



estei
BORDEAUX

ÉCOLE SUPÉRIEURE DES TECHNOLOGIES
ÉLECTRONIQUE, INFORMATIQUE, INFOGRAPHIE

PROJET DE MASTER 2

Autoscope Troisième 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

Table des matières

Table des matières	1
I Partie de groupe	2
1 Avancement général	3
II Thomas LEPOIX	4
2 Hardware	5
2.1 Rappel sur l'architecture de la carte	5
2.2 Conception du circuit	6
2.3 Contraintes de design	7
2.3.1 Contraintes électromagnétiques	7
2.3.2 Contraintes mécaniques	7
2.4 Routage de la carte	9
III Thomas ABGRALL	10
3 Driver de la commande des moteurs	11
3.1 Rappel	11
3.2 Fonctionnalités	11
3.3 Câblage	12
3.4 Avancement du programme	12
3.5 Procédure de test	12
IV Clément AILLOUD	14
4 Driver de la centrale inertielles (IMU)	15
4.1 Industrial I/O subsystem	15
4.2 Calculs de la position de la centrale inertielles dans l'espace	15
4.2.1 Calcul d'angle entre deux vecteurs	16

Première partie

Partie de groupe

Chapitre 1

Avancement général

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 qu'une tâche est particulièrement urgente ou critique pour le projet.

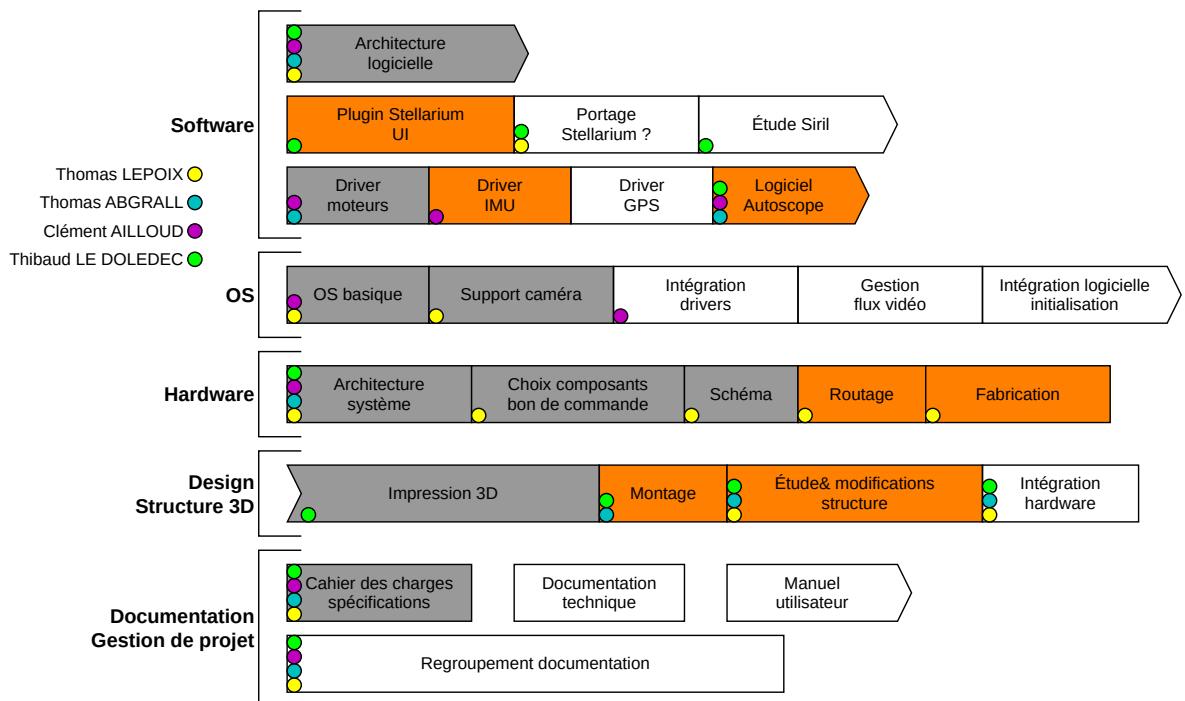


FIGURE 1.1 – Diagramme de l'organisation temporelle du travail sur le projet

Deuxième partie

Thomas LEPOIX

Chapitre 2

Hardware

2.1 Rappel sur l'architecture de la carte

En guise de rappel, ci-dessous un schéma de l'architecture de la carte du télescope.

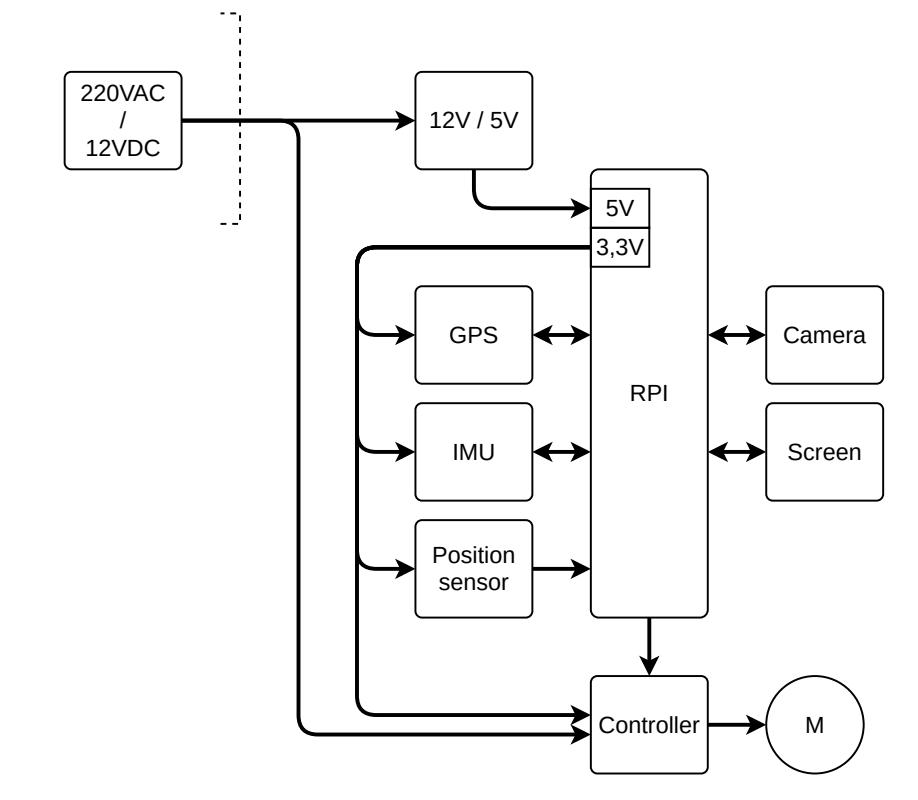


FIGURE 2.1 – Schéma structurel de premier niveau du télescope

2.2 Conception du circuit

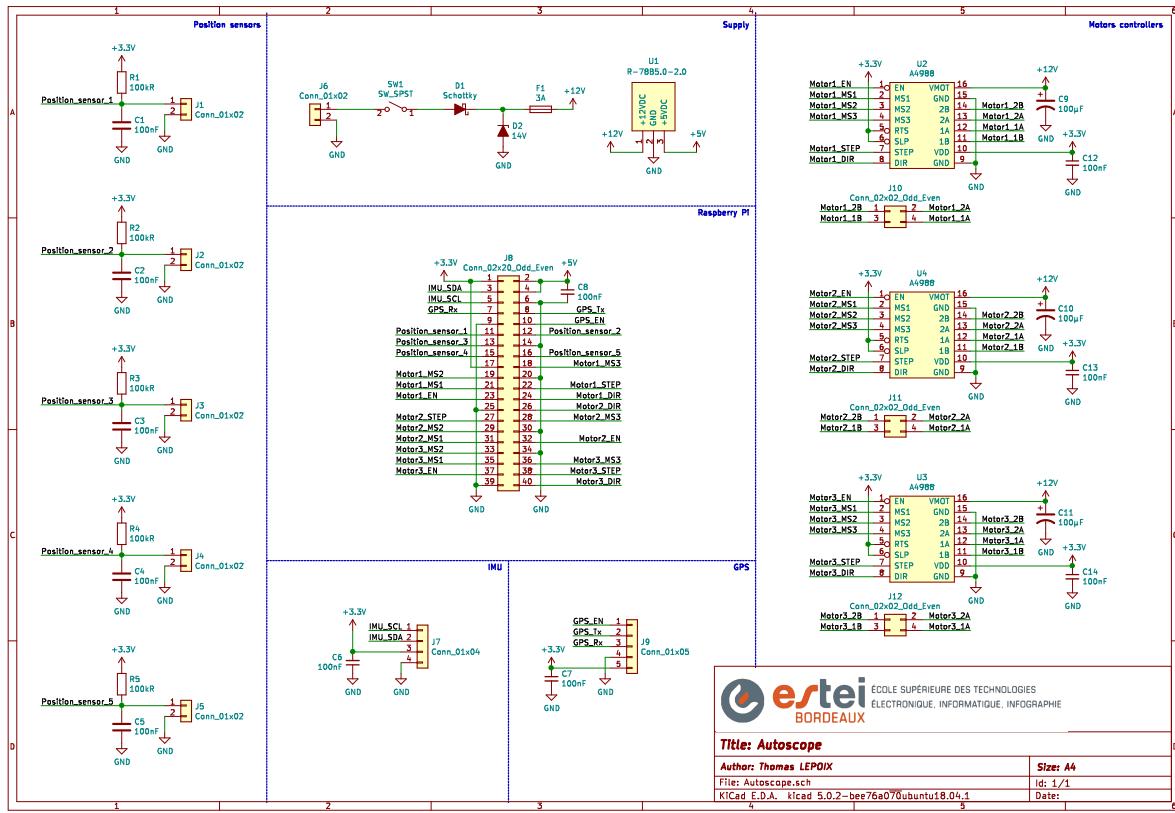


FIGURE 2.2 – Schéma structurel de la carte du télescope

Ce schéma ne présente pas de subtilité particulière, la plupart des composants étant des connecteurs.

L'alimentation est composée de :

- Un interrupteur d'allumage
- Une diode polarisante
- Une diode zener protégeant des surtensions
- Un fusible protégeant des surintensités
- Un convertisseur DC/DC intégré

L'environnement des boutons poussoirs servant de capteurs de butée aux mouvements du télescope est le suivant :

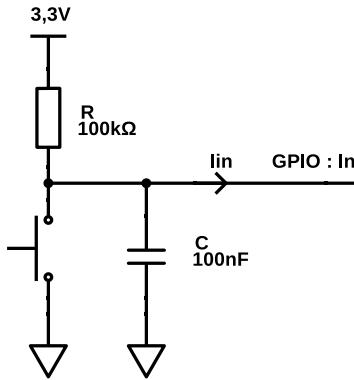


FIGURE 2.3 – Schéma de l'environnement des capteurs de butée

La valeur élevée des résistances de pullup $100k\Omega$ a pour but de réduire au maximum le courant consommé lors de l'appui, à $33\mu A$. Le courant prélevé par l'entrée GPIO de la Raspberry Pi est de l'ordre de $0,5\mu A$.

Les condensateurs de $100nF$ permettent de filtrer les parasites générés par les rebonds propres aux boutons ainsi que les perturbations électromagnétiques.

2.3 Contraintes de design

2.3.1 Contraintes électromagnétiques

La première contrainte vient de la proximité du système électronique de deux moteurs, ceux-ci générant d'importantes perturbations électromagnétiques. Cela peut être particulièrement dérangeant pour le fonctionnement de la centrale inertielle et du GPS.

La solution la plus simple et efficace est de déporter ces deux modules le long de la structure du télescope.

2.3.2 Contraintes mécaniques

Ensuite viennent les contraintes mécaniques de l'association de la carte à la Raspberry Pi.

Tout d'abord, l'emplacement du connecteur et des fixations sont à prendre en compte.

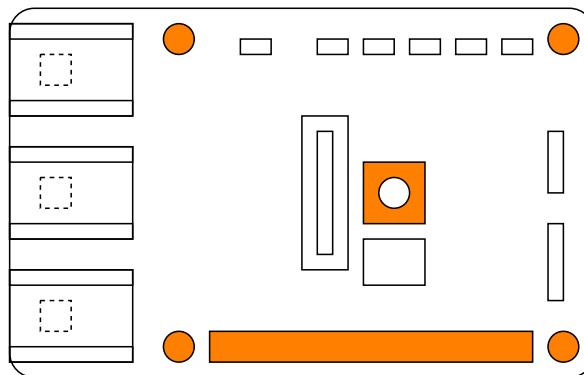


FIGURE 2.4 – Schéma mécanique de la carte vue de dessus

Le connecteur d'alimentation, centré sur la carte, est un connecteur cylindrique comme ceux des ordinateurs portables. Le télescope étant amené à tourner sur lui-même, ce connecteur devrait permettre le mouvement tout en empêchant le câble d'alimentation de s'emmêler ou de se détériorer.



FIGURE 2.5 – Connecteurs d'alimentation utilisés

Puis concernant la distance entre la carte et la Raspberry Pi, la hauteur des plus hauts éléments de la Raspberry Pi est à prendre en compte. Ainsi que la hauteur de certains condensateurs de la carte, ne pouvant donc être placés n'importe où.

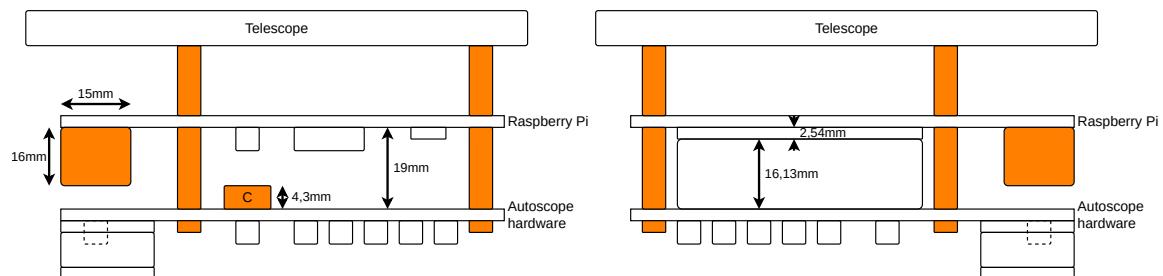


FIGURE 2.6 – Schéma mécanique de la carte vue de profil

Il faudra de plus utiliser un connecteur particulièrement haut (16,13mm) pour relier la carte à la Raspberry Pi.

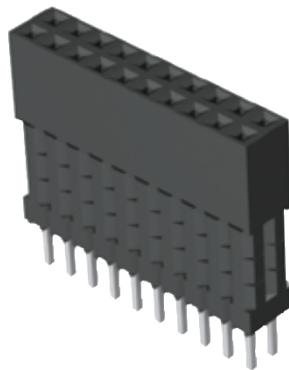


FIGURE 2.7 – Type de header utilisé

2.4 Routage de la carte

Le routage de la carte n'est pas terminé, voici un aperçu de son avancement actuel :

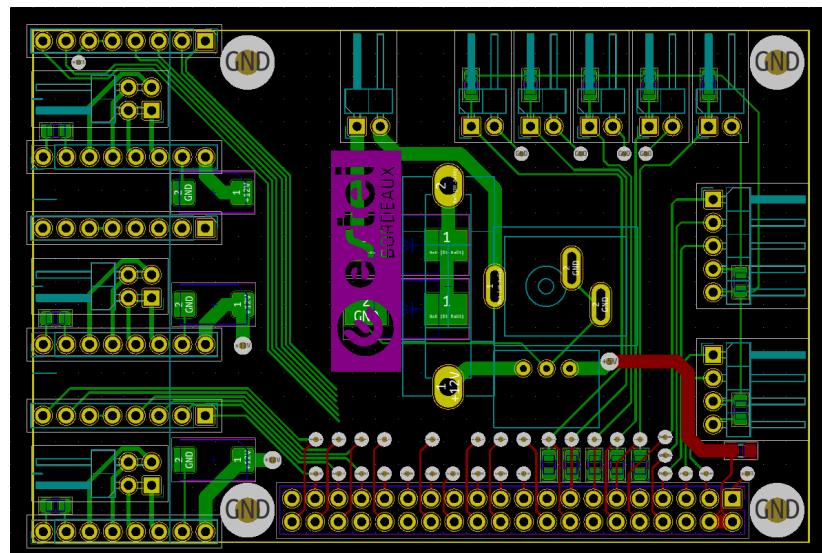


FIGURE 2.8 – Aperçu du routage en cours

Troisième partie

Thomas ABGRALL

Chapitre 3

Driver de la commande des moteurs

3.1 Rappel

Le driver moteur a pour mission de permettre de contrôler les moteurs de rotation, inclinaison et zoom du télescope. Mais également assurer la sécurité du système de pilotage à bas niveau. Le système de sécurité consiste à interrompre le mouvement d'inclinaison ou de zoom si ils atteignent leur fin de course.

3.2 Fonctionnalités

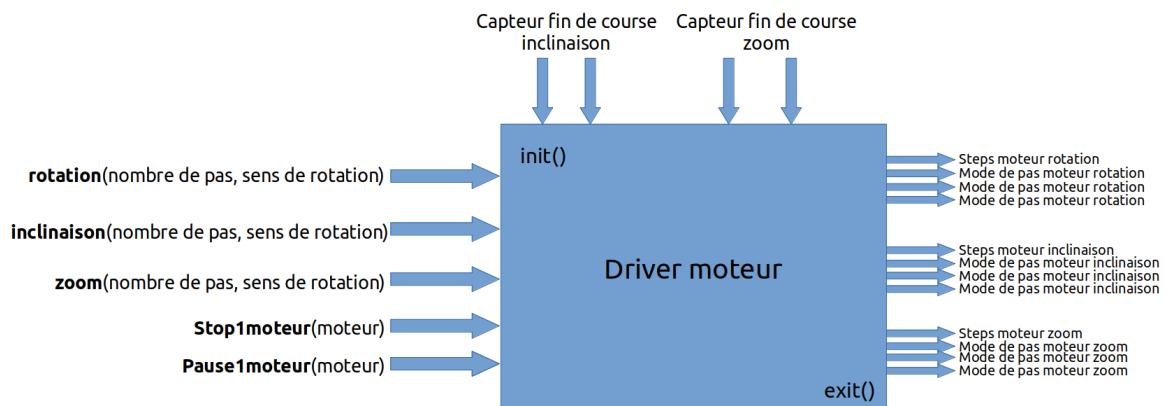


FIGURE 3.1 – Schéma des entrées / sorties du driver moteur

Le driver va permettre l'utilisation des fonctions qui se trouvent à gauche du bloc "driver moteur". Chacune des fonctions agit sur les sorties. Les capteurs de fin de course servent de sécurité, cette partie est indépendante du reste et arrête le moteur qui a atteint sa position de butée. Les sorties "mode de pas moteur..." permettent de choisir dans quel mode de rotation doit tourner le moteur (pas complet, demi-pas, quart de pas, huitième de pas, seizième de pas). Réduire la division de pas permet d'affiner la rotation afin d'atteindre l'angle le plus exact possible à ce qui a été commandé.

3.3 Câblage

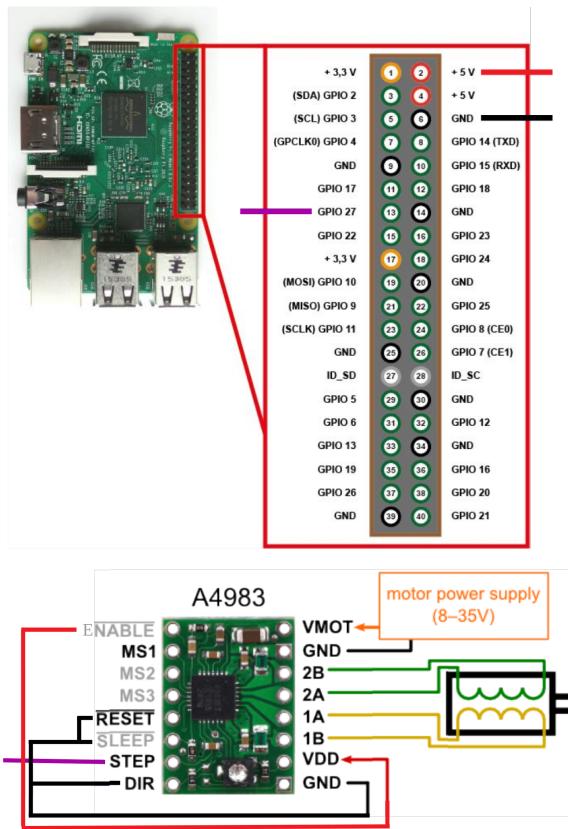


FIGURE 3.2 – Schéma de câblage d'un contrôleur moteur sur la Raspberry Pi

3.4 Avancement du programme

La première version du driver qui est validé depuis fin janvier permet :

- Initialiser le driver
- Lancer la rotation (à 5ms max par période de step)
- Lancer l'inclinaison (à 5ms max par période de step)
- Stopper les moteurs grâce aux capteurs de fin de course

À suivre :

- La partie qui consiste à changer de mode de pas est codé mais pas testé.
- Étendre le système pour gérer le zoom.

3.5 Procédure de test

Pour valider mon driver, je l'ai testé avec une Raspberry Pi, un contrôleur moteur et un moteur.

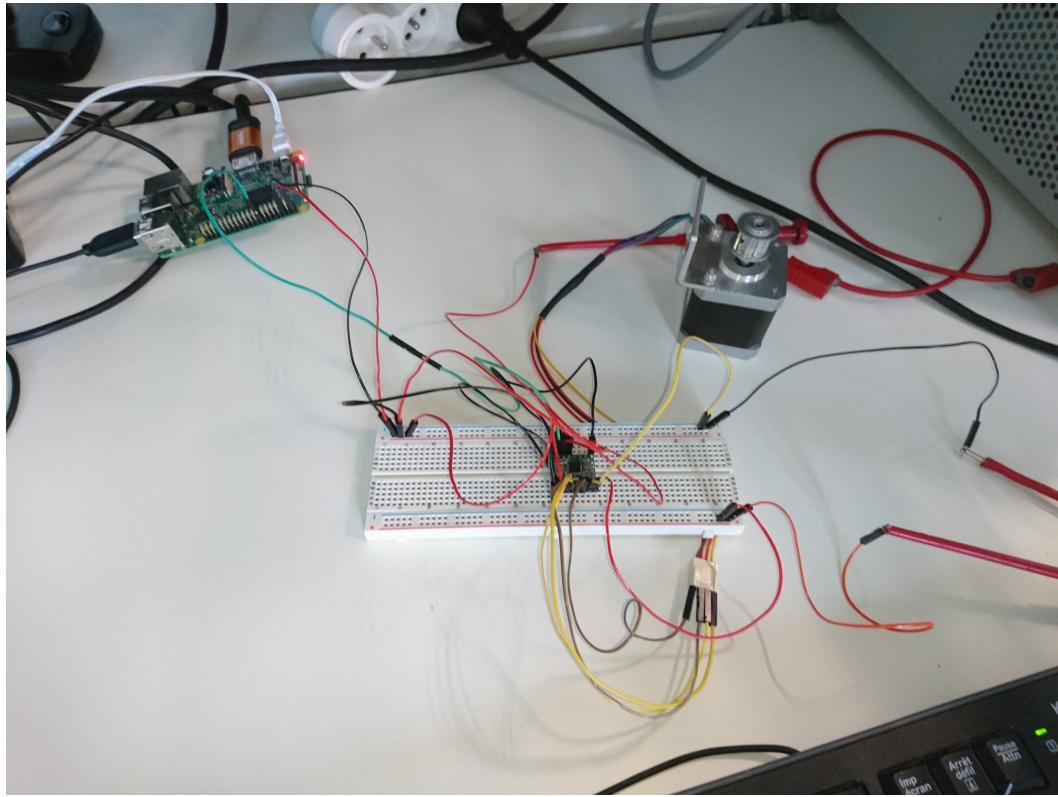


FIGURE 3.3 – Photo du banc de test

J'ai pu valider les fonctionnalités développées, cependant la vitesse max atteinte par le moteur contrôlé par le driver est inférieure à celle que l'on peut atteindre en contrôlant le moteur avec un générateur basse fréquence. Voici le résultat de la commande envoyée par la Raspberry Pi par la pin Step au contrôleur moteur pour faire avancer à chaque front montant le moteur de 1 pas.

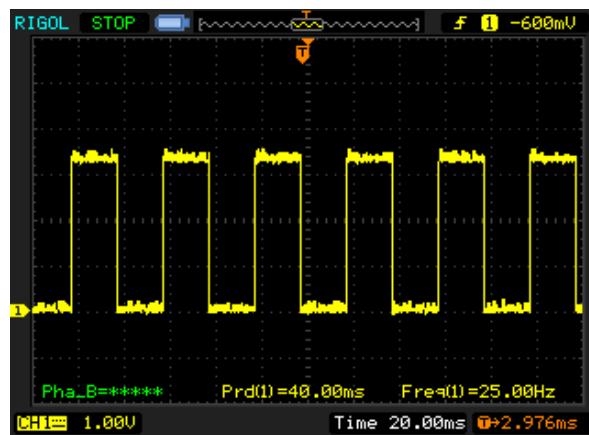


FIGURE 3.4 – Mesure à l'oscilloscope de la commande moteur générée

Quatrième partie

Clément AILLOUD

Chapitre 4

Driver de la centrale inertielles (IMU)

4.1 Industrial I/O subsystem

L'objectif de ce sous-système est de faciliter les interactions utilisateur avec un périphérique depuis l'espace utilisateur d'un système Linux. En général, cela s'adresse aux périphériques réalisant une conversion analogique-numérique ou une conversion numérique-analogique.

L'accès aux données se fait par l'intermédiaire d'un ou plusieurs fichiers :

```

1 root@raspberrypi0-wifi:~# ls sys/bus/iio/devices/iio\:device0
2 dev          in_accel_z_raw      in_anglvel_z_raw    power/
3 in_accel_scale   in_anglvel_scale   in_temp_input    subsystem/
4 in_accel_x_raw    in_anglvel_x_raw    name           uevent
5 in_accel_y_raw    in_anglvel_y_raw    of_node/

```

Le fichier **name** contient le nom du périphérique :

```

1 root@raspberrypi0-wifi:~# cat /sys/bus/iio/devices/iio\:device0/name
2 mpu-9250

```

Axes	Accéléromètre	Gyroscope	Température
x	in_accel_x_raw	in_anglvel_x_raw	in_temp_input
y	in_accel_y_raw	in_anglvel_y_raw	
z	in_accel_z_raw	in_anglvel_z_raw	

FIGURE 4.1 – Tableau d'explication des fichiers d'interface du driver de l'IMU

4.2 Calculs de la position de la centrale inertielles dans l'espace

Pour l'instant, les calculs sont réalisés avec l'accéléromètre uniquement.



FIGURE 4.2 – Photo du module IMU utilisé pour les tests

La taille des registres de données de l'accéléromètre est de 16-bits pour chaque axe.

4.2.1 Calcul d'angle entre deux vecteurs

Soit les coordonnées X et Y correspondants aux données de l'accéléromètre.

Calcul du rayon :

$$r = \sqrt{x^2 + y^2}$$

Calcul de l'angle :

$$\alpha = \arccos\left(\frac{x}{r}\right) \times \frac{180}{\pi}$$

Par exemple, posons $x = 512$ et $y = 854$:

Calcul du rayon :

$$r = \sqrt{512^2 + 854^2} = 995,7$$

Calcul de l'angle :

$$\alpha = \arccos\left(\frac{x}{r}\right) \times \frac{180}{\pi} = 59^\circ$$