

Contrôle d'un télescope amateur par une tablette via le wifi : Un exemple d'implémentation "presque" open source.

Yves A. Delhaye

29 décembre 2015

Résumé

Je décris ici comment contrôler un télescope amateur de type "GOTO" via le wifi. Des solutions complètement commerciales existent. Il existe cependant des solutions open source qui, bien qu'elles demandent plus "d'huile de coude", sont aussi plus ambitieuses. La solution décrite ici n'est pas complètement "open source" mais elle limite beaucoup les coûts et ouvre quelques perspectives intéressantes.

Cette version est une mise à jour de décembre 2015 par rapport à une réalisation qui datait de novembre 2013.

Introduction

Si vous ne comprenez pas les parties techniques, j'en suis désolé mais je n'ai pas de solution ! Sans doute pouvez-vous passer ce qui vous semble trop abscons!?

La documentation du projet d'origine est un peu incomplète et parfois il y a même des erreurs! Je décris ici comment j'ai fait et ce qui marche pour moi. Comme souvent, ceci est un "proof of concept" qui me satisfait. Il y a énormément à faire pour que ceci devienne une solution "clé sur porte".

Par choix raisonné, je travaille exclusivement sous Linux/Unix et la solution décrite ici est implémentée sous Debian/Linux.

J'assume mon statut de "physicien-geek" !

1 Le Pourquoi

Lorsqu'on organise une séance d'observation avec des amis non astronomes, la patience de ceux-ci peut être mise à épreuve. Lorsqu'on a prévu une planification, des nuages peuvent venir troubler le meilleur plan. Il faut improviser. Lorsque, comme moi, on débute dans l'astronomie amateur, on peut se retrouver dépourvu : "Y a-t-il quelque chose de bien à observer dans ce dernier coin de ciel dégagé?"

J'ai donc cherché une réponse relativement facile à mettre en œuvre.

2 Le but

Le but était d'avoir une solution transportable limitant au maximum le nombre de câbles et disposant d'une interface la plus visuelle possible.

J'ai testé des solutions "PC" mais celles-ci manquent un peu de souplesse : câbles dans tous les coins, peu mobiles, fragiles ...

Je cherchais donc des logiciels fonctionnant sur "tablettes" via le wifi.

3 Etat de l'art

3.1 Liaison planétarium - télescope

Il est assez standard de lier un télescope à un ordinateur. La plupart des logiciels de planétarium permettent d'envoyer des commandes à un télescope via une liaison série.

3.1.1 Liaison USB-série

L'entrée vers le télescope étant une connexion série (dans mon cas "RJ11") et la "sortie" de l'ordinateur étant de l'USB, la connexion ordinateur-télescope se fait par un convertisseur USB-série (DB9) et un câble DB9-RJ11.

3.2 Logiciels

J'ai, tout d'abord, fait un tour des logiciels open source. Malheureusement, aucun ne convenait.

3.2.1 Stellarium

Parmi les logiciels de planétarium, "Stellarium" est assez standard. Malheureusement, sa version mobile ne permet pas de commander un télescope.

3.2.2 Kstars

Kstars dans ses dernières versions (>2.1) permet un contrôle poussé d'un télescope. Malheureusement il n'y a pas de version mobile.

3.2.3 Xephem

Xephem n'a pas de version mobile.

3.2.4 Sky Safari

"Sky Safari", de la société californienne "Southern Stars", est un logiciel pour "mobile" : tablettes ou GSM. Il est à la fois très complet sans être "lourd". Dans ses versions "plus" et "pro", il est capable de piloter un télescope. Il n'est malheureusement ni gratuit ni open source. Mais son prix est minime.

3.3 Wifi

"Southern Stars" vend un boîtier qui se connecte au télescope et qui permet de connecter la tablette au télescope via le wifi. Le boîtier fait serveur "wifi" et est configurable via une interface web.

Célestron vend aussi un boîtier semblable mais qui ne fonctionne qu'avec son logiciel.

3.4 Tablettes

"Sky Safari" fonctionne aussi bien sur iPad que sur Android.

3.5 Open source

Je décris assez longuement ici ce qu'on peut entendre par "open source".
Je reconnais faire un peu de prosélytisme.

En premier lieu, il faut absolument distinguer la notion de logiciel gratuit et de logiciel libre. Ce n'est pas parce qu'un logiciel est gratuit qu'il doit être considéré comme un logiciel libre et un logiciel libre n'est pas nécessairement gratuit.

Pour clarifier un peu les choses dans une jungle relativement inextricable, j'appellerai "logiciel libre" un logiciel distribué sous la licence GPL (General Public Licence). En résumé, voici les principaux points de cette licence :

- tout le monde peut utiliser le logiciel ;
- le source du programme doit obligatoirement être fourni avec la distribution contenant le programme ;
- le point précédent est là pour permettre à quiconque de modifier le logiciel, soit pour l'adapter à des besoins particuliers, soit pour l'améliorer ;
- on a le droit de redistribuer les copies dans la mesure où on fournit également le source du programme.

Si un logiciel répond à ces exigences, il peut être qualifié de logiciel libre sinon, ce n'est pas un logiciel libre. Par exemple, une montagne de logiciels freeware ne sont pas des logiciels libres (source non disponible, interdiction de modifier le programme sans l'accord du propriétaire, ...). On remarquera qu'il n'est jamais fait mention explicite d'argent. S'il est précisé que tout le monde peut redistribuer des copies, il n'est absolument pas interdit de les faire payer. En pratique, c'est d'ailleurs ce qui se passe pour les distributions occupant beaucoup de place mémoire et qui seraient très pénibles de rapatrier via Internet. Le projet GNU a inventé une licence couvrant le logiciel libre qu'il a appelée "copyleft". Elle protège le logiciel en empêchant, par exemple, une entreprise de

modifier le programme et de déclarer alors que ce programme modifié n'est plus libre. En effet, il y a deux obligations pour le logiciel libre. La première est qu'une version modifiée d'un logiciel libre est ipso facto un logiciel libre, couverte par la même licence. La seconde est que la prise de cette licence est irrévocable : le développeur d'un programme donne irrémédiablement son produit à la communauté.

4 Sources d'inspiration

4.1 openDrive - Ethersex

Je mentionne rapidement openDrive mais cette solution va trop loin en termes de réalisation technique et pas assez par rapport à mes besoins.

<http://opendrive.gizmor.org/forum/index.php>

Il s'agit de remplacer intégralement l'électronique de contrôle d'une monture par de l'électronique et des programmes open source. Il faut une solide dose de connaissances en électronique et en programmation pour s'y attaquer. C'est aussi le genre de projet qui est extrêmement chronophage. Et je cherchais une solution à court et moyen terme.

4.2 iAstroHub

Mon goût du "bricolage" informatique (ou, si l'on préfère, du hacking) m'a fait acheter un RaspberryPi. Il est clair que cet appareil peut servir de point d'accès wifi.

<http://learn.adafruit.com/setting-up-a-raspberry-pi-as-a-wifi-access-point?view=all>

Quelques recherches sur google m'ont amené à une série de pages web intéressantes :

— <http://www.raspberrypi.org/phpBB3/viewtopic.php?t=15723&p=405831>

— <http://www.cloudynights.com/ubbthreads/showflat.php/Cat/0/Number/5369851/>

— <http://sourceforge.net/projects/iastrhub/>

Ce projet a évolué pour maintenant ne plus utiliser le raspberry mais un "mk808", mais il fonctionne encore pour l'essentiel sur Raspberry.

4.2.1 Readme.txt

Voici une traduction d'un extrait du "Readme" d'Astrohub pour pouvoir mieux appréhender ce dont il s'agit. :

iAstroHub est :

- une solution unique dans le monde des tablettes de contrôle d'appareils d'astrophotographie;
- une application web intégrant des logiciels bien établis fonctionnant en arrière-plan. L'utilisateur ayant accès à tous ces programmes via une interface web sur la tablette.

Comment Astrohub fonctionne-t-il ?

- tous les accessoires sont connectés à un hub USB. Le hub étant connecté au MK808.
- Le MK808 est configuré comme point d'accès wifi.
- IAstroHub fonctionne comme un serveur web. Le serveur web contrôle différents logiciels qui tournent en tâche de fond. Les logiciels comme les accessoires sont configurables via une interface web.
- On peut accéder à l'interface web depuis un appareil mobile : GSM ou tablette.

Les accessoires sont :

1. Autoguide-age : QHY5 ou QHY6
2. CCD : QHY8L
3. DSLR : Canon or Nikon DSLR
4. adaptateur USB série (FTDI232) pour contrôler l'appareil photo DSLR.

5. adaptateur USB série (FTDI232) pour contrôler la monture du télescope
6. Carte son USB (C-Media) et mini haut-parleur (messages de confirmations et d'alertes)
7. USB hub : Belkin Ultra Slim Hub
8. MK808

5 Matériel

5.1 Le strict nécessaire

5.1.1 Raspberry

Voici ce qu'en dit Wikipedia :

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 qui a la taille d'une carte de crédit, permet l'exécution de plusieurs variantes du système d'exploitation libre GNU/Linux et des logiciels compatibles. Il est fourni nu (carte mère seule, sans boîtier, ni alimentation, ni clavier, ni souris, ni écran) dans l'objectif de diminuer les coûts et de permettre l'utilisation de matériel de récupération.

Cet ordinateur est destiné à encourager l'apprentissage de la programmation informatique. Il est cependant suffisamment ouvert (ports USB voire réseau) et puissant (ARM 700 MHz, 256 Mo de mémoire vive pour le modèle d'origine, 512 Mo sur les dernières versions) pour permettre une grande palette d'utilisations. Son circuit graphique BMC Videocore 4 en particulier permet de décoder des flux Blu-Ray full HD (1080p 30 images par seconde), d'émuler d'anciennes consoles et d'exécuter des jeux vidéo relativement récents.

Son prix de vente était estimé à 25 \$, soit 19,09 €, début mai 2011. Les premiers exemplaires ont été mis en vente le 29 février 2012 pour environ 25 €. Début 2013, plus d'un million de Raspberry Pi ont déjà été vendus.

J'utilise la version avec 512 Mo de mémoire.

5.1.2 Carte

Le Raspberry utilise une carte SD comme disque dur.

J'ai utilisé une carte micro-SD de 16 GB avec un adaptateur vers lecteur SD.

Il vaut mieux mettre le prix pour une bonne carte car les trop faibles vitesses d'accès peuvent être pénalisantes avec une carte bon marché.

5.1.3 Alimentation

En standard, l'alimentation du Raspberry se fait via un port USB. Le voltage doit être compris entre 4,5 V et 5,7 V. Il faut une alimentation suffisamment puissante : en effet le Raspberry demande sans accessoire 700 mA. Mais des accessoires supplémentaires vont ajouter à cette demande. Un transformateur pour alimenter un GSM débite typiquement 5,5 V et 1A. L'ampérage a intérêt à être supérieur d'à peu près 200 mA par accessoire ajouté. Un transformateur USB depuis un allume-cigare sera insuffisant. Le Raspberry va rebooter dès que la demande de courant sera supérieure à ce que peut débiter votre alimentation amenant un comportement erratique.

Pour une observation sur batterie, il faut probablement utiliser la sortie 5 V d'une batterie pour télescope mais je ne l'ai pas encore fait.

Les batteries USB renseignées sur ce site <http://www.magdiblog.fr/divers/comment-alimenter-votre-raspberry-pi-avec-une-batterie/> ont l'air très bien.

5.1.4 Dongle wifi/USB

J'ai choisi le dongle d' "element14", la firme qui commercialise le raspberry : <http://be.farnell.com/element14/wifi-usb-for-raspberry-pi/dp/2133900>

La portée de celui-ci est un peu faible et je perds la connexion avec mon point d'accès wifi dans la maison si mon télescope est au fond du jardin à une dizaine de mètres. J'envisage donc d'utiliser un autre "dongle" plus puissant comme celui-ci : <http://www.adafruit.com/products/1030>.

Il faut être attentif au fait que le dongle wifi doit pouvoir être un "master" et pas seulement client.

5.1.5 Convertisseur USB/Série

L'adaptateur USB série (FTDI232) conseillé par le projet iAstroHub est celui que je possédais.

5.1.6 Tablette

J'utilise aussi bien un GSM sous Android qu'un iPad. Il faut du wifi mais c'est standard sur tous ces appareils. Avoir une boussole et un GPS dans la tablette n'est pas un luxe.

5.2 Debugging

Pour configurer le Raspberry, il peut être utile d'y accéder comme "PC" c'est à dire via des "câbles" et un écran. Ceci se fait à la maison (au labo) et pas lors d'une observation.

5.2.1 Écran

Le raspberry dispose de plusieurs sorties vidéo. J'utilise un convertisseur HDMI-VGA depuis la sortie HDMI du Raspberry vers un écran d'ordinateur.

Il est aussi possible de connecter un petit écran sensitif. Le Raspberry pourrait donc se substituer à la raquette de contrôle du "goto".

Pour l'instant, je laisse active l'interface graphique par défaut¹ et on y accède sans mot de passe.

Ce n'est pas idéal en termes de sécurité mais le but est de fournir rapidement une interface "facile" pour pouvoir résoudre des problèmes.

5.2.2 Clavier, souris

Un clavier QWERTY et une souris USB connectés à un hub USB sont des compléments nécessaires de cette configuration "DEBUG".

5.3 Autoguidage

Une webcam est une alternative aux QHY5, QHY6. C'est encore à tester.

5.4 Acquisition

J'envisageais de faire de l'astrophotographie planétaire avec une webcam Philips TOUCAM PRO qui traîne chez moi mais celle-ci est malheureusement incompatible avec le Raspberry.

Il y a une caméra Raspberry

<http://au.element14.com/jsp/search/productdetail.jsp?SKU=2302279&COM=RaspberryPi> qui se branche directement sur une nappe et qui a déjà été utilisée pour faire de l'astrophotographie : <http://www.raspberrypi.org/archives/4170>.

Mais, mon fournisseur était en rupture de stock quand j'ai voulu la commander.

1. LXDE

6 Logiciels

6.1 Sur la tablette

6.1.1 Sky Safari

Il faut la version "Plus" (\approx 25 euros) (ou "Pro") car la version de base ne permet pas de contrôler un télescope.

6.1.2 SSH

Sur la tablette pour accéder en ligne de commande au Raspberry.

6.1.3 VNC

Pour accéder en mode graphique au Raspberry.

6.1.4 web

En principe, cela vient avec iAstroHub mais je ne l'ai pas encore implémenté.

6.2 Sur le Raspberry

6.2.1 Système opératoire

J'ai installé le système le plus standard pour Raspberry : Raspbian. Il s'agit d'une version de Debian Linux (Wheezy) adaptée pour le Raspberry.

6.2.2 Logiciels

Voici une liste non exhaustive des logiciels que j'ai installé selon

<http://sourceforge.net/projects/iastrohub/files/>

Je n'ai pas encore réussi à installer convenablement LAZARUS et n'ai donc malheureusement pas pu compiler "Cartes du ciel".

- gphoto2
- Lin_guider
- CCD
- INDI
- xvfb et vnc
- xdotool
- Astrometry
- DslrDashBoard
- NGinx

6.3 Configurations

6.3.1 Raspberry

Sky Safari envoie ses instructions par le réseau. Il faut donc configurer le wifi.

6.3.1.1 Point d'accès wifi J'avais configuré le Raspberry pour accéder à mon wifi à la maison.

Ceci amène parfois des conflits si je suis à la limite de la portée du point d'accès de la maison. Le Raspberry basculant entre les deux configurations. Il faut désactiver cette possibilité. L'interface wifi est donc configurée en "statique" alors que le RJ45 est laissé en dhcp.

```
# vi /etc/network/interfaces

auto lo
iface lo inet loopback

iface eth0 inet dhcp

allow-hotplug wlan0
iface wlan0 inet static
address 10.0.0.1
netmask 255.255.0.0
gateway 10.0.0.1
```

Le Raspberry est configuré comme point d'accès avec comme nom "iAstroHub" et avec "10.0.0.1" comme adresse IP.

Le mot de passe du point d'accès wifi est "abcdeabcdeabc".

Les logiciels doivent être installés :

```
sudo apt-get install iw
sudo apt-get install hostapd
sudo apt-get install dnsmasq
```

Il faut configurer le logiciel de point d'accès "hostap" :
Commençons par modifier le fichier de configuration par défaut :

```
# vi /etc/default/hostapd

DAEMON_CONF="/etc/hostapd/hostapd.conf"

# vi /etc/hostapd/hostapd.conf

interface=wlan0
driver=nl80211
ssid=iAstroHub
channel=3
hw_mode=g
macaddr_acl=0
ignore_broadcast_ssid=0
auth_algs=1
wep_default_key=0
wep_key0="abcdeabcdeabc"
```

Ensuite, il faut configurer "dnsmasq" qui fait serveur dhcp :

```
# vi /etc/dnsmasq.conf
```


Il faut préciser que l'on passe par le wifi ("wlan0") et que les adresses IP disponibles vont de 10.0.0.2 à 10.0.0.5 (4 adresses c'est plus qu'assez).

```
interface=wlan0
dhcp-range=10.0.0.2,10.0.0.5,255.255.255.0,12h
```

Les modifications doivent être activées au démarrage. Ceci peut se faire via le fichier `"/etc/rc.local"` :

```
# vi /etc/rc.local

/sbin/ifconfig wlan0 down
/sbin/ifconfig wlan0 10.0.0.1 netmask 255.255.255.0 up
/usr/sbin/service hostapd restart
/usr/sbin/service dnsmasq restart
/sbin/ifconfig wlan0 10.0.0.1 netmask 255.255.255.0 up
```

La dernière ligne semble nécessaire si le port RJ45 n'est pas utilisé. Apparemment "hostap" a pour effet de faire perdre sa configuration réseau au wifi ! D'où la nécessité de reconfigurer celui-ci.

6.3.1.2 Mode graphique NO! Il est impératif de faire fonctionner le raspberry en mode "CLI" c-à-d. en ligne de commande.

S'il démarre en mode graphique, il active automatiquement un menu graphique pour configurer le wifi. Cette configuration "écrase" celle du point d'accès.

Ceci m'a coûté TROIS HEURES de debug : le serveur démarre puis se crashe.

On peut lancer le mode graphique via la commande "startx" et alors la configuration wifi n'est pas écrasée.

6.3.2 Port série

Les instructions envoyées par Sky Safari sont reçues sur le port ethernet 5555 du Raspberry et redirigées vers le port USB série. (Ses instructions sont les coordonnées équatoriales de l'objet que l'on veut observer.)

```
python /home/pi/www/tcp_serial_redirect.py -p /dev/ttyUSB0 -P 5555 &
```

Cette commande doit aussi être placée dans `"/etc/rc/local"`.

Le fichier `"tcp_serial_redirect.py"` peut être placé ailleurs, par exemple dans `"/usr/local/bin"`.

Dans la version de départ du projet, c'est "ser2net" qui était utilisé mais celui-ci n'a jamais vraiment fonctionné chez moi.

6.3.3 Sky Safari

Il faut aller dans le menu "settings" de Sky Safari.

Se diriger alors vers "Telescope" et là vers "Setup"

Il faut préciser une série de points :

— Scope Type : chez moi "Celestron CGE/CGEM :CGE Pro"

— Mount Type : chez moi "Equatorial GoTo (German)"

— Communication Settings :

— Choisir "Connect vi WIFI" (le bluetooth est peut-être une solution intéressante.)

- Auto-Detect SkyFi : laisser non sélectionné
- IP Address : "10.0.0.1"
- Port Number : "5555" (Voir la configuration du Raspberry pour ces 2 derniers points)
- Set Time et Location : laisser non sélectionné
- Save Log File : au choix, je sélectionne pour le "debug".
- Readout rate : "4 per second"
- SkyFi Settings Web Page : ouvre un "browser" pour accéder à l'interface web qui n'est pas encore implémentée.

Il est aussi recommandé de faire apparaître le viseur. Cette option s'active dans la suite du menu.

6.3.4 Modèle de télescope

Il y a une très grande gamme de modèles de télescope.

6.3.4.1 Déterminer le modèle de monture via la raquette Il y a souvent moyen de savoir quel est le modèle de la monture via la raquette.

7 Premiers essais

7.1 Ce qui fonctionne

7.1.1 Côté télescope

7.1.1.1 Télescope Il faut régler son télescope comme d'habitude càd. la mise en station et la triangulation standard.

7.1.1.2 Câbles On connecte alors le port série du télescope au Raspberry via le convertisseur USB-série.

7.1.1.3 Raspberry Alors, et alors seulement, on peut mettre le Raspberry sous tension. Il n'est pas question de risque mais c'est ce qu'il faut faire pour que le Raspberry "voit" le télescope.

7.1.2 Sky Safari

Il faut aller dans le menu "scope" de Sky Safari.

Appuyer alors sur "connect".

Sélectionner un objet à observer et appuyer sur "GoTo".

"Enjoy the sky!" (Ça c'est pour Linda!)

7.2 À régler

7.2.1 Wifi

Je devrais "nettoyer" la configuration.

Une antenne plus puissante ne serait pas un luxe.

7.2.2 Sécurité

La clé wifi est en "wep". Il serait mieux de passer à du "wpa2".

7.2.3 Alimentation du Raspberry

Pour l'instant, ceci fonctionne sur secteur. Sur la batterie du télescope, via un chargeur USB "allume-cigare" le Raspberry redémarre toutes les cinq minutes! Il faut une batterie capable de délivrer du 2 ampères pendant plusieurs heures!

J'ai commandé une batterie.

7.2.4 Script d'installation

Il faudrait écrire un script d'installation "de base".

Ce n'est pas très compliqué en ce qui concerne le réseau.

Là où il faut downloader et compiler, c'est un peu plus lourd (gestion des erreurs).

8 Alternatives

8.1 Stellarium et javascript

Comme annoncé, la version mobile de Stellarium ne permet pas de contrôler un télescope, mais j'ai testé ceci avec succès sur PC : <https://github.com/fconline/node-telescope-server>

Node est une implémentation de javascript "stand alone".

9 Le futur

9.1 Passer au mk808

Ce nano-ordinateur est plus puissant que le Raspberry et moins gourmand en watts. Il est dépourvu de port RJ45.

9.2 Interface web

Le serveur web "nginx" est installé et activé mais il n'accède encore à rien du tout. La documentation de iAstroHub est déplorable sur ce point.

9.3 Acquisition

Je suis tenté par l'ajout d'un disque dur externe (pour stocker des photos) et la commande d'une webcam Raspberry "picam". J'ai testé quelques webcams. Certaines fonctionnent d'autres pas!

9.4 Autoguidage

Comme pour le point précédent je n'ai encore aucune expérience dans ce domaine. Mon idée est d'utiliser, soit la "picam", soit une webcam pour l'autoguidage et/ou pour de l'acquisition;

10 État des lieux

La plupart des points précédents ont été réalisés avant novembre 2014. Comme le projet évolue, je vais tenir ici un "logbook".

10.1 Au 29 décembre 2015

J'ai acquis un Raspberry 2 et l'ai configuré faire "point d'accès" wifi et pour piloter le télescope. Les premiers tests sont positifs.

J'ai testé une "picam" sur un Raspberry B+.

Je voudrais installer et configurer " DslrDashBoard" sur le Raspberry 2.

11 Conclusion

Même s'il y a encore beaucoup de progrès à faire (sécurité, interface web, autoguidage, astro-photographie, ...), le "strict minimum", à savoir le pilotage du télescope depuis une tablette via le wifi est déjà très satisfaisant.

Il est sans doute plus facile d'utiliser le boîtier "SkiFi" mais l'occasion a fait le larron. Il était amusant et intéressant d'apprendre comment "Sky Safari" et d'autres logiciels dialoguaient avec le télescope.

Dans l'état, il s'agit encore d'un projet de "geek". Le rendre un peu plus "user friendly" ne serait pas un luxe !

Mais il a le mérite d'exister et d'être utilisable pour une fraction du prix de la solution "SkyFi".

J'ai bien l'intention de pousser plus ce projet.

Table des matières

1	Le Pourquoi	2
2	Le but	2
3	Etat de l'art	2
3.1	Liaison planétarium - télescope	2
3.1.1	Liaison USB-série	2
3.2	Logiciels	2
3.2.1	Stellarium	2
3.2.2	Kstars	3
3.2.3	Xephem	3
3.2.4	Sky Safari	3
3.3	Wifi	3
3.4	Tablettes	3
3.5	Open source	3
4	Sources d'inspiration	4
4.1	openDrive - Ethersex	4
4.2	iAstroHub	4
4.2.1	Readme.txt	4
5	Matériel	5
5.1	Le strict nécessaire	5
5.1.1	Raspberry	5
5.1.2	Carte	5
5.1.3	Alimentation	5
5.1.4	Dongle wifi/USB	6
5.1.5	Convertisseur USB/Série	6
5.1.6	Tablette	6
5.2	Debugging	6
5.2.1	Écran	6
5.2.2	Clavier, souris	6
5.3	Autoguidage	6
5.4	Acquisition	6
6	Logiciels	7
6.1	Sur la tablette	7
6.1.1	Sky Safari	7
6.1.2	SSH	7
6.1.3	VNC	7
6.1.4	web	7
6.2	Sur le Raspberry	7
6.2.1	Système opératoire	7
6.2.2	Logiciels	7
6.3	Configurations	7
6.3.1	Raspberry	7
6.3.1.1	Point d'accès wifi	7
6.3.1.2	Mode graphique NO!	9
6.3.2	Port série	9
6.3.3	Sky Safari	9
6.3.4	Modèle de télescope	10
6.3.4.1	Déterminer le modèle de monture via la raquette	10

7 Premiers essais	10
7.1 Ce qui fonctionne	10
7.1.1 Côté télescope	10
7.1.1.1 Télescope	10
7.1.1.2 Câbles	10
7.1.1.3 Raspberry	10
7.1.2 Sky Safari	10
7.2 À régler	10
7.2.1 Wifi	10
7.2.2 Sécurité	10
7.2.3 Alimentation du Raspberry	11
7.2.4 Script d'installation	11
8 Alternatives	11
8.1 Stellarium et javascript	11
9 Le futur	11
9.1 Passer au mk808	11
9.2 Interface web	11
9.3 Acquisition	11
9.4 Autoguidage	11
10 État des lieux	11
10.1 Au 29 décembre 2015	12
11 Conclusion	12