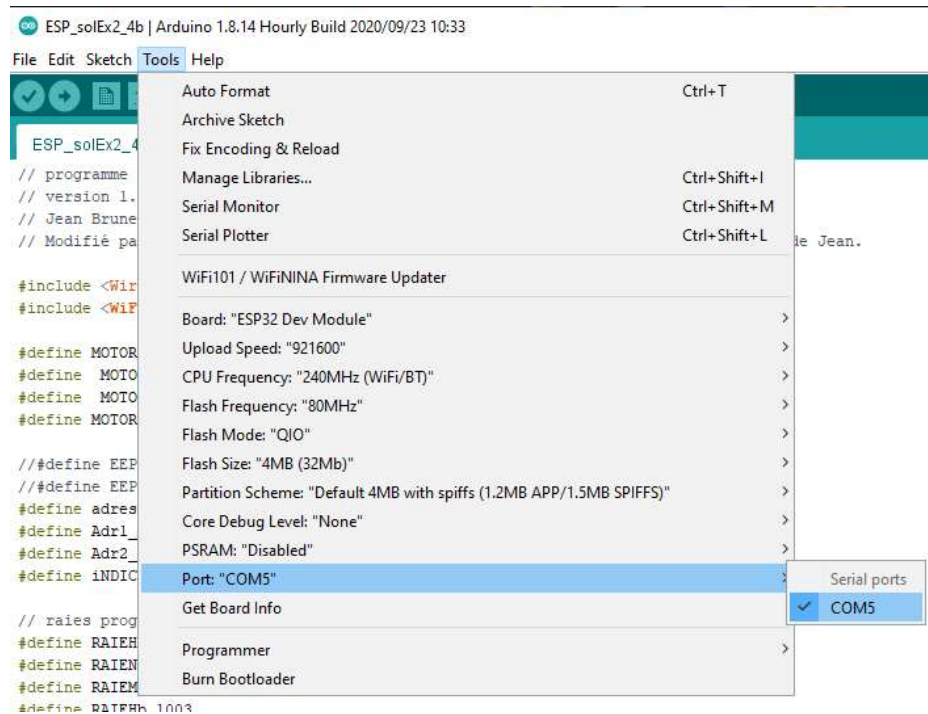
Vérifiez sur quel « port Com » l'ESP est connecté à votre PC/MAC :

Clic droit sur la fenêtre Windows ; Device Manager :



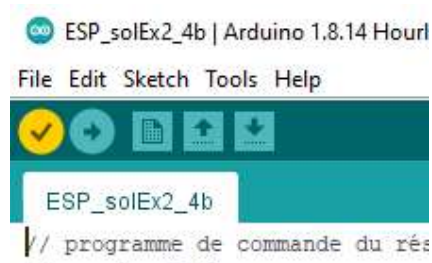
Dans mon cas, c'est le port « Com5 » qui est assigné à l'ESP.

Vérifiez ensuite l'équivalence de connexion ici :

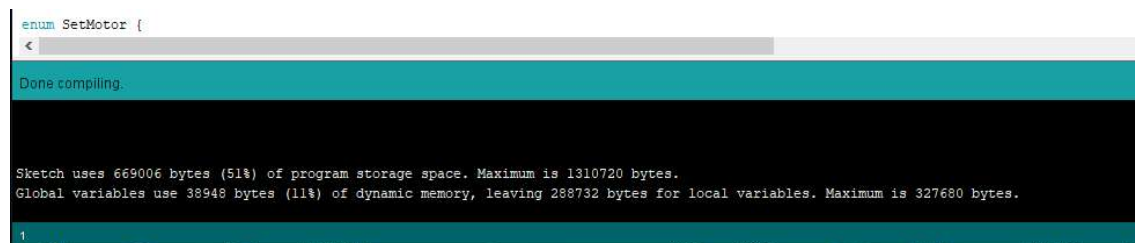


Vous êtes maintenant prêts à téléverser le programme dans l'ESP.

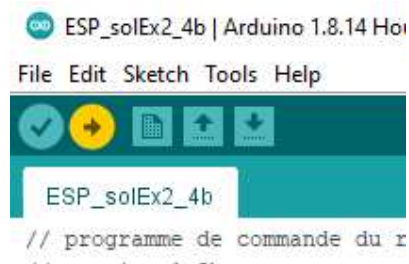
On vérifie que tout est prêt en cliquant sur le bouton « v » ici :



La console devrait vous indiquer ceci :



Dans ce cas, le téléversement peut s'effectuer en appuyant sur le bouton « boot » de l'ESP et en cliquant sur la flèche :



La console nous renseigne du statut de téléversement, et conclut par :

```
Writing at 0x00060000... (76 %)
Writing at 0x00060000... (80 %)
Writing at 0x00064000... (84 %)
Writing at 0x00068000... (88 %)
Writing at 0x0006c000... (92 %)
Writing at 0x00070000... (96 %)
Writing at 0x00074000... (100 %)
Wrote 669120 bytes (412149 compressed) at 0x00010000 in 6.6 seconds (effective 811.3 kbit/s)...
Hash of data verified.
Compressed 3072 bytes to 128...
Writing at 0x00008000... (100 %)
Wrote 3072 bytes (128 compressed) at 0x00008000 in 0.0 seconds (effective 2730.6 kbit/s)...
Hash of data verified.
```

Bravo, vous avez programmé votre ESP. On va pouvoir tester tout cela ...

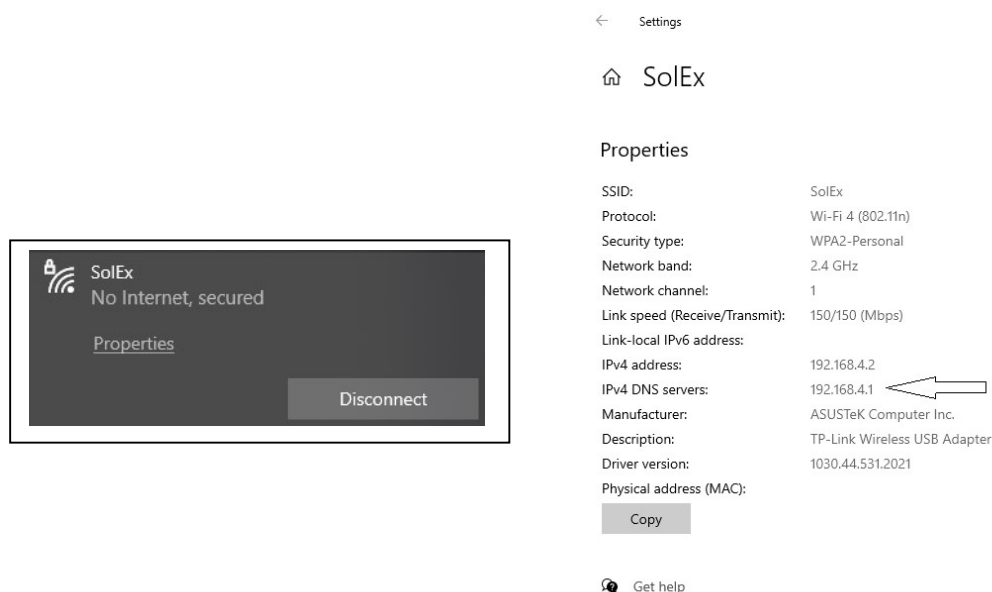
Réglage de position du moteur / ordre zero :

Alimentons notre montage ...

L'interface est appelée après connexion par WiFi au serveur Sol'Ex. Le password par défaut est : solex1234

Pour appeler l'interface, entrez l'adresse http suivante : 192.168.4.1

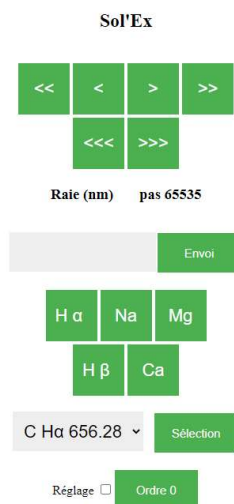
Si cette adresse est déjà utilisée par votre réseau, vous trouverez celle définie par votre DNS serveur pour le Sol'Ex ici : (ouvrez « Propriétés » de SolEx dans vos connections réseau WiFi)



The image shows a Windows Settings window for the SolEx network. On the left, a preview shows the SolEx status: 'No Internet, secured' with a 'Properties' link and a 'Disconnect' button. On the right, the 'Properties' section lists network details. An arrow points to the 'IPv4 DNS servers' field, which contains '192.168.4.1'.

Properties	
SSID:	SolEx
Protocol:	Wi-Fi 4 (802.11n)
Security type:	WPA2-Personal
Network band:	2.4 GHz
Network channel:	1
Link speed (Receive/Transmit):	150/150 (Mbps)
Link-local IPv6 address:	
IPv4 address:	192.168.4.2
IPv4 DNS servers:	192.168.4.1
Manufacturer:	ASUSTeK Computer Inc.
Description:	TP-Link Wireless USB Adapter
Driver version:	1030.44.531.2021
Physical address (MAC):	

L'interface apparaît comme suit :



The image shows the Sol'Ex web interface. It features a title 'Sol'Ex', a set of navigation buttons (left, right, first, last), a display showing 'Raie (nm) pas 65535', a table of spectral lines, and a 'Sélection' button. At the bottom, there is a 'Réglage' checkbox and an 'Ordre 0' button.

Sol'Ex			
<<	<	>	>>
<<<	>>>		
Raie (nm) pas 65535			
		Envoi	
H α	Na	Mg	
H β	Ca		
C H α 656.28	Sélection		
Réglage <input type="checkbox"/>	Ordre 0		

Cochez la case « réglage », puis cliquez sur la case verte/bleue « Ordre 0 » :

Réglage ☒ Ordre 0

Cela a pour effet de vider le champ « Raie (nm) » et de mettre le nombre de pas à zéro.

Raie (nm) pas 0

Utilisez ensuite les flèches pour ramener le réseau en position « ordre 0 »

<< < > >>
<<< >>>

... et ajustez précisément l'ordre 0 au centre de la fenêtre de votre caméra, dans la zone que vous utiliserez lors des scans du soleil.

Lorsque c'est fait, décochez la case « Réglage », afin de tester les longueurs d'ondes programmées.

Réglage ☐ Ordre 0

3 zones de sélections sont possibles pour faire pivoter le réseau vers la longueur d'onde cible :

Les cases préprogrammées (H α , Na, Mg, H β , Ca) ;

La zone à choix multiple et le bouton « Sélection » (30 raies d'intérêt y sont encodées) ;

Le champ libre et le bouton « Envoi ».

Envoi

H α Na Mg
H β Ca

C H α 656.28 Sélection

Selon votre choix, le moteur fait tourner le réseau jusqu'à atteindre la longueur d'onde de votre choix. L'affichage indique alors la longueur d'onde centrée et le nombre de pas moteur de cette position.

Raie (nm) 587.63 pas 1356

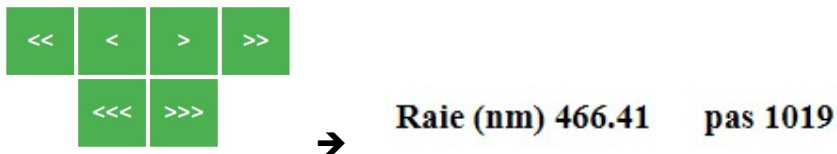
Vous avez ensuite le choix de changer de longueur d'onde d'intérêt en utilisant l'un des 3 modes de sélection : les cases ou champs préprogrammés, ou l'utilisation de la boîte d'envoi :





466.814	Envoi
---------	-------

En appuyant sur « Envoi », le moteur diminuera ou augmentera le nombre de pas depuis la position actuelle vers votre nouvelle cible par incrément de plusieurs pas moteur, jusqu'à approcher votre cible (mais ne se positionne pas avec précision)

Raie (nm) 478.05 pas 1049 ➔ Raie (nm) 464.84 pas 1015

Pour affiner la position, vous pouvez alors utiliser les boutons de direction jusqu'à atteindre votre cible avec plus de précision :



La flèche   incrémente de 2 pas moteur ;  10 pas ;  25 pas.

Si vous arrêtez la motorisation en coupant son alimentation, la dernière position est mémorisée dans l'EEPROM, de sorte que si vous n'avez pas modifié manuellement la position du réseau, lors du rallumage du système, les informations sont reprises là où elles ont été interrompues.

Il peut toutefois être utile (ou souhaité) de vérifier cette position. Dans ce cas, l'appui sur le bouton « Ordre 0 » fera revenir le réseau vers l'ordre zéro. Attention de ne pas cocher la case réglage dans ce cas.

Nota Bene : Les programmations enregistrées dans le fichier « ESP_SolEx.INO » que vous avez uploadé sur l'ESP32 sont valables sans modification pour la seule utilisation d'un réseau 2400tr/mm, en combinaison avec les roues dentées préconisées dans le système proposé :

- Poulie principale de 60-GT2, poulie moteur de 16-GT2 et courroie GT-2

Bon amusement !

Jean et Stéphane.

Matériel utilisé :

Partie mécanique :

PLA+ (Sunlu) pour les pièces à imprimer (lit à 80°C ; tête d'impression à 225°C) ;

5 Ruthex M3 & 2 Ruthex M4 ; rondelles pour M3 et M4

2 vis M4 50mm ; 4 vis M3 15mm et 2 vis M3 40mm + écrous

1 vis sans tête M3 (pour fixer l'axe dans le porte réseau

1 axe 5mm diamètre, longueur 40mm : <https://www.amazon.fr/dp/B07P45W61W>

2 roulements à billes 625ZZ : <https://www.amazon.fr/dp/B079DKGWS9>

1 moteur 28BYJ-48 5V (voir plus bas pour référence)

1 poulie 60 dents <https://www.amazon.fr/dp/B07N65GCL6>

1 poulie 16 dents <https://www.amazon.fr/DollaTek-aluminium-al%C3%A9sage-largeur-Imprimante/dp/B07DK3DCMS>

1 courroie 158 dents <https://www.amazon.fr/dp/B07D923SDS>

Partie électronique :

ESP32-WROOM-32 Dev kit C4: <https://www.amazon.fr/dp/B08BTS62L7>

ULN2003 & moteur 28BYJ-48 5V: https://www.amazon.fr/HUABAN-moteurs-28BYJ-48-module-ULN2003/dp/B08B4J33HX/ref=sr_1_6

Connecteur USB pour alimentation : <https://www.amazon.fr/dp/B0BQMWC5FL>

EEPROM : <https://www.amazon.fr/dp/B07RHMPRK1>

Module d'alimentation: <https://www.amazon.fr/DollaTek-d'alimentation-Conversion-Tension-Multi-Sortie/dp/B07DK4NTH3>

Carte de prototypage 7x9cm : <https://www.amazon.fr/HeyNana-prototype-perfor%C3%A9-circuit-Prototype/dp/B092455RD6>

Cable USB pour programmer l'ESP ;

Alimentation 12V ou USB en fonction du montage choisi.