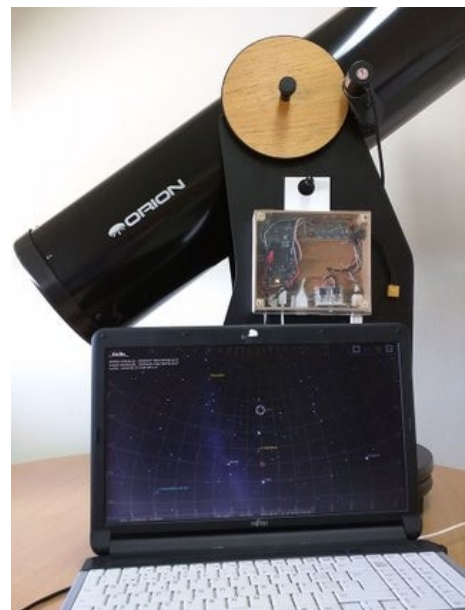
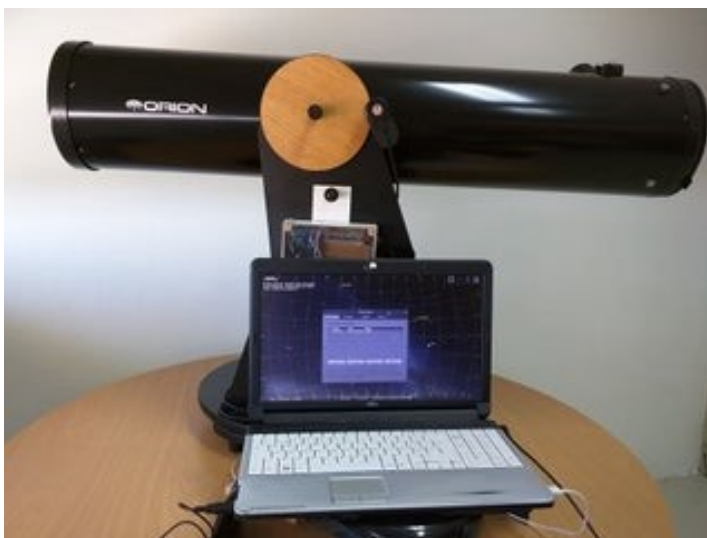
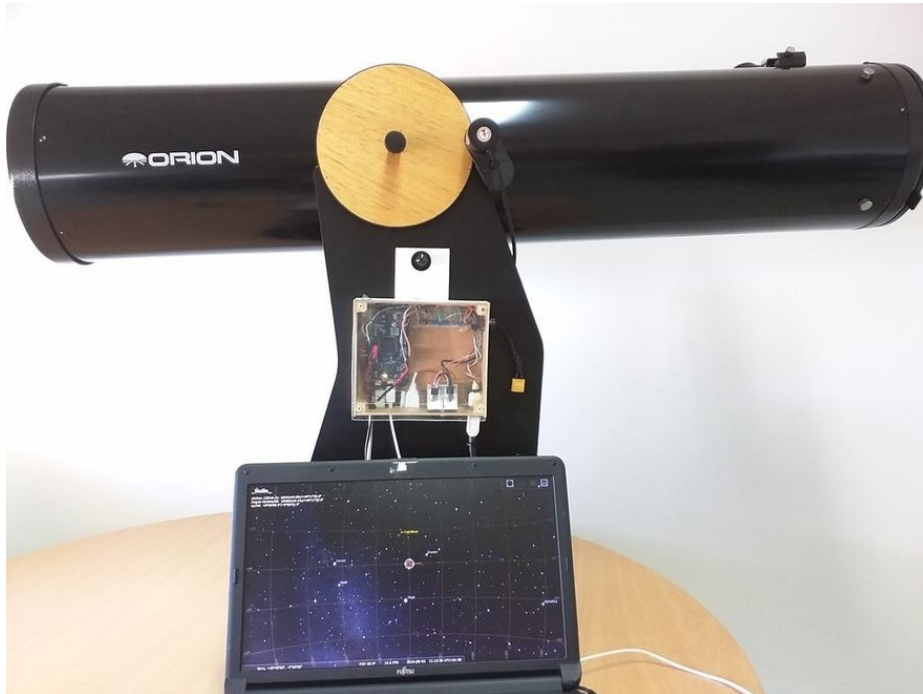


# Contrôlez votre télescope à l'aide de Stellarium & Arduino

Par [NelsonFerraz](#) dans [Workshop Science](#)

Page originale : <https://www.instructables.com/id/Control-Your-Telescope-Using-Stellarium-Arduino/>



L'astronomie me fascine depuis que je me suis tournée vers le ciel nocturne. Récemment, j'ai acheté un télescope [newtonien de 8 ''](#) livré avec une [monture dobsonienne](#). C'est une jolie monture (bon marché, portable et très facile à utiliser), mais pour obtenir ce que vous voulez voir, vous devez connaître le ciel comme une paume de vos mains (ce qui peut être un problème pour un débutant comme moi ..!).

Pour vous aider à connaître le ciel, il existe plusieurs logiciels que vous pouvez utiliser et [stellarium](#) est mon logiciel préféré. Stellarium est un projet open source avec de nombreuses fonctionnalités astronomiques utiles, y compris un plugin permettant la connexion entre un ordinateur et un contrôleur de télescope (comme celestron, meade ou sky-watcher).

Après quelques nuits passées à utiliser mon télescope, je me suis rendu compte que cela pourrait être utile si, à tout moment, je pouvais comparer la position de mon télescope avec la position de la cible que je cherchais.

Vous pouvez dire : oui ! Achetez un [télescope goto](#) et vous verrez votre problème résolu! Eh bien, vous avez absolument raison !

Cependant, à cause de mon passé électrique et parce que je me considère comme un amateur, ma première pensée a été : - Je peux (et je le ferai) construire un contrôleur de bricolage pour mon télescope.

Pourquoi construire un contrôleur plutôt que d'acheter un prêt à l'emploi ?

- Ce sera très amusant d'essayer de faire quelque chose de nouveau.
- C'est une bonne contribution à apprendre beaucoup de nouvelles choses (autrement évitées en utilisant une solution prête à l'emploi) comme : les coordonnées alt-azimut et célestes, le temps sidéral local, le mouvement de la terre, la trigonométrie sphérique ;
- Vous pouvez contrôler votre propre budget et le maintenir en dessous de 50 USD ;
- Votre télescope sera personnalisé pour répondre à vos besoins.
- Votre télescope deviendra "célèbre" parmi vos amis ;
- Si vous n'êtes pas intéressé à dépenser des milliers de dollars dans une monture de choix, cela vous aidera vraiment lorsque vous commencerez à diriger votre télescope vers le ciel ;

Cette instruction vous guidera, étape par étape, sur la manière de compléter votre propre contrôleur en utilisant Stellarium (logiciel d'astronomie) et Arduino (plate-forme de prototypage open source reposant sur un matériel et un logiciel facile à utiliser).

**Et pourquoi Arduino ?** Eh bien, il n'ya pas de réponse simple à cela ... Personnellement, je préfère Arduino pour plusieurs raisons, notamment :

- Ce n'est pas cher et facile à trouver ;
- C'est un projet open source ;
- Avoir une énorme communauté et des tonnes d'informations en ligne ;
- Il est facile de trouver des bibliothèques qui peuvent vous aider à utiliser beaucoup de choses différentes ;
- Vous pouvez facilement ajouter de nombreux boucliers et capteurs spécialement conçus pour Arduino, ou vous pouvez en adapter d'autres ;
- Pour la plupart des projets, c'est assez fiable ;
- Il est très polyvalent et flexible (ports d'E / S, communication, etc.) ;
- C'est convivial ;
- Utilise "C" comme langage de programmation ;
- Il est facile d'interagir avec les ports d'E / S, numériques et analogiques ;

Maintenant que vous êtes déjà familiarisé avec ce "petit" projet, voici un résumé de ce que vous devez faire :

- Trouvez un moyen de placer 2 capteurs de position (encodeurs) pour lire l'azimut et l'altitude ;
- Connectez-les à un contrôleur comme [Arduino](#) ;
- Construisez l'interface matérielle (électronique élémentaire) ;
- Programmez le contrôleur pour calculer les coordonnées et communiquer avec un ordinateur ;
- Installez et configurez Stellarium pour fonctionner avec votre contrôleur.

Cela vous coûtera un peu de temps et (beaucoup) de patience, mais vous serez satisfait du résultat. Regardez la vidéo ci-dessus et ...

**- Ça vaudra le coup !**

# Étape 1 : Outils et matériel

À mon avis, un projet de bricolage devrait être quelque chose que vous pouvez réaliser dans votre garage avec vos propres outils. Il devrait également avoir un petit budget, sinon, il serait préférable d'acheter ce que vous cherchez. De plus, j'aime toujours utiliser des matériaux recyclés ou quelque chose que je peux récupérer de la ferraille.

## Les outils dont vous avez besoin :

- Scie à main et / ou scie sauteuse
- Papier sablé, ponceuse, râpe
- Foret et un ensemble de foret
- Fer à souder
- Règle, étrier, compas
- Tournevis, pinces, cutter, ...
- Un ordinateur portable avec [stellarium](#) et [arduino](#) IDE (vous pouvez télécharger et installer les deux gratuitement)

## Matériaux :

- 2 encodeurs incrémentaux [d'ebay](#)
- 2 poulies GT2, [de ebay](#)
- 2 courroies de distribution GT2 avec 600mm, [de ebay](#)
- 2 roulements 5175 (trou de 5mm et 5mm d'épaisseur), mais vous pouvez en utiliser d'autres
- Arduino DUE et câble USB (UNO a beaucoup de limitations pour ce projet ..)
- Un petit proto PCB pour souder les composants (à la place, vous pouvez travailler avec une planche à pain)
- Quelques vis, boulons, écrous, rondelles
- 2 ressorts (que vous pouvez enlever d'un stylo ...) ou 2 élastiques
- Plaque de métal (j'utilise une plaque d'aluminium de 2 mm que j'ai prise dans un ancien meuble, mais l'acier devrait convenir)
- Contreplaqué (5 mm et 3 mm d'épaisseur) et quelques morceaux de bois (12x12mm)
- Fils et connecteurs en ferraille (anciens câbles série ou VGA, câbles USB, ...)
- Soudure, tube thermo rétractable, colliers de serrage, ruban adhésif, etc.
- Quelques connecteurs dupont pour Arduino (optionnel)
- Peinture noire, colle à bois, colle chaude
- 4 trimmers (10 kOhms)
- 4 résistances (470 Ohm)

## Étape 2 : Mécanique, Partie 1 - Fabriquer des engrenages à partir d'une courroie de distribution



C'est une étape simple qui vous guidera dans la construction d'une roue dentée à partir d'une courroie de distribution.

Parce que nous avons besoin de lire 2 axes (altitude et azimut), nous devons construire 2 roues. Ces roues dentées seront attachées et fixées au support et seront responsables de l'utilisation des encodeurs lorsque vous tournez votre télescope (azimut) et de haut en bas (altitude).

Pourquoi construire un engrenage? Premièrement, je pensais pouvoir connecter directement les encodeurs au télescope. Ensuite, j'ai compris qu'il serait préférable d'utiliser un pignon fixé au support et un télescope pour piloter les encodeurs. Cet engrenage permettra une connexion mécanique plus douce (pour éviter les charges mécaniques sur le codeur) et, ce qui est plus important, permettra plus de résolution du codeur - en multipliant les tours du codeur par un rapport de réduction. Cette configuration (courroie de distribution à 300 dents et poulie de codeur à 20 dents) donne un rapport de transmission de 15 (rapport =  $300/20 = 15$ ). Cela signifie qu'un tour complet du télescope donnera 15 tours complets de l'encodeur, de sorte que vous multipliez la résolution de l'encodeur par 15!

## Faire la roue:

Commencez à dessiner un cercle dans un morceau de contreplaqué de 3 mm. Utilisez un compas et n'oubliez pas de faire une bonne marque au centre du cercle - vous en aurez besoin!

Comme la courroie de distribution GT2 a un périmètre de 600 mm, vous devez dessiner un cercle de rayon 95,5 mm (ou 191 mm de diamètre). Si votre courroie de distribution a un périmètre différent, vous pouvez trouver le diamètre du cercle (d) en faisant:  $d = P / \pi$ . Rayon (r) est  $r = d / 2$ .

J'ai travaillé avec un puzzle pour découper le cercle, puis je l'ai poncé avec une ponceuse, comme le montre la photo ci-dessus. Pour obtenir le meilleur résultat, commencez à couper un cercle légèrement plus grand, puis poncez-le jusqu'à ce que la courroie de distribution puisse entrer. Pour éviter de coller la ceinture, je la laisse vraiment serrée autour du contreplaqué.

Essayez de rendre le cercle aussi parfait que possible!

Après avoir installé la courroie de distribution (la mienne a une largeur de 6 mm), coupez l'excédent de caoutchouc pour obtenir une épaisseur équivalente à celle du contreplaqué (3 mm).

Pour plus de stabilité, j'ai découpé et collé un cercle plus petit (environ 20 mm de rayon) (voir les images ci-dessus) en contreplaqué de 5 mm.

Percer un trou central dans les deux roues. Ces trous doivent avoir le même diamètre que les boulons existants. J'ai donc foré un trou de 6 mm à partir de l'altitude et de 10 mm à partir de la roue d'azimut.

Retirez la courroie de distribution et terminez votre travail avec une couche de peinture noire autour du contreplaqué. Une fois la peinture sèche, réinstallez la courroie de distribution et le tour est joué!



## Étape 3: Mécanique, Partie 2 - Installer les roues



Installez la roue d'altitude directement sur le télescope, en utilisant le boulon existant. Il devrait être facile et rapide à installer et à désinstaller (transport et emballage faciles).

De la même manière, utilisez le boulon existant au bas de la monture et installez-y la roue d'azimut. Ajoutez une rondelle et un écrou (M10) pour serrer la roue. J'ai utilisé un écrou à oreilles à la place, car il deviendra utile lorsque vous aurez besoin de démonter ou d'ajuster la roue.

**Il s'agit d'une étape importante:** assurez-vous que vos roues sont suffisamment serrées pour éviter tout glissement lors de l'opération!

À ce moment, vous pouvez vérifier si les roues sont centrées et droites en faisant tourner le télescope et la monture.

Bon travail!

## Étape 4: Mécanique, partie 3 - Fabrication du support de codeur azimutal



Maintenant que les roues sont installées, nous devons trouver un moyen de placer et d'utiliser les encodeurs.

N'hésitez pas à utiliser le matériel que vous voulez (ou que vous avez déjà) et les tailles qui correspondent le mieux à votre télescope et à votre monture. Au lieu de construire un bloc d'oreiller pour tenir le roulement, vous pouvez acheter un roulement de bloc d'oreiller bon marché d' [ebay](https://www.ebay.com) . Cela devrait être plus rapide et plus facile, mais comme je disposais de 2 roulements sur le plateau, je les ai utilisés à bon escient!

**Pour être un peu plus organisé, j'ai séparé cette étape en deux parties: celle-ci expliquant comment construire le support de codeur Azimut et la suivante (très similaire) permettant d'afficher le support Altitude.**

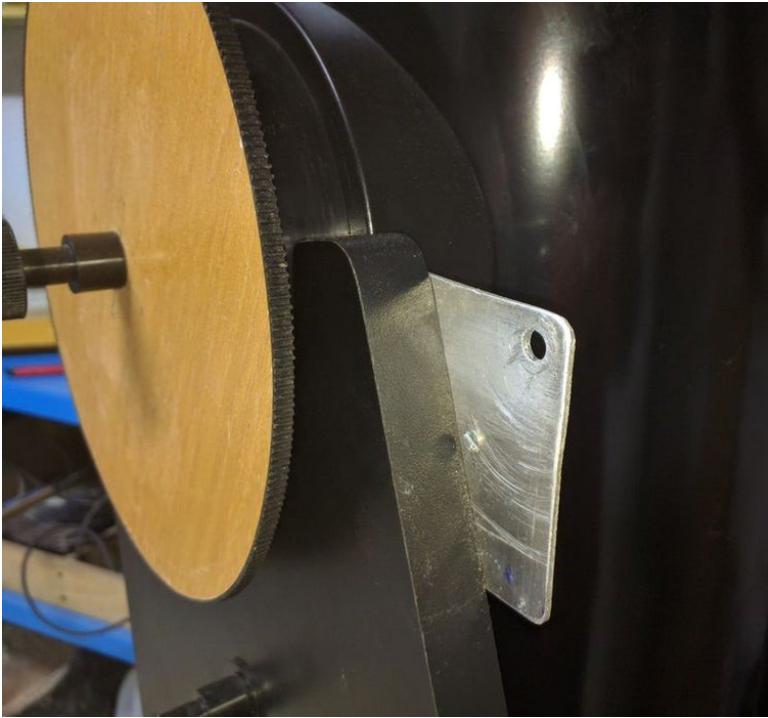
### **Construire les supports de l'encodeur:**

Comme vous pouvez l'imaginer, il n'est pas facile d'obtenir un assemblage mécanique parfait avec des outils fabriqués à la main. Donc, j'ai perdu du temps supplémentaire à faire des supports flexibles (en utilisant des roulements et des ressorts). De cette façon, je pense que je peux éviter les charges indésirables sur les codeurs (qui peuvent causer des dommages) en adoucissant le fonctionnement mécanique.

Ce faisant, j'ai également amélioré la portabilité de tout: il devrait être plus facile de placer et de retirer les roues dentées en tirant le codeur vers l'arrière. Il sera également plus facile d'effectuer certains ajustements mécaniques futurs.

Au lieu d'une énorme description de cette étape (et de la suivante), je vais vous montrer comment faire en **utilisant une vidéo et quelques images (lisez attentivement tous les commentaires).**

## Étape 5: Mécanique, partie 4 - Fabrication du support de codeur Altitude



Maintenant que nous avons déjà le détendeur de l'Azimut, construisons, en **suivant les mêmes principes** , le détendeur de l'Altitude.

Encore une fois, comme une image parle pour mille mots, j'ai décrit cette étape en utilisant les [images et la vidéo ci-dessus](#) (et leurs commentaires).

Si vous avez accompli l'étape précédente sans problème majeur, cette étape devrait être un "morceau de gâteau"!



## Étape 6: Mécanique, Partie 5 - Tout concocter



### Installez les encodeurs et testez-les:

Vissez les supports sur la monture du télescope, puis installez les encodeurs. Encore une fois, j'aime bien utiliser des écrous à oreilles (voir les images).

**NOTE:** À propos de la façon dont j'ai conçu, construit et attaché les encodeurs et les roues:

Vous avez peut-être remarqué que j'avais pris soin de ne pas endommager le télescope et le support, car j'ai trouvé un moyen d'installer toutes les pièces mécaniques sans percer, coller ou faire quelque chose qui pourrait endommager ou perdre de la valeur du télescope. Ainsi, vous pouvez placer et retirer les roues et les encodeurs de votre télescope et, à la fin, votre télescope sera exactement le même que celui que vous avez acheté (sauf un très petit et imperceptible 4 trous ...).

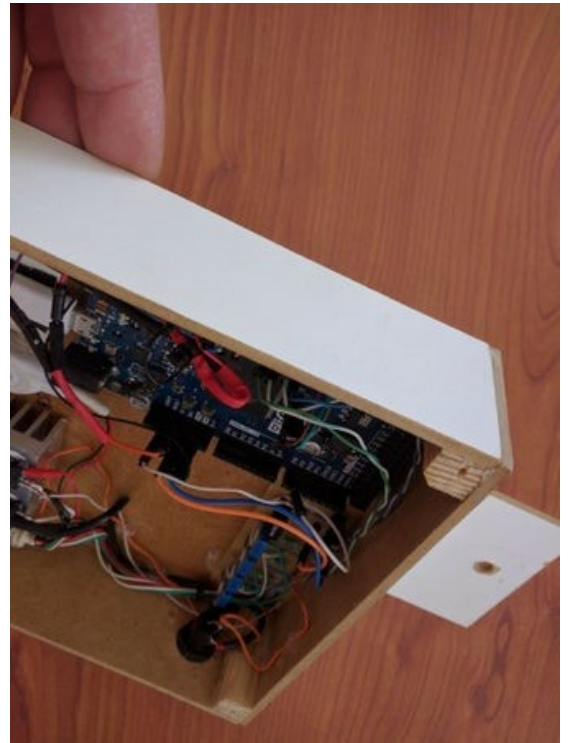
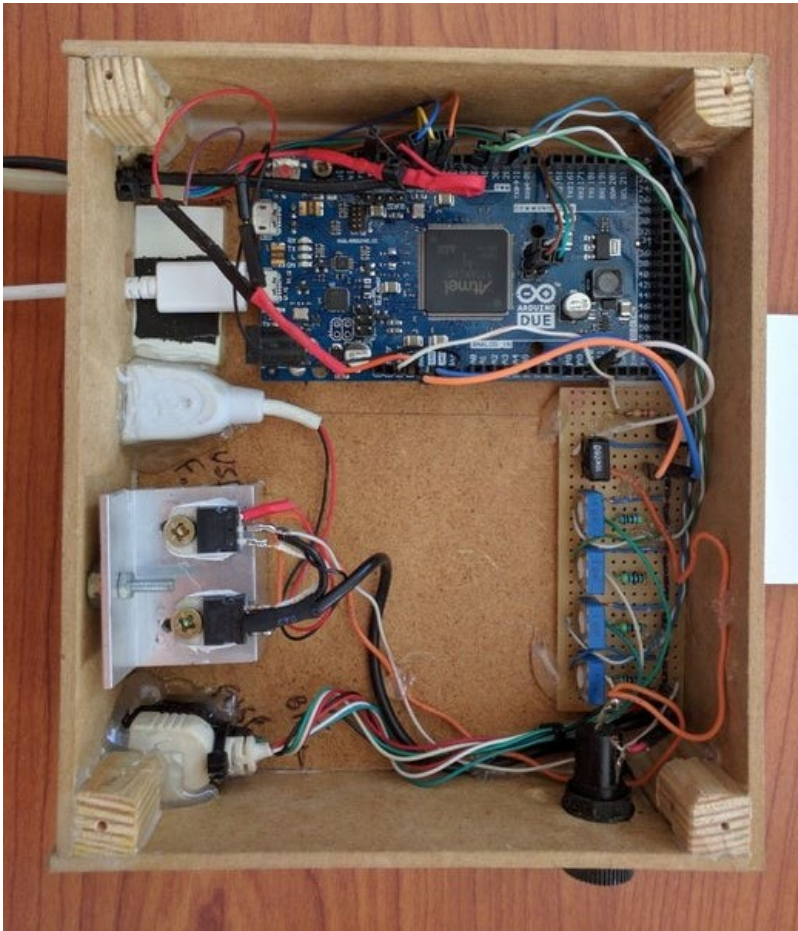
À ce stade, vous devriez vérifier à nouveau si les roues sont centrées et droites, en tournant le télescope de haut en bas. Assurez-vous que le fonctionnement est lisse et, surtout, qu'il n'y a pas de glissement entre les roues et les poulies de l'encodeur.

La **vidéo ci - dessus** vous montre exactement ce que j'ai fait jusqu'à présent.

Vous pouvez confirmer que, si vous tournez le télescope d'un tour complet, le codeur effectuera une rotation exacte de 15 fois.

Jusqu'ici, si tout s'est bien passé, **vous pouvez vous reposer un moment !!**

## Étape 7: Construire une boîte pour placer l'électronique



Il est souhaitable, si vous construisez une boîte, de placer tous les composants, y compris l'Arduino.

La boîte que j'ai fabriquée ne gagnera jamais un concours de beauté, mais elle est assez grande pour contenir tout ce dont j'ai besoin, elle est légère et portable, elle est le plus important, elle tient bien et protège les composants électroniques.

J'ai utilisé 5 morceaux de contreplaqué de 3 mm collés ensemble pour former une boîte de 16x14x5 (cm), comme vous pouvez le vérifier à partir des images ci-dessus.

Pour renforcer la boîte, j'ai collé des morceaux de bois aux coins. Comme vous pouvez le constater, la couverture est fabriquée à partir d'un morceau d'acrylique. Ce n'est pas obligatoire d'être transparent, mais ça devrait être amovible. Il suffit de percer un trou à chaque coin pour le visser.

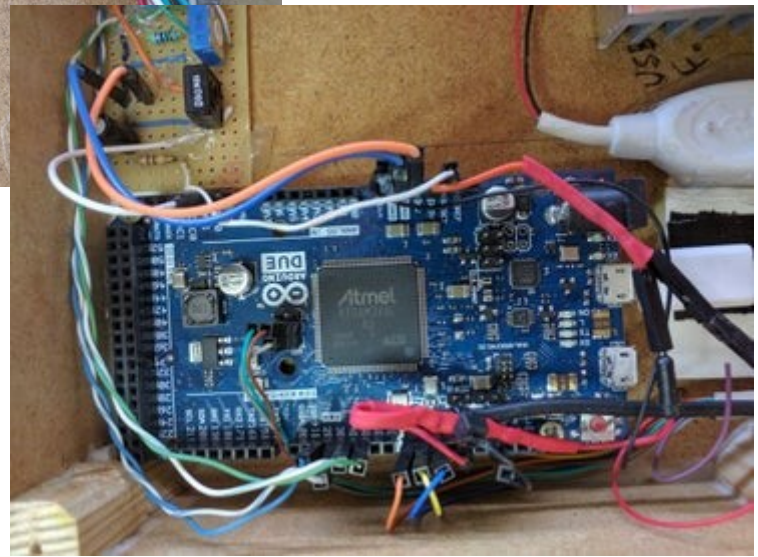
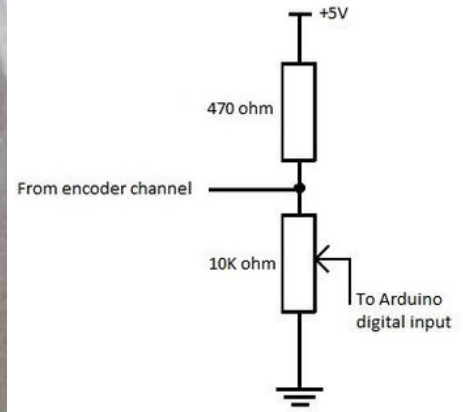
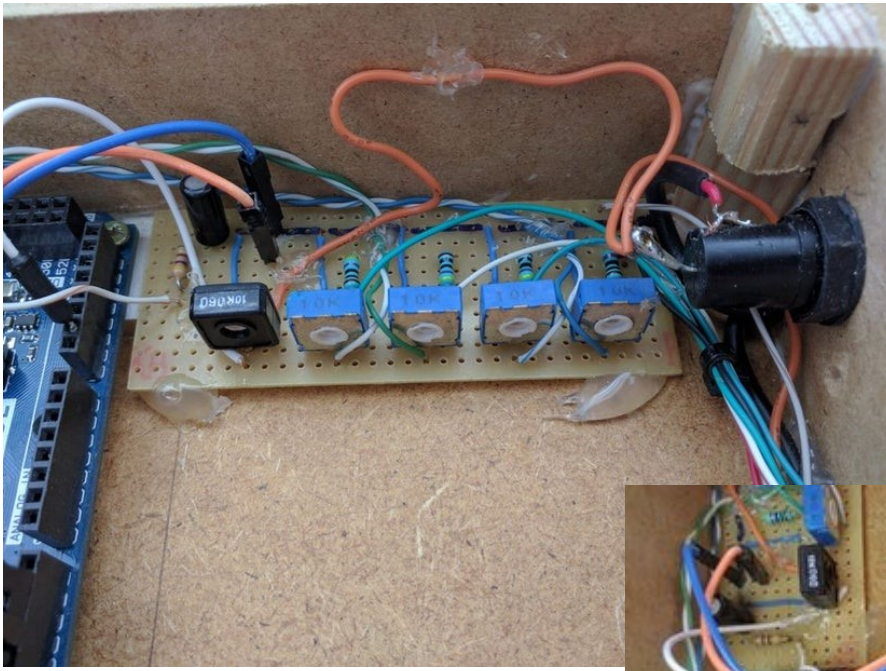
Percer des trous pour passer les câbles.

Vous pouvez ajouter un crochet pour accrocher la boîte à un endroit pratique ou, comme dans la dernière image ci-dessus, coller un morceau de contreplaqué avec un trou que vous pouvez utiliser pour accrocher la boîte à l'aide d'un boulon existant du support.

**REMARQUE: étant** donné que j'utilise également ce boîtier à de nombreuses autres fins, des câbles supplémentaires circulent. S'il vous plaît les ignorer!



## Étape 8: Électronique - Créez l'interface et connectez-la



Après avoir construit la boîte, placez simplement l'Arduino et fixez-le à l'aide de colle chaude ou même de petites vis.

Connectez le câble USB (j'utilise le port de programmation sur Arduino).

### Construire l'interface:

Nous devons construire l'interface matérielle pour 2 raisons:

1. Adapter le niveau de tension du codeur (tension minimale = 5V) à Arduino DUE (3,3V);
2. Ajoutez une résistance de rappel (nécessaire car la sortie des encodeurs est un collecteur ouvert).

Ci-dessus, vous trouverez les schémas que j'ai utilisés pour le canal de chaque encodeur. Vous aurez besoin de construire 4 comme ceux-ci.

C'est une interface très simple, où une résistance de 470 ohms est utilisée pour tirer la sortie du codeur et un trimmer de 10K est utilisé comme diviseur de tension. La broche centrale du potentiomètre est ensuite connectée aux broches d'entrée de l'Arduino.

J'ai utilisé la configuration suivante:

- codeur 1, voie A - connecté à la broche 2
- codeur 1, voie B - connecté à la broche 3
- codeur 2, canal A - connecté à la broche 5
- codeur 2, canal B - connecté à la broche 4

**AVERTISSEMENT:** Avant de connecter les encodeurs à Arduino, assurez-vous de régler les potentiomètres de manière à obtenir une sortie de 3,3V exactement. Il est très important que vous ne dépassiez pas 3.3V, sinon vous endommageriez l'Arduino.

Pour le faire en toute sécurité, connectez le fil rouge des encodeurs à la broche Arduino + 5V et le fil noir à la terre. Ajustez tous les trimmers jusqu'à lire 3.3V sur la goupille du milieu Vous devrez peut-être tourner légèrement le codeur pour trouver la position "ON" sur chaque canal.

Pour devenir plus portable, au lieu de souder directement les fils du codeur au circuit imprimé, j'ai ajouté des connecteurs USB que j'ai pris à partir de 2 câbles USB. Souder les extrémités femelles à votre PCB et les extrémités mâles à vos encodeurs.

En agissant de la sorte, assurez-vous de ne pas échanger de fil - le fait de respecter les couleurs vous aiderait beaucoup!

Comme je l'ai déjà dit, j'utilise cet Arduino à d'autres fins, alors ignorez tous les fils et composants supplémentaires que vous voyez sur les images ci-dessus - vous n'en aurez pas besoin!

## Étape 9: Étapes importantes avant de créer le code

Pour devenir un peu plus "instructif", je sens que je vous dois au moins une explication simple sur la façon dont j'ai programmé l'Arduino et sur les techniques que j'avais l'habitude de faire (presque) n'importe quel télescope "parler" avec Stellarium.

En dépit des mathématiques, de la trigonométrie, de la programmation et du traitement du signal, ne vous inquiétez pas, je vais essayer de ne pas être trop volumineux!

### Connaître les encodeurs:

De manière très simple, un [codeur](#) ([codeur d'](#) angle ou codeur de position) est un dispositif (capteur) capable de lire et, par conséquent, de transmettre une position ou un angle.

Il y a beaucoup de [types de codeurs](#) . Pour ce but spécifique, j'ai choisi les [incrémentaux](#) . Bien qu'ils soient difficiles à intégrer à Arduino (programmés en C), ces encodeurs sont fiables, bon marché (du moins ceux que j'ai achetés ..!), Faciles à trouver et, le plus important, ils peuvent être très précis tout en ayant résolution.

Le problème majeur avec ce type de codeurs est qu'ils ne vous donnent jamais leur position (angle)... mais vous donnent des impulsions... étrange? Alors, comment pouvons-nous connaître l'angle? Voici une très petite explication:

1. Il y a deux canaux à chaque encodeur (canaux A et B);
2. Chaque canal peut basculer entre ON et OFF (0 ou 1) lorsque vous tournez l'encodeur. C'est un pouls!
3. Les 2 canaux ne sont pas en phase. Il y a un décalage de 90° entre eux. Cela s'appelle un [signal en quadrature](#) ;
4. En utilisant des signaux en quadrature, nous pouvons connaître la direction du mouvement (si un canal est en avance ou en retard par l'autre);
5. Il est également possible de multiplier la résolution par 2 ou par 4 (propriétés en quadrature);
6. Nous pouvons calculer la position en comptant le nombre d'impulsions (CW et CCW);

La technique: Pour éviter de perdre des impulsions (car le code prend du temps à s'exécuter ...), j'ai utilisé des broches et des fonctions spéciales sur Arduino: les [interruptions](#). Ainsi, chaque fois que l'une de ces broches change d'état (c'est-à-dire que le codeur a été déplacé), cette broche spéciale (via l'interruption) active un morceau de code spécial (ISP - Interrupt Service Routine). [Ici](#) vous pouvez lire des informations supplémentaires sur l'utilisation des interruptions sur Arduino.

Cette technique permet de définir une section de code "sensible au temps", et c'est ce que j'ai fait pour compter toutes les impulsions, sans exception.

### Communication avec Stellarium

Rien de trop difficile ici!

Le plugin Stellarium fonctionne parfaitement avec le protocole LX200. Je pense que c'est un protocole ancien et obsolète, mais il est simple à utiliser et, à cette fin, fonctionne parfaitement.

Lorsqu'il est connecté à Arduino, Stellarium envoie périodiquement deux chaînes: ": GR #" - prêt à recevoir le RA et ": GD #" - à recevoir le DEC. Le code vérifie ces chaînes et envoie ensuite au port série la chaîne correspondante avec l'ascension droite actuelle (RA) et la déclinaison (DEC).



## Conversion du système de coordonnées

Maintenant, la partie compliquée ... Chaque montage dobsonien (ou [altazimut](#) ) utilise deux axes (altitude et azimut, comme vous le savez déjà!) Pour créer un système de coordonnées. Ce système est très intuitif et facile à utiliser car il est lié à votre vie quotidienne (monter / descendre et droite / gauche ...).

Mais, ce système de coordonnées est très pauvre si vous parlez d'astronomie. Cela arrive que le ciel est "vivant" et semble se déplacer continuellement. En fait, toutes les étoiles décrivent une circonférence centrée sur l'étoile polaire. Ils tournent à 360° en 24H environ (15° par heure).

Comme vous pouvez l'imaginer, si, à un moment donné, vous pointez un objet à l'aide d'une coordonnée donnée, vous devez recalibrer après quelques minutes vos coordonnées car l'objet a été déplacé. Les coordonnées altazimuth dépendent donc du temps. Ils dépendent également de l'emplacement, comme vous pouvez l'imaginer!

Ainsi, pour éviter d'utiliser un système de coordonnées qui change chaque seconde, les astronomes ont trouvé un moyen très ingénieux de déterminer les coordonnées du ciel: le [système de coordonnées équatorial](#) .

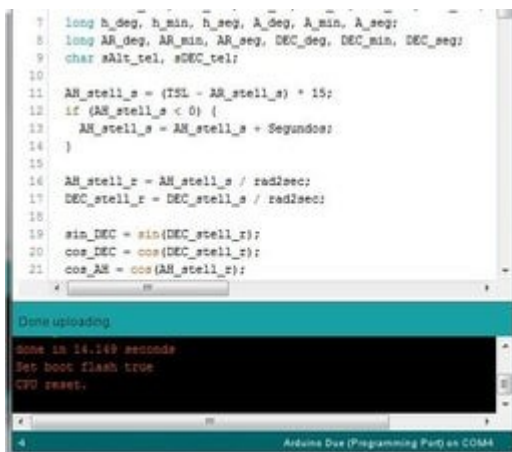
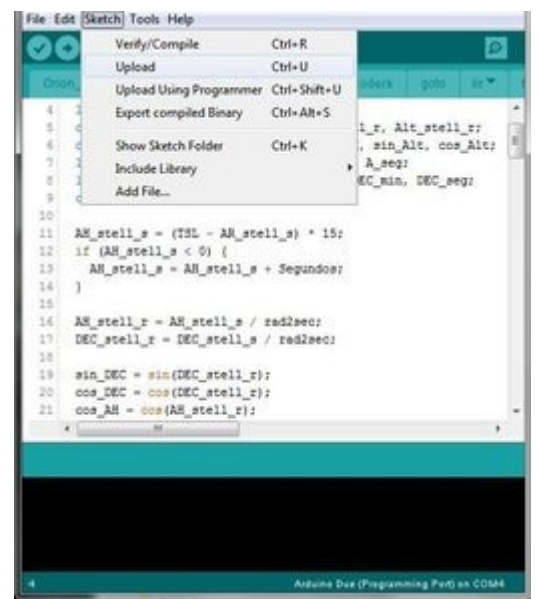
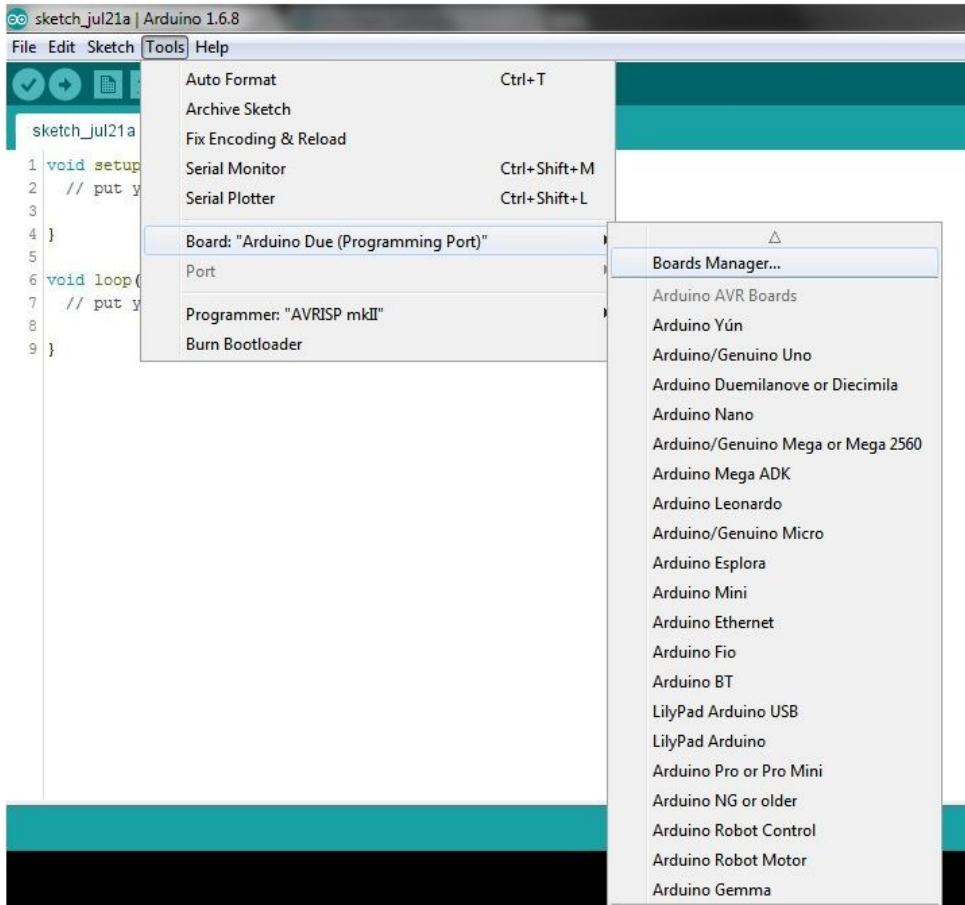
Parce que Stellarium, grâce au protocole LX200, ne pouvait lire que les coordonnées équatoriales, je devais trouver un moyen de convertir les lectures du télescope (altazimuth) en système de coordonnées équatoriales. Cette opération est un peu complexe car elle implique une trigonométrie sphérique et des concepts de temps (heure sidérale locale - LST).

Tout est fait dans cette conversion, une fois dans un cycle de calcul, en utilisant la fonction "AZ\_to EQ ()".

Pour en savoir plus sur les systèmes de coordonnées, l'astronomie de position ou le LST, vous pouvez aller [ici](#) , [ici](#) et [ici](#) ...

**Bienvenue dans mon monde!**

# Étape 10: Installez Arduino et chargez le code!



J'espère que vous connaissez déjà Arduino, mais si ce n'est pas le cas, ne vous inquiétez pas, vous pouvez obtenir beaucoup d'informations [en ligne](#) . Néanmoins, je vais vous expliquer comment recommencer à zéro!

Il est assez facile de charger le code dans votre Arduino. Si vous n'avez pas l'IDE Arduino, vous pouvez le télécharger librement à partir d' [ici](#) . J'utilise la version 1.6.8, mais d'autres devraient également fonctionner (juste au cas où, essayez de ne pas utiliser certaines versions plus anciennes). Il suffit de télécharger et d'installer sur votre ordinateur.

Assurez-vous d'avoir installé les pilotes DUE (cartes SAM, 32 bits) - vérifiez les images ci-dessus.

Connectez le câble USB à votre ordinateur et ouvrez le code (voir les pièces jointes). Assurez-vous que les fichiers de code (NF\_v1.ino et config.h) se trouvent dans un dossier du même nom ("NF\_v1").



Sélectionnez, sous l'onglet "Outils", la carte (Arduino DUE - Port de programmation) et le port que vous utilisez. Notez que le port (le port 4 dans mon cas) peut varier d'un ordinateur à l'autre. Ce sera le port série que votre ordinateur associe à Arduino.

Une fois que tout est configuré, cliquez sur pour télécharger le code (Sketch> Upload).

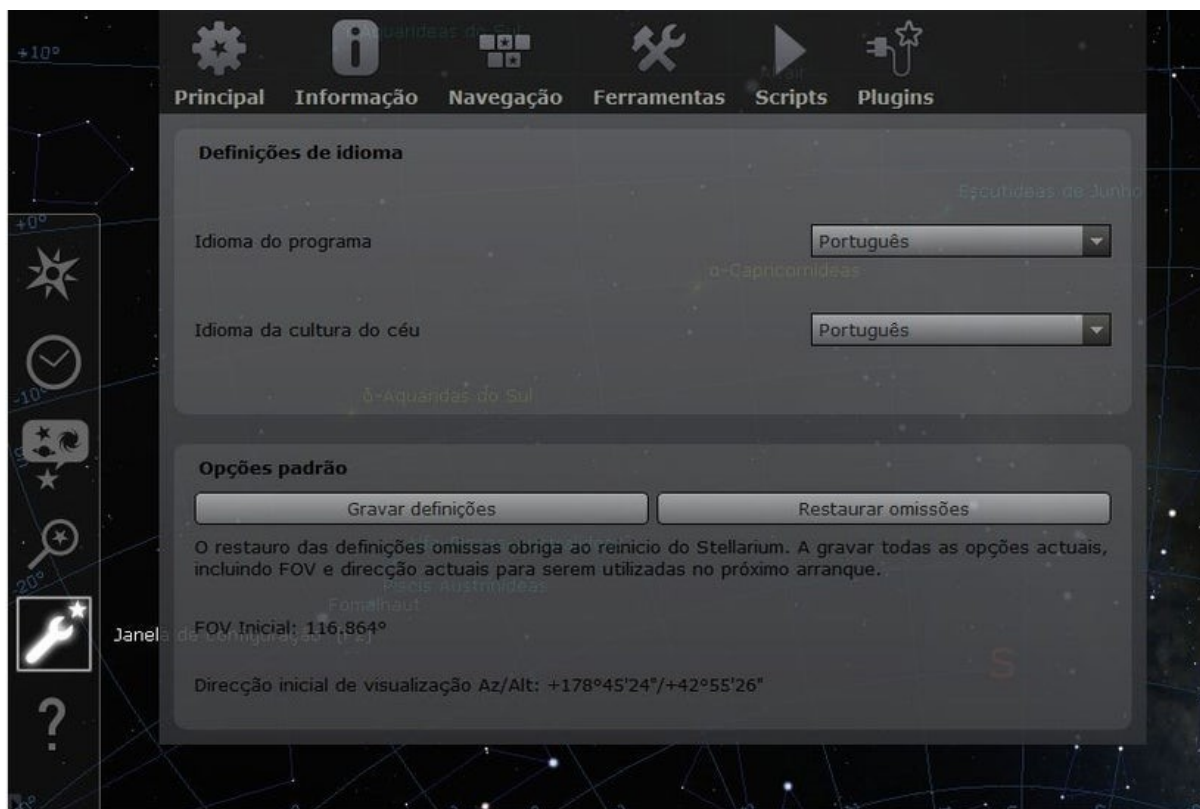
Un message "Done uploading" apparaîtra au bout de quelques instants au bas de l'écran. Cela signifie que votre code est compilé et téléchargé dans votre Arduino.

Après cela, vous êtes prêt à partir!

### Les pièces jointes

-  [config.h](#)       [NF\\_v1.ino](#)

# Étape 11: Installez et configurez Stellarium



## **Tu es presque là!**

Si Stellarium n'est pas installé sur votre ordinateur, il est temps de le faire!

Vous pouvez le télécharger gratuitement à partir de la [page d'accueil Stellarium](#) .

Stellarium est un très bon logiciel pour l'astronomie. Il offre un environnement convivial et, si vous choisissez une version récente (version 0.13.1 ou), devrait déjà intégrer le plugin Télescope, il ne vous reste plus qu'à l'activer!

Allez dans la fenêtre de configuration et choisissez l'onglet "plugins". Cochez la case pour activer le plugin. La prochaine fois que vous ouvrirez Stellarium, vous devriez avoir un nouvel outil dans la barre inférieure (voir les images ci-dessus).

Cliquez sur "l'outil télescope" puis sélectionnez "configuration du télescope".

Cela ouvrira une nouvelle fenêtre. À cette fenêtre, choisissez "Ajouter un télescope".

Pour configurer votre télescope, suivez les images ci-dessus. Vous n'avez besoin que de ça une fois!

Une chose importante à faire: configurez votre emplacement, sinon toutes les coordonnées seront fausses.

Il vous suffit donc de cliquer sur, à gauche de votre écran, le gestionnaire de localisation et d'ajouter votre localisation (je veux dire, l'emplacement de votre site d'observation ...!).

Entre autres choses, vous pouvez personnaliser certaines fonctionnalités de visualisation. Faites un tour et apprenez à vous en servir!

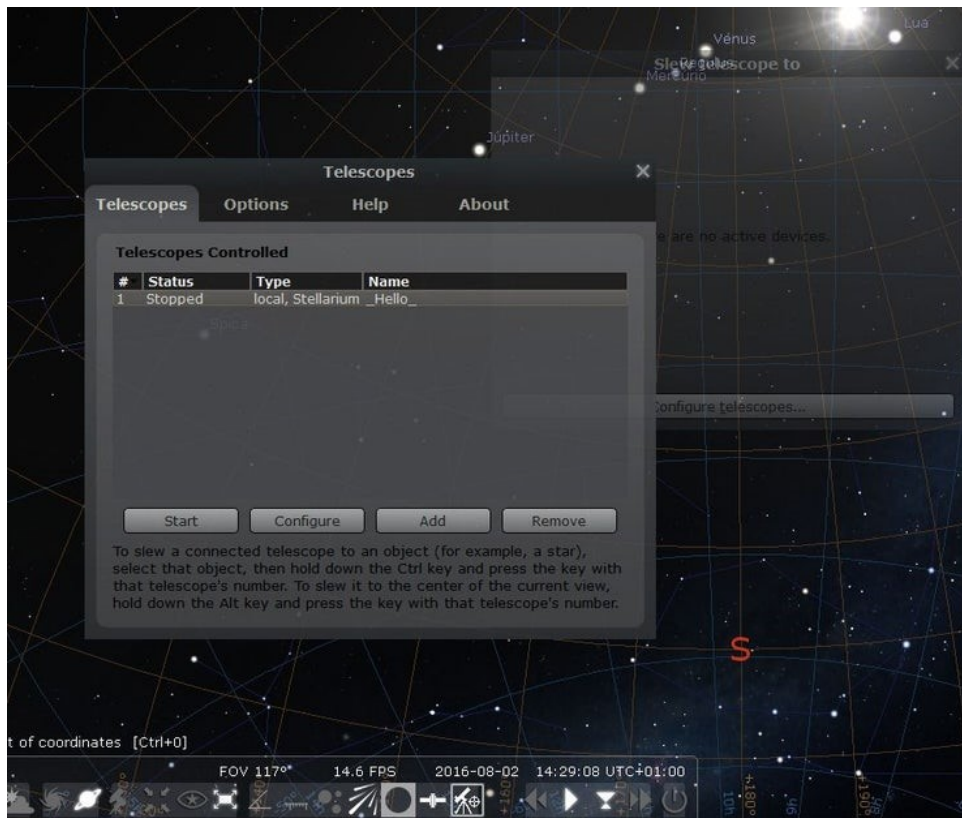
Pour cela, il est souhaitable d'activer certaines fonctionnalités dans les onglets "Voir" et "Configuration", comme vous pouvez le voir sur les images ci-dessus.

**Enfin, vous êtes prêt à connecter votre télescope à Stellarium!**

Passez à l'étape suivante pour apprendre à le faire!



# Étape 12: Testez et courez!



```

1
2 //.....
3 // Config your sensor pins and location here!
4 //.....
5
6
7 // Define the number of pulses that your encoder (1 and 2) gives by turn, and multiply by 4
8 // In my case: 400 x 15 = 4 (400 pulses by turn) 15 is the gear ratio, so:
9 long pulses_enc1 = 36000;
10 long pulses_enc2 = 36000;
11
12 // Define DOE's pins
13 #define enc_1A 2 // define DOE pin to encoder 1-channel A
14 #define enc_1B 3 // define DOE pin to encoder 1-channel B
15 #define enc_2A 4 // define DOE pin to encoder 2-channel A
16 #define enc_2B 5 // define DOE pin to encoder 2-channel B
17
18 // enter your latitude (example: North 45°19'20")
19 int latNS = 40; // this means 40° North
20 int latNE = 30;
21 int latSE = 30;
22
23 // enter Pole Star right ascension (RA: HH:MM:SS)
24 int poleAR_RA = 2; // this means 2 hours, 52 minutes and 16 seconds
25 int poleAR_MM = 52;
26 int poleAR_SS = 16;
27
28 // enter Pole Star hour angle (H: HH:MM:SS)
29 int poleH_RA = 14;
30 int poleH_MM = 37;
31 int poleH_SS = 1;
32

```

## **Vous n'êtes plus qu'à quelques pas de votre récompense!**

Assurez-vous que tout est en place (électrique et mécanique) et utilisez votre télescope en ouvrant le plug-in du télescope. Cliquez sur "Configurer le télescope". Une nouvelle fenêtre apparaîtra et la seule chose que vous devez faire est de cliquer sur "Démarrer" et ... voilà!

Attendez quelques secondes (environ 10 secondes) et un nouvel objet (Bonjour) apparaîtra quelque part dans le ciel du Stellarium. Il aura le même nom que vous avez donné auparavant ...! Déplacez-vous et vous le trouverez!

### **Effectuer le test de mouvement:**

Déplacez votre télescope de haut en bas. Vous devriez voir l'objet bouger en conséquence chez Stellarium.

### **Est-ce que tu le vois? BON TRAVAIL!**

Dépannage: Si votre objet a tendance à se déplacer dans la mauvaise direction (azimut ou altitude), cela signifie que les canaux de votre codeur (respectivement azimut ou altitude) sont inversés. Corrigez-les en changeant les fils correspondants sur Arduino ou, tout simplement, en changeant le code et chargez-le à nouveau (voir image). Veillez noter que le codeur 1 est le codeur d'altitude et le codeur 2 est le codeur d'azimut. Utilisez le fichier "config.h" pour apporter vos modifications.

### **Configurez votre nombre d'encodeurs**

Arduino doit avoir configuré le nombre d'impulsions que vos encodeurs donnent tour à tour. Donc, en utilisant le fichier "config.h", définissez le nombre d'impulsions que vous avez par tour, puis multipliez par 4. Dans mon cas, j'ai un codeur de 600 impulsions et un rapport de transmission mécanique de 15 fois. Cela signifie que je vais avoir 9 000 impulsions par tour. En multipliant ce nombre par 4, le nombre que j'ai écrit était 36000, comme le montre l'exemple ci-dessus. Ce sera la **résolution globale**, équivalente à un incroyable **arc de 36 secondes** (1296000/36000). Comparativement, une pleine lune a environ 30 minutes d'arc de large (0,5°).

### **Alignez votre télescope:**

Pour conserver une précision maximale, le support du télescope doit être parfaitement mis à niveau (utilisez un niveau à bulle).

Par défaut, le télescope doit être aligné au nord (azimut 0°) et nivelé horizontalement (0° altitude) avant de se connecter à Stellarium. Vous pouvez le faire facilement en dirigeant le télescope vers l'étoile polaire (0° nord environ) puis en descendant jusqu'à ce qu'il soit horizontal (utilisez un niveau à bulle ...!).

### **Configurer le lieu et l'heure:**

Maintenant, si vous connectez votre télescope à Stellarium, vous verrez probablement que votre télescope n'est pas dirigé vers le nord comme il se doit. Ne vous inquiétez pas, votre position et votre heure sidérale locale (LST) ne sont pas les bonnes!

Si vous vous en souvenez, nous devrions avoir notre heure sidérale locale (LST) et notre position étalonnée. Pour cela, il suffit d'ouvrir le code avec Arduino IDE et d'éditer (avec l'onglet "config.h") votre position actuelle et l'ascension droite (AR), ainsi que l'angle horaire (H) de l'étoile polaire (pour le moment). Vous vous connectez à Stellarium). Il vous suffit de configurer votre latitude. La longitude sera calculée par AR et H.

Pour configurer votre lieu et votre heure, suivez les étapes décrites dans les images ci-dessus.

Rechargez le code et reconnectez-vous à Stellarium. Le code fera le reste du travail!

**UNE NOTE FINALE:**

**Jamais, mais ne pointez jamais votre télescope au soleil !** Cela peut provoquer une cécité irréversible.

Enfin, vous pouvez diriger votre télescope vers ce que vous voulez, en utilisant Stellarium pour vous guider!

**J'espère que vous l'appréciez autant que moi!**