D3R3 – Rapport d'avant-projet

Automatisation du placement d'une parabole

Lise CHAUVIN

Giles DEVILLERS

Thomas GRAGEON

Alexandre MINOT

Coach : Vincent GRIMAUD

Groupe RoboTic 1

E. Com : Véronique AUGER

Groupe de projet n°304

Client : Christophe TAILLIEZ

Sommaire

Introduction

Dans le cadre du projet tutoré de deuxième année, nous travaillons en collaboration avec Strategic Telecom Sécurité civile.

Notre objectif est d'automatiser le positionnement d'une parabole située sur un camion censé se déplacer pendant des interventions. Son positionnement est actuellement effectué manuellement et il nous a été demandé d'y remédier à l'aide d'un rotor et d'un vérin.

Il nous faudra choisir un vérin adapté qui répondra au cahier des charges ainsi qu'une carte d'interface nous permettant de le piloter, afin de régler la parabole en site. Il nous faudra également piloter le rotor pour régler la parabole en azimut. Pour nous connecter au satellite, il faudra connaître son type, qui dépend de la position géographique du camion. Une interface graphique sera codée sur le Raspberry Pi pour informer l'utilisateur de plusieurs paramètres et lui permettre de régler la parabole manuellement.

Dans un premier temps, nous allons présenter le budget et la répartition du travail, nous discuterons ensuite des solutions techniques envisagées pour répondre au cahier des charges.

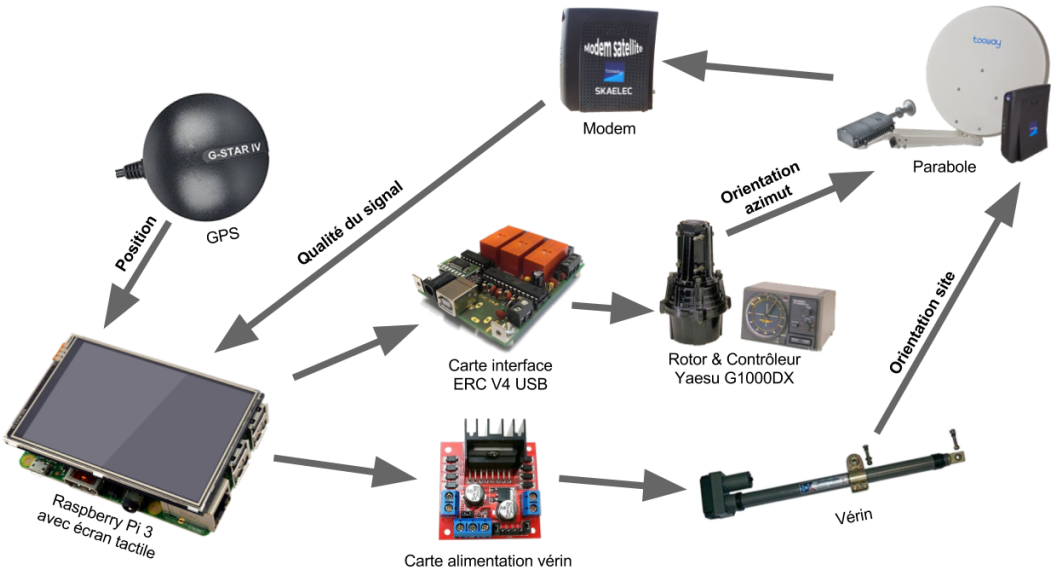
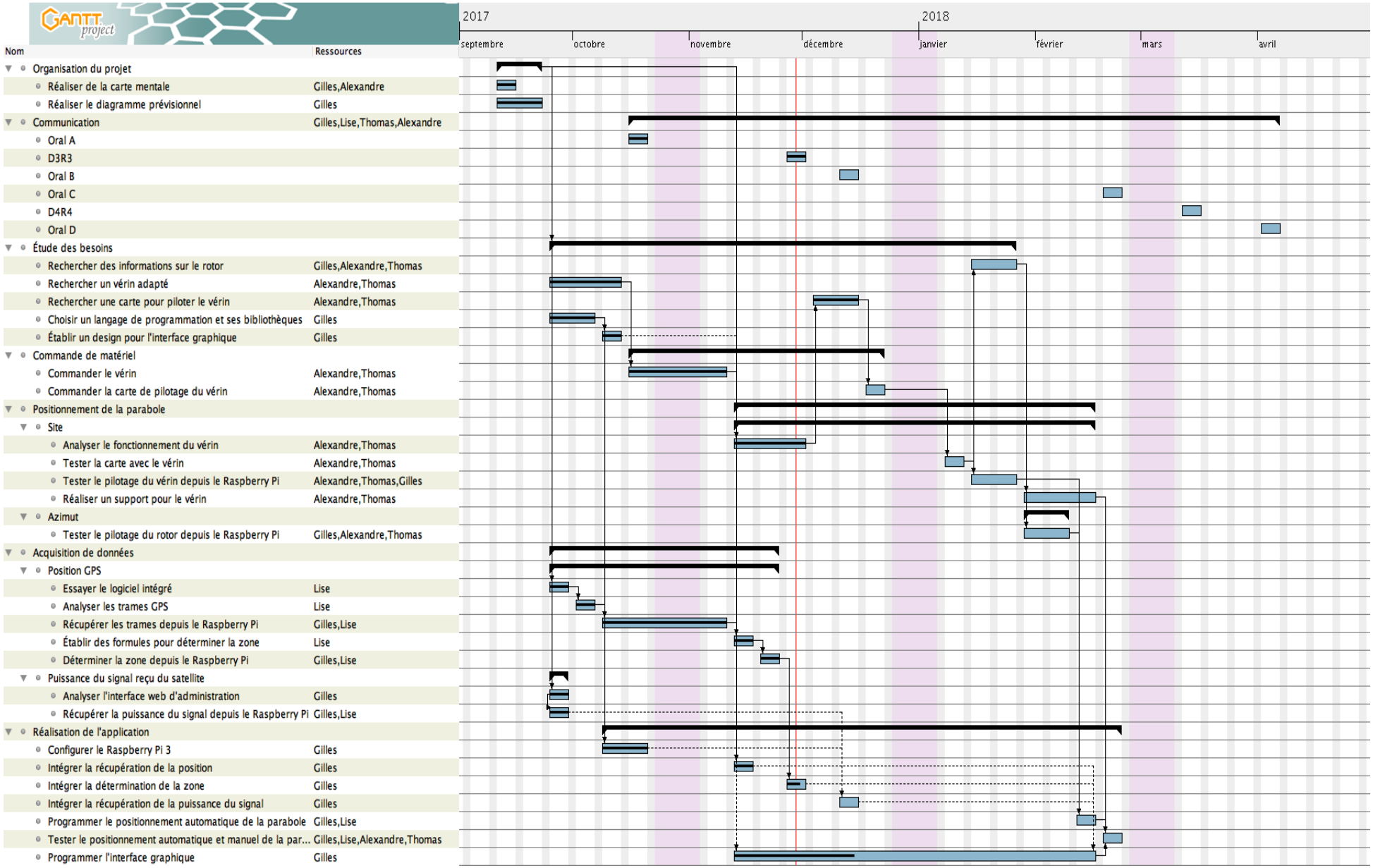


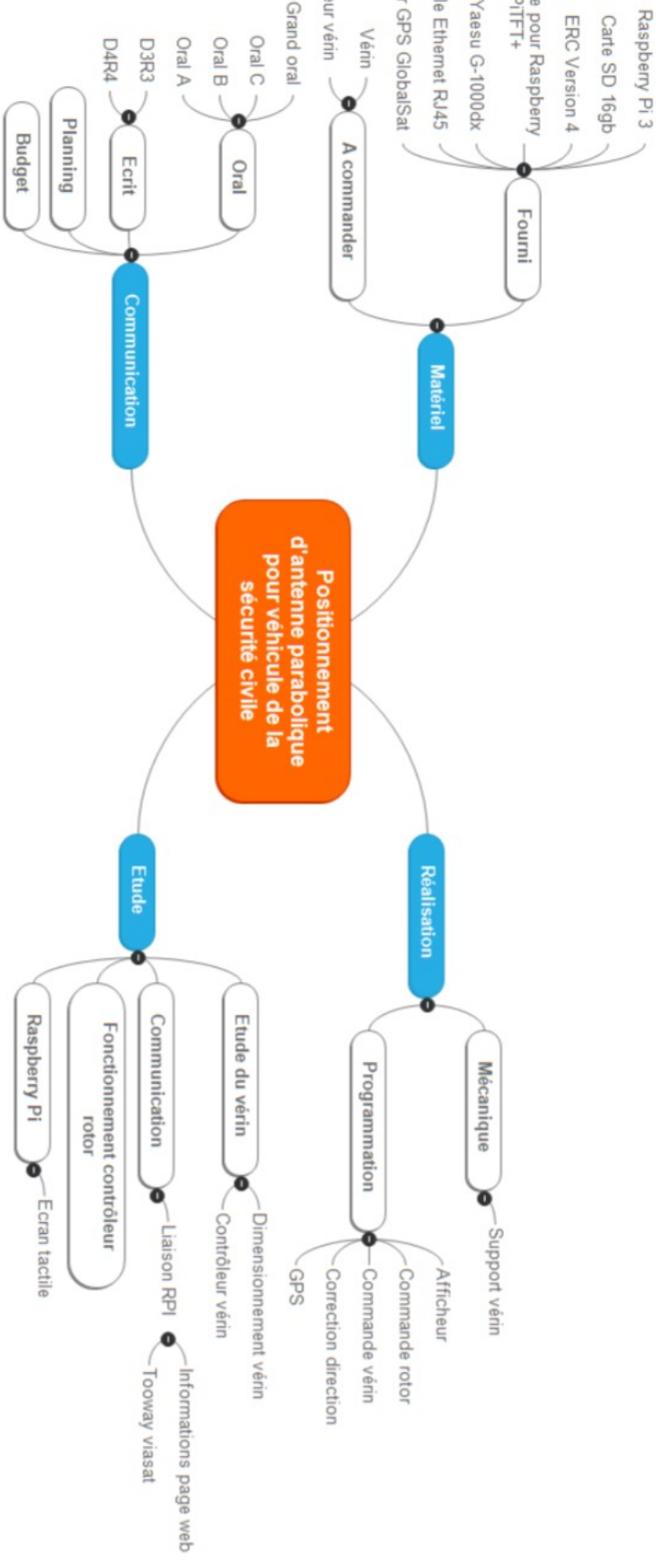
Illustration 1 : Structure du projet

1. Diagramme de Gantt



Nom	Ressources
Organisation du projet	
Réaliser de la carte mentale	Gilles,Alexandre
Réaliser le diagramme prévisionnel	Gilles
Communication	Gilles,Lise,Thomas,Alexandre
Oral A	
D3R3	
Oral B	
Oral C	
D4R4	
Oral D	
Étude des besoins	
Rechercher des informations sur le rotor	Gilles,Alexandre,Thomas
Rechercher un vérin adapté	Alexandre,Thomas
Rechercher une carte pour piloter le vérin	Alexandre,Thomas
Choisir un langage de programmation et ses bibliothèques	Gilles
Établir un design pour l'interface graphique	Gilles
Commande de matériel	
Commander le vérin	Alexandre,Thomas
Commander la carte de pilotage du vérin	Alexandre,Thomas
Positionnement de la parabole	
Site	
Analyser le fonctionnement du vérin	Alexandre,Thomas
Tester la carte avec le vérin	Alexandre,Thomas
Tester le pilotage du vérin depuis le Raspberry Pi	Alexandre,Thomas,Gilles
Réaliser un support pour le vérin	Alexandre,Thomas
Azimut	
Tester le pilotage du rotor depuis le Raspberry Pi	Gilles,Alexandre,Thomas
Acquisition de données	
Position GPS	
Essayer le logiciel intégré	Lise
Analyser les trames GPS	Lise
Récupérer les trames depuis le Raspberry Pi	Gilles,Lise
Établir des formules pour déterminer la zone	Lise
Déterminer la zone depuis le Raspberry Pi	Gilles,Lise
Puissance du signal reçu du satellite	
Analyser l'interface web d'administration	Gilles
Récupérer la puissance du signal depuis le Raspberry Pi	Gilles,Lise
Réalisation de l'application	
Configurer le Raspberry Pi 3	Gilles
Intégrer la récupération de la position	Gilles
Intégrer la détermination de la zone	Gilles
Intégrer la récupération de la puissance du signal	Gilles
Programmer le positionnement automatique de la parabole	Gilles,Lise
Tester le positionnement automatique et manuel de la par...	Gilles,Lise,Alexandre,Thomas
Programmer l'interface graphique	Gilles

2. Carte mentale



3. Budget

4. Partie mécanique

Pour pouvoir orienter la parabole en site, il nous faut un vérin possédant des caractéristiques bien précises.

A. Critères de sélection du vérin

- Le vérin doit avoir une force de poussée suffisante pour pouvoir soulever la parabole d'environ 15 kilogrammes. La force réelle de poussée reste inconnue en l'absence du matériel nécessaire à la mesure de cette force en Newton-mètre (N.m). De plus, cette varie en fonction de l'endroit de la poussée. En effet, plus la poussée est près du centre, plus sa force est importante.
- Le vérin doit pouvoir fonctionner en étant alimenté en 12 ou 24 volts maximum, tension pouvant être fournie par le camion de la sécurité civile.
- Le vérin doit être positionnable. En effet, il est nécessaire de connaître sa position afin de l'orienter précisément.
- La taille du vérin peut être variable. En effet, la fixation du vérin sur la parabole va être réalisée par nos soins et adaptée à cette taille.

B. Critères du vérin sélectionné

Pour répondre aux critères ci-dessus, le vérin retenu est le Super Jack III de chez Jaeger. Les caractéristiques du vérin sont les suivantes :

- Taille : 12 pouces
- Charge statique : 225 kg
- Charge dynamique : 135 kg
- Alimentation : 36 volts DC
- Précision du capteur : 76 impulsions par pouce
- Température de fonctionnement : - 30 °C à 50 °C

Illustration 2 : Photo du vérin choisi

Malgré une alimentation supérieure à celle mentionnée dans nos critères, ce vérin fonctionne en 24 volts, et même s'il perd en charge statique et dynamique, sa force reste malgré tout largement suffisante pour pouvoir orienter la parabole, qui ne pèse que 15 kg.

C. Tests effectués sur le vérin

Plusieurs tests ont été réalisés sur le vérin, dont un test en 24 volts et un test sur son capteur. Tous les tests ont été réalisés avec une alimentation stabilisée.

Le capteur utilisé dans ce vérin est un interrupteur à lames souples. Le contact est réalisé sous l'effet d'un champ magnétique. Lorsque le moteur tourne, il fait tourner un aimant qui active l'interrupteur à chaque tour qu'il fait et laisse passer le signal que l'on envoie sur l'entrée « sensor » du vérin.

De plus, nous avons vérifié les caractéristiques du capteur : nous avons bien les 76 impulsions par pouce. Donc, une impulsion représente 3,34 mm.

Ce vérin ne peut pas être commandé directement par le Raspberry Pi. C'est pour cela qu'une carte d'interface est obligatoire. Cette carte n'a pas encore été commandée.

D. Carte d'interface du vérin

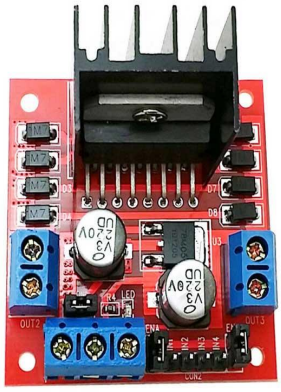


Illustration 3 : Carte de commande du vérin L298N

Le choix de la carte d'interface s'est porté sur le L298N.
Ses caractéristiques sont les suivantes :

- Tension de conduite : 5 à 36 volts
- Fournit jusqu'à 2 A
- Tension de commande compatible avec le Raspberry Pi

Branchements :

BROCHE	FONCTION
OUT1, OUT2	Branchement moteur 1

OUT3, OUT4	Branchement moteur 2
IN1, IN2	Choix de polarité du moteur 1
IN3, IN4	Choix de polarité du moteur 2
ENA	Signal PWM pour la vitesse du moteur 1
ENB	Signal PWM pour la vitesse du moteur 2
Entrée + et GND	Alimentation des moteurs
5 V	Sortie 5 V

Cette carte d'interface correspond parfaitement à nos besoins puisque le L298 gère l'inversion de polarité et du fait de sa tension de commande. Pour la contrôler avec le Raspberry Pi, il faudra envoyer un signal soit sur IN1 soit sur IN2 pour choisir le sens de rotation du moteur (on choisira ainsi de faire sortir ou rentrer le vérin), il faut ensuite envoyer un signal sur ENA pour le faire fonctionner.

E. Rotor

Le rotor qui nous est fourni est un YAESU G-1000DX, avec les caractéristiques suivantes :

- Charge statique : 6000kg/cm
- Charge dynamique : 600 à 1100 kg/cm
- Temps d'une rotation à 360 ° : 43 à 93 secondes
- Diamètre : 186 mm
- Hauteur : 300 mm
- Charge verticale : 200 kg

Illustration 4: Rotor Yaesu

Pour piloter le rotor, nous devrons utiliser une carte d'interface comme pour le vérin. Celle qui nous est fournie est un ERC Version 4, elle communiquera avec le Raspberry Pi par USB. Il faudra envoyer la commande via le protocole YAESU GS-232A ou GS-232B à la carte.

5. Récupération de la position GPS du camion

Afin de déterminer à quel satellite nous allons nous connecter, il faut connaître la position géographique du camion de la sécurité civile. Pour cela, un récepteur GPS nous a été fourni.

A. Fonctionnement du récepteur GPS

Le récepteur que nous a fourni la Sécurité Civile est une tête GPS G-Star IV de chez GlobalSat. Cette tête se branche en USB à un ordinateur auquel elle envoie des trames selon le protocole de communication standardisé NMEA 0183 (National Marine Electronics Association). Ces trames sont les suivantes :

- Trame GGA : fournit l'heure, les coordonnées en longitude et latitude, l'altitude, le nombre de satellites trouvés, la précision de la mesure
- Trame GLL : fournit l'heure, la longitude, la latitude
- Trame GSA : indique le nombre de satellites trouvés et la précision de la mesure
- Trame GSV : indique le nombre de satellites trouvés, la qualité du signal de chaque satellite, son azimut et son élévation
- Trame VTG : indique la direction et la vitesse du récepteur GPS

B. Solutions envisagées

Désirant connaître la longitude et la latitude du camion, nous avons le choix entre les trames GGA et GLL. Dans l'éventualité où nous voudrions connaître l'altitude du camion, nous envisageons d'exploiter la trame GGA.

Une fois les coordonnées récupérées, nous pourrons localiser assez précisément le camion et déduire à quel satellite nous nous connecterons pour avoir une réception optimale.

Ci-contre une capture d'écran du site web qui cartographie les satellites en Europe. Nous nous en servirons pour obtenir les coordonnées des satellites et ainsi évaluer à quel satellite nous connecter.

Nous devrons donc réaliser un programme afin de déterminer à quel satellite nous connecter en fonction des coordonnées GPS associées au camion. Nous prévoyons d'utiliser le langage Python 3 pour faciliter la mise en commun des différents codes.

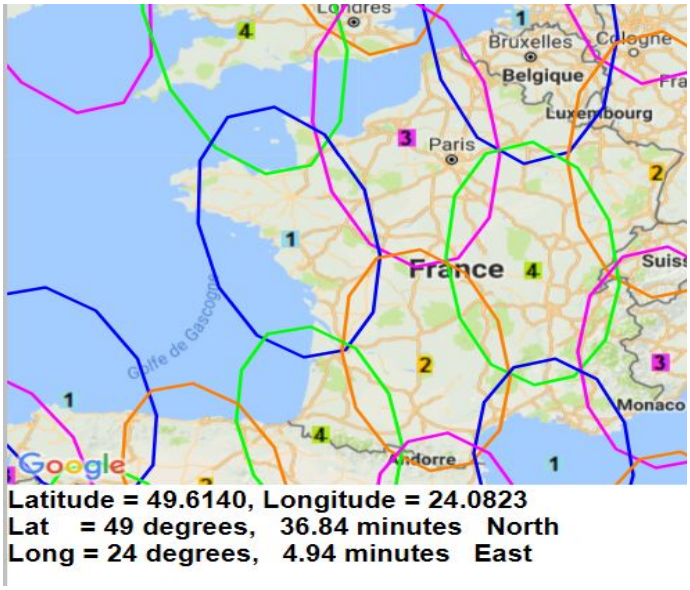


Illustration 5: Capture d'écran des zones de satellites en France

6. Le cœur du système

Afin de récupérer la position depuis le GPS pour ensuite l'exploiter, sélectionner le satellite approprié auquel se connecter et enfin orienter la parabole vers celui-ci, nous avons besoin d'un appareil programmable qui fera office de « *cerveau* », contrôlant toute l'installation.

Un Raspberry Pi 3 nous a été imposé pour réaliser cette tâche. Il s'agit d'un ordinateur embarqué compact, puissant et peu cher. De plus, un écran LCD tactile y sera attaché et permettra de contrôler le positionnement de la parabole ainsi que de vérifier l'état du système de façon simple et rapide.

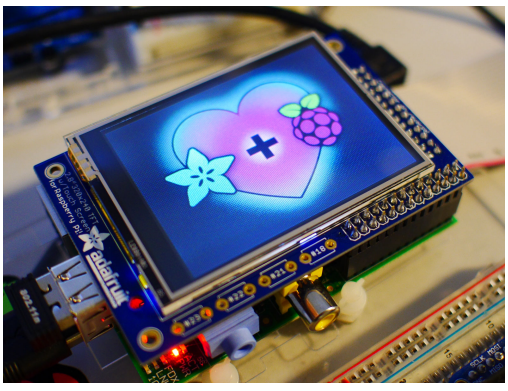


Illustration 6: Écran tactile monté sur le Raspberry Pi

Illustration 7: Raspberry Pi 3

Le GPS sera branché sur un port USB du Raspberry Pi 3, nous y avons accès via une connexion série standard. La communication avec la carte de commande du rotor et du vérin se fera par les GPIO (General Purpose Input/Output), à savoir des broches programmables individuellement afin d'envoyer des impulsions de commande.

Pour la programmation de notre application, nous avons décidé d'utiliser le langage de programmation **Python**. C'est un langage adapté à notre application, car il est simple d'utilisation, possède beaucoup de bibliothèques et permet de développer rapidement.

Pour réaliser l'interface graphique, nous avons choisi la bibliothèque **Pygame**, très populaire, bien documentée et simple d'utilisation.

Cette interface affichera des informations récupérées depuis le GPS telles que la position ou le nombre de satellites connectés. Certaines informations affichées proviendront du modem Tooway de la parabole comme la puissance du signal ou le niveau de bruit. Des boutons permettront de lancer un positionnement automatique de la parabole ou au contraire d'activer un mode de contrôle manuel de la position de la parabole.

En plus des premiers essais de réalisation de l'interface graphique, nous avons réussi à communiquer avec le *GPS* avec succès. Nous utilisons pour cela la bibliothèque standard **pySerial**. La bibliothèque **pynmea2** facilite l'exploitation des *trames* de données et nous permet d'extraire facilement les informations qui nous intéressent.

Enfin, notre application doit accéder à la page web de configuration du modem de la parabole. Celle-ci permet de configurer le satellite à utiliser en fonction de la position du camion et affiche la qualité du signal avec le satellite en question. La bibliothèque standard **Requests** nous permet d'effectuer les requêtes *HTTP* nécessaires à la communication avec cette page depuis notre application.

Bilan

À ce stade d'avancement du projet, le fonctionnement vérin est parfaitement compris et est prêt à être programmé. Grâce à la récupération des coordonnées GPS, la détection du satellite auquel se connecter a pu être réalisée. Les informations de la parabole ont pu être récupérées depuis la page de configuration du modem. De plus, la programmation de l'interface graphique a déjà débuté.

Par conséquent, il nous reste à effectuer la commande du vérin, à comprendre et à commander le rotor, ainsi qu'à fixer le vérin sur la parabole. Nous devrons aussi finir de réaliser l'interface graphique puis piloter le rotor et le vérin pour positionner la parabole en azimut et en élévation.

Conclusion

Cet avant-projet nous a permis d'apprendre à choisir un vérin électrique qui correspond à des critères précis afin de répondre au cahier des charges. Il nous a également été nécessaire de nous adapter, notamment en choisissant le langage de programmation Python, qui n'était pas connu de tous les membres du groupe. Nous avons aussi pu nous familiariser avec le Raspberry Pi 3 et plus particulièrement avec son écran tactile. Cet avant-projet a renforcé notre compréhension des liaisons série grâce à la lecture des trames fournies par le récepteur GPS. Nous avons aussi appris grâce aux différents oraux et écrits à gérer notre temps et à présenter nos travaux devant un client.

Index des illustrations