

PROJET DE MASTER 2

Autoscope

Cinquième compte rendu

Auteurs :

Thomas ABGRALL
Clément AILLOUD
Thibaud LE DOLEDEC
Thomas LEPOIX

MASTER Systèmes Embarqués

E.S.T.E.I.

École Supérieure des Technologies Électronique, Informatique, et Infographie
Département Systèmes Embarqués

15 avril 2019

Table des matières

Table des matières	1
I Partie de groupe	2
1 Avancement général	3
1.1 Prévisions pour le dernier sprint	3
2 Accessibilité au projet sur internet	5
2.1 Dépôt principal	5
2.2 Dépôt du système d'exploitation de la Raspberry-Pi	6
2.3 Dépôt du plugin de Stellarium	6
3 Avancement par tâche	7
3.1 Hardware	7
3.2 Système d'exploitation	7
3.3 Communication inter-processus	9
3.4 Stellarium	10

Première partie

Partie de groupe

Chapitre 1

Avancement général

En raison de l'abandon du projet par Thibaud LE DOLEDEC et Clément AILLOUD partis en stage, le projet ne pourra être mené à terme. Pour aborder la dernière ligne droite avant l'évaluation finale, un choix a donc été fait des tâches sur lesquelles travailler en priorité, au détriment d'autres qui demeureront inachevées.

Ci-dessous un diagramme représentant l'avancement des différentes tâches du projet. Le gris indique qu'une tâche est terminée ou à un niveau d'avancement satisfaisant et garantissant une maturité proche. Le orange indique les tâches sur lesquelles nous travaillerons en priorité avant la fin du projet. Les flèches représentent des liens de dépendance entre les tâches.

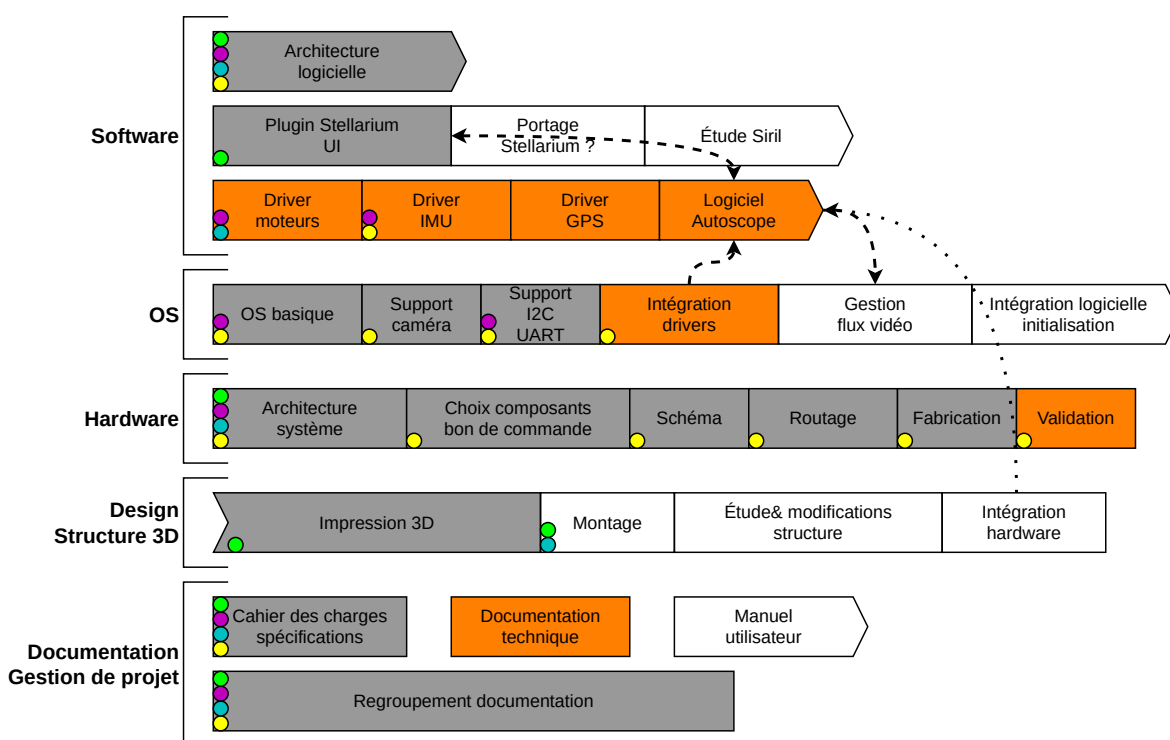


FIGURE 1.1 – Diagramme de l'organisation temporelle du travail sur le projet

1.1 Prévisions pour le dernier sprint

Malgré son aspect visuel intéressant pour promouvoir le projet ainsi que de son aspect central (il s'agit tout de même d'un télescope), le travail sur la structure du télescope sera

écarté des tâches prioritaires. Cela pour deux raisons :

- Il s'agit d'une partie du projet demandant beaucoup de temps et d'implications par rapport à ce dont nous disposons. Nous n'aurions sans doute pas le temps de terminer cela avant l'évaluation.
- N'ayant ni des connaissances particulières en optique, ni la maîtrise d'un logiciel de modélisation 3D, nous ne sommes pas plus qualifiés qu'une personne aléatoire voulant contribuer au projet. Nous pensons donc qu'il vaut mieux nous concentrer sur des tâches faisant partie de notre domaine de qualification, que nous serions capable de réaliser plus facilement ou mieux qu'une personne aléatoire. L'élaboration des drivers correspond typiquement à ce cas.

Le travail sur les drivers, quant à lui devra être avancé autant que faire se peut. En effet le logiciel principal du télescope dépend lourdement des interfaces avec les drivers et ne peut être commencé tant que les drivers ne sont pas fonctionnels.

De plus, ce logiciel étant la clef de voûte du projet, la finalisation de certaines tâches comme certains éléments du plugin Stellarium ou la gestion du flux vidéo au sein de l'OS doit être réalisée conjointement à l'écriture de ce logiciel.

L'absence de ces drivers commence donc à bloquer l'évolution d'autres tâches.

Enfin un travail de documentation devra être fait pour qu'il soit possible à une personne future (nous-mêmes ou qui que ce soit) de travailler sur ce projet ou de réutiliser notre travail.

Chapitre 2

Accessibilité au projet sur internet

Le projet étant libre, il est disponible sur Github sous licence GPL-2. Nous tachons d'accompagner nos différents dépôts d'une documentation claire permettant d'obtenir les informations suivantes :

- Une description brève et précise sur le contenu du dépôt et son rôle au sein du projet.
- Une explication de la procédure à suivre pour utiliser le contenu du dépôt.
- Une explication de la procédure à suivre pour travailler sur le dépôt.

2.1 Dépôt principal

`https://github.com/thibaudledo/Autoscope`

Le dépôt est organisé comme suit :

- Branche **master** : Sources du logiciel principal et explications sur le projet dans son ensemble.
- Release **alpha** : Paquets et fichiers binaires pour utiliser le projet "out of the box" (release expérimentale).
- Branche **hardware** : Fichiers Blender de la structure du télescope et fichiers KiCad de la carte électronique du projet.
- Branche **doc** : Documentations et datasheets des composants et éléments utilisés pour le projet.
- Branche **latex** : Fichiers LaTeX et **.pdf** des comptes rendus sur le projet.
- Branche **hello_mod** : Sources d'un driver helloworld servant d'exemple.
- Branche **a4988_mod** : Sources du driver des contrôleurs moteur et des capteurs de fin de course des moteurs.
- Branche **mpu_9250_mod** : Sources du driver de la centrale inertielle.
- Branche **mtk3339_mod** : Sources du driver du GPS.

2.2 Dépôt du système d'exploitation de la Raspberry-Pi

<https://github.com/thomaslepoix/meta-autoscope>

Il s'agit de la couche de métadonnées utilisées par Yocto pour construire le système d'exploitation Linux utilisé par la Raspberry-Pi du télescope. Une image pré-compilée du système d'exploitation figurera sur la release **alpha** du dépôt principal.

Le dépôt est organisé comme suit :

- Branche **rpi** : Métadonnées Yocto et explications de comment compiler et installer le système d'exploitation.
- Branche **rpi-repo** : Données utilisées par Repo pour synchroniser le dépôt à d'autres dépôts de métadonnées Yocto utilisées pour construire l'OS.

2.3 Dépôt du plugin de Stellarium

<https://github.com/thibaudledo/Autoscope-Stellarium-plugin>

Il s'agit du plugin de Stellarium contenant l'interface par laquelle l'utilisateur interagira avec le télescope. Un paquet pré-compilé pour linux de Stellarium incluant le plugin figure sur la release **alpha** du dépôt principal.

Le dépôt est organisé comme suit :

- Branche **master** : Sources du plugin, patch des sources de Stellarium et explications de comment compiler une version de Stellarium intégrant le plugin.

Chapitre 3

Avancement par tâche

Les différentes avancées depuis la dernière revue de projet sont expliquées ici sans rentrer dans les détails techniques. Ceux-ci figureront dans le rapport de l'évaluation finale.

3.1 Hardware

Le circuit imprimé a été fabriqué sans encombre. Il est actuellement en cours de validation.

- Aucune fuite de courant ni discontinuité de piste.
- Le convertisseur 12V/5V est opérationnel.
- La protection aux surtensions l'est également.
- L'environnement des interrupteurs de butée n'a pas encore été testé.

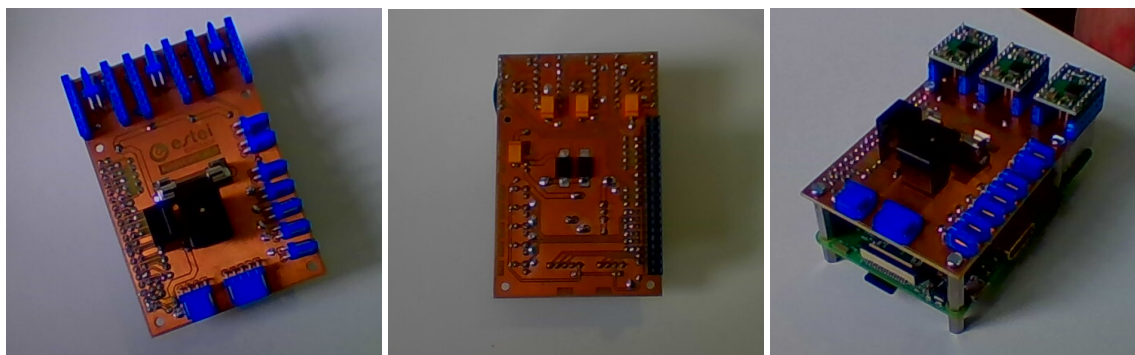


FIGURE 3.1 – Photos du circuit imprimé du télescope

3.2 Système d'exploitation

La configuration du hotspot Wifi a été rétablie, il s'agissait d'une erreur discrète dans un fichier de configuration de `connman`. Voici donc un aperçu de la connexion au télescope depuis un ordinateur distant, ainsi que du serveur FTP qui redevient par là même utilisable.

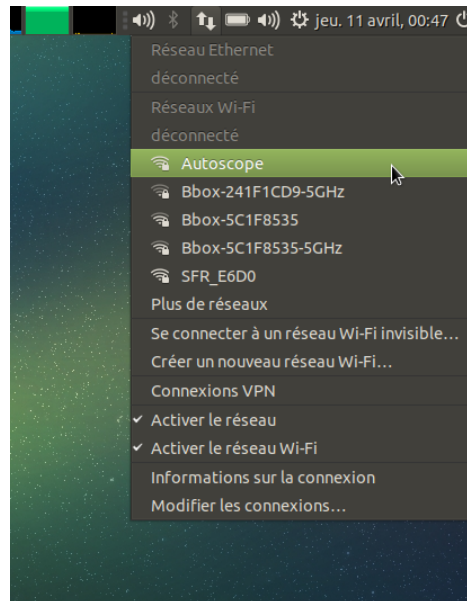


FIGURE 3.2 – Connexion à la Raspberry-Pi depuis un ordinateur distant

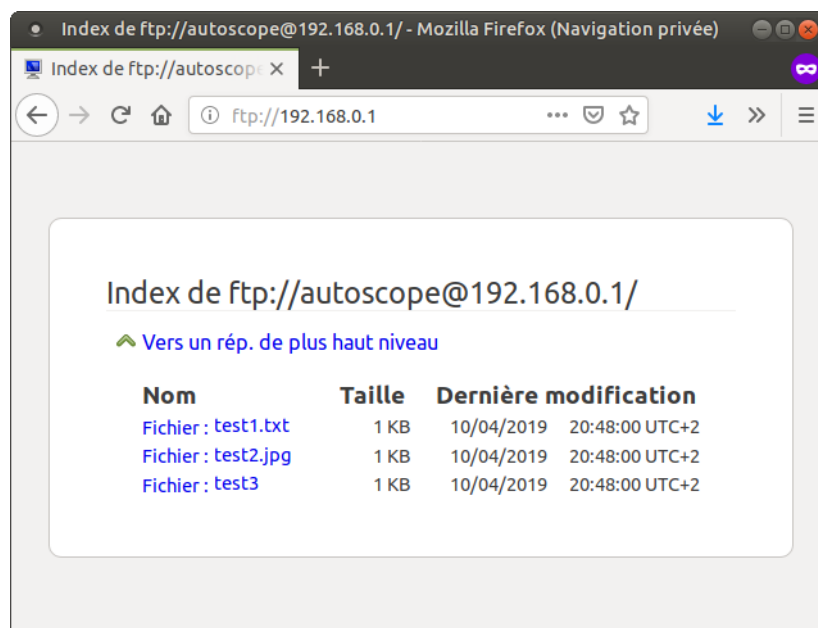


FIGURE 3.3 – Accès au serveur FTP de la Raspberry-Pi depuis un navigateur

De plus le support de la liaison UART et du bus I2C ont été activé. Ci dessous des trames relevées à l'analyseur logique pour chaque protocole. La trame I2C n'est pas acquittée car aucun périphérique n'est présent pour le faire, il s'agit d'une trame d'adresse envoyée dans le vide par la Raspberry-Pi.

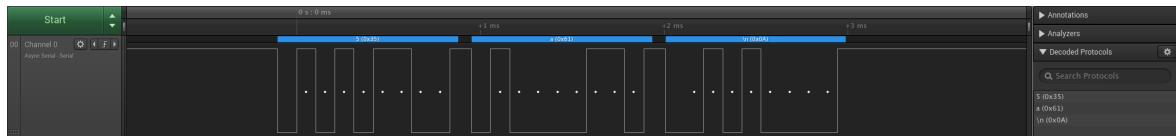


FIGURE 3.4 – Trame UART relevée à l'analyseur logique



FIGURE 3.5 – Trame I2C relevée à l'analyseur logique

3.3 Communication inter-processus

Cette partie consiste à rendre le driver accessible à l'application principal du système. Pour cela j'utilise la manière qui utilise `ioctl()`. Le moyen technique utilise des fichiers virtuels en lecture/écriture entre l'espace utilisateur (user-space, dans lequel il y aura l'application) et la couche kernel (dans laquelle il y a les drivers d'accessible). Via ce fichier les ordres sont envoyés au driver. Un code utilisant l'`ioctl` à été validé, l'implémentation dans le driver de contrôle des moteurs est bientôt finie.

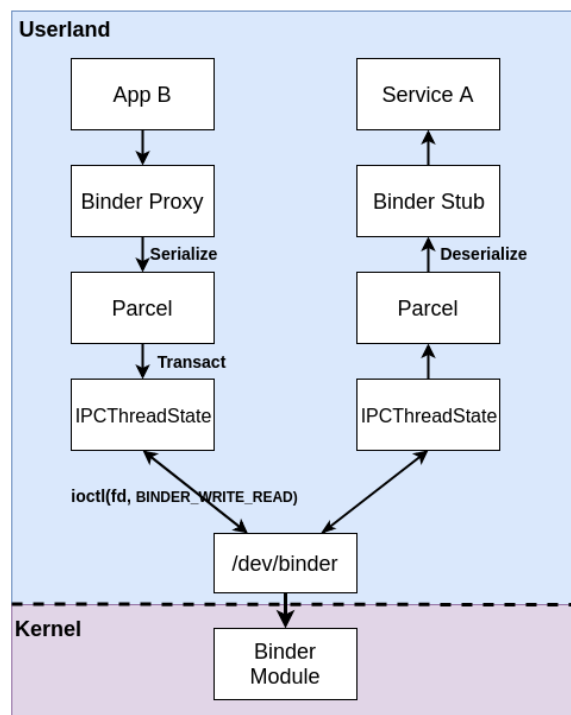


FIGURE 3.6 – Schéma des couches lors de la communication app/driver

3.4 Stellarium

La procédure de compilation de Stellarium ainsi que de ses dépendances a été étudiée et détaillée dans le **README** du dépôt du plugin produit par Thibaud LE DOLEDEC.

<https://github.com/thibaudledo/Autoscope-Stellarium-plugin>

Le plugin devient alors accessible et utilisable. La procédure permet en particulier de livrer des paquets binaires pré-compilés d'une version de Stellarium intégrant le plugin, qu'il ne reste alors qu'à installer comme n'importe quel logiciel. Un paquet compilé pour Linux est disponible ici : <https://github.com/thibaudledo/Autoscope/releases>

Ci-dessous un aperçu de l'interface de l'interface qu'offre le plugin :

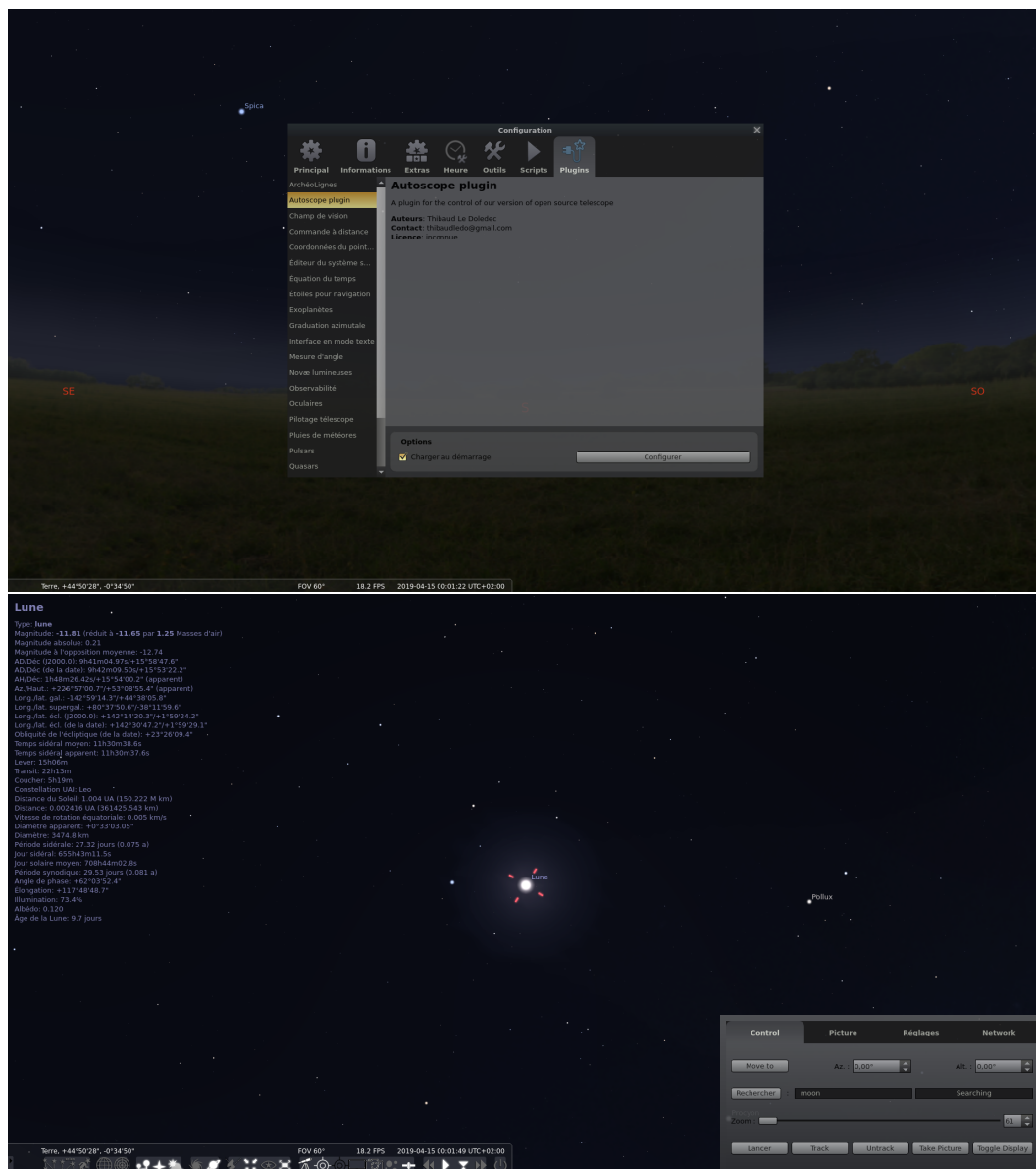


FIGURE 3.7 – Aperçu de l'interface du plugin Stellarium