12/12/2017

Projet tutoré - Oral B

Automatisation du positionnement d'une parabole de réception

Lise **Chauvin**
Gilles **Devillers**
Thomas **Grageon**
Alexandre **Minot**

2ème année DUT GEII
Groupe Robotic 1

Coach : Vincent **Grimaud**

Référent professionnel : Christophe **Tailliez**

Sommaire

- Introduction
- Structure du système
- Carte mentale
- Diagramme prévisionnel
- Budget actuel
- Travail réalisé
- Conclusion

Introduction

La camionnette de la **Sécurité Civile** sur laquelle nous allons travailler est équipée d'une parabole **Tooway** qui permet d'accéder à **internet** par **satellite**.

Le **positionnement** de la parabole s'effectue **manuellement** en montant sur le toit de la camionnette et en écoutant les « bips » émis par celle-ci en fonction de la **qualité** de réception du signal satellite.

Notre travail est d'automatiser le positionnement de cette parabole afin d'obtenir **facilement** et **rapidement** le **meilleur** signal possible.

Structure du système

GPS

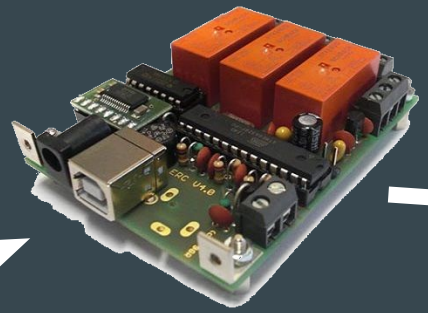
Modem

Parabole

Position

Qualité du signal

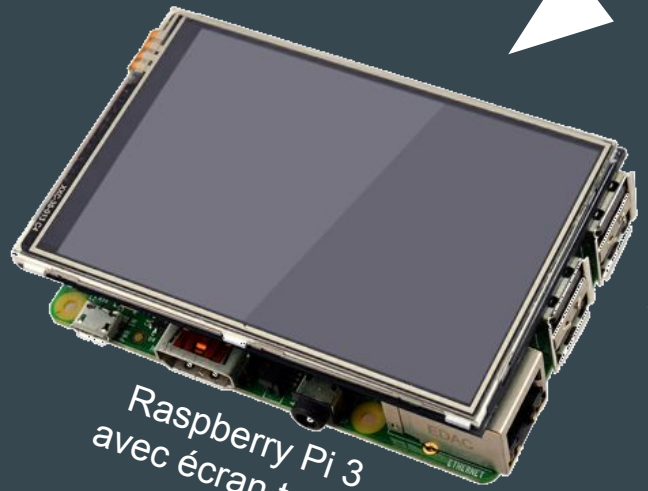
Orientation azimut



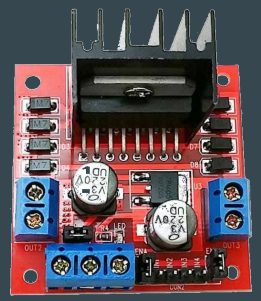
Carte interface ERC V4 USB

Rotor & Contrôleur Yaesu G1000DX

Orientation site



Raspberry Pi 3 avec écran tactile



Carte alimentation vérin

Vérin

Carte mentale

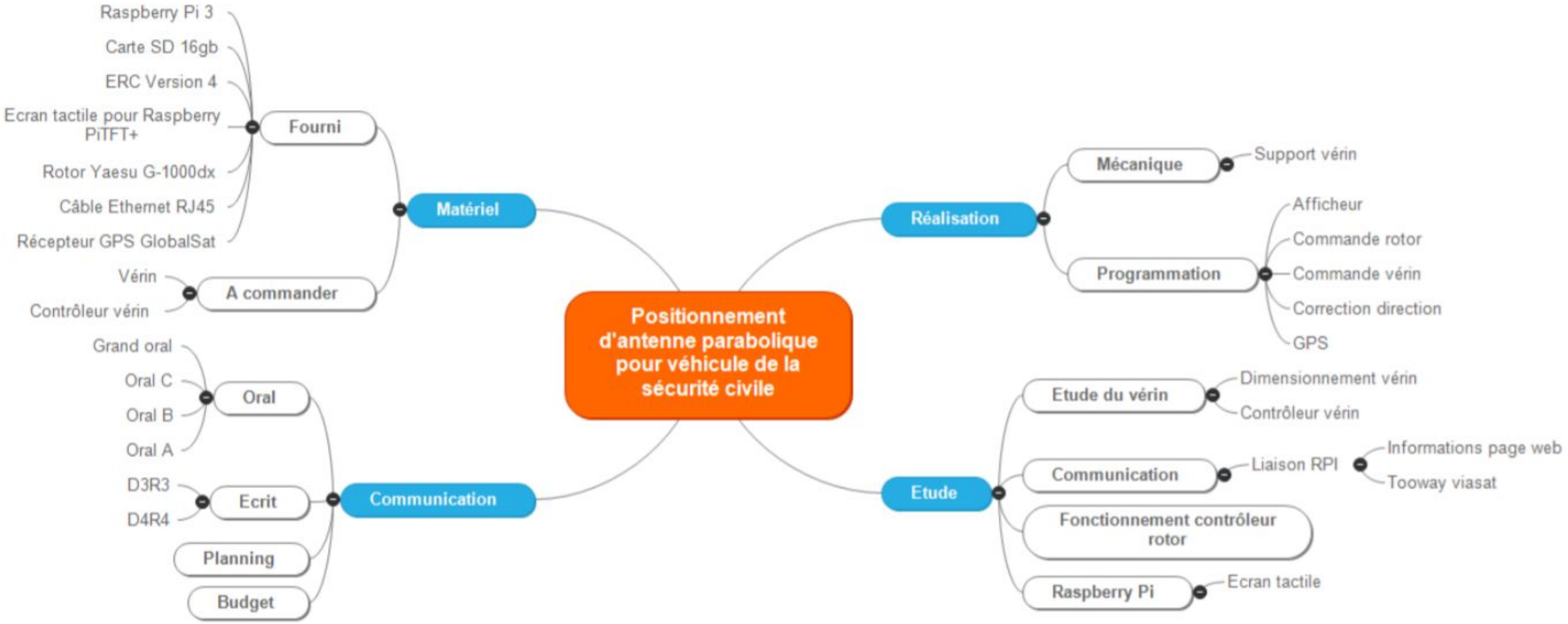
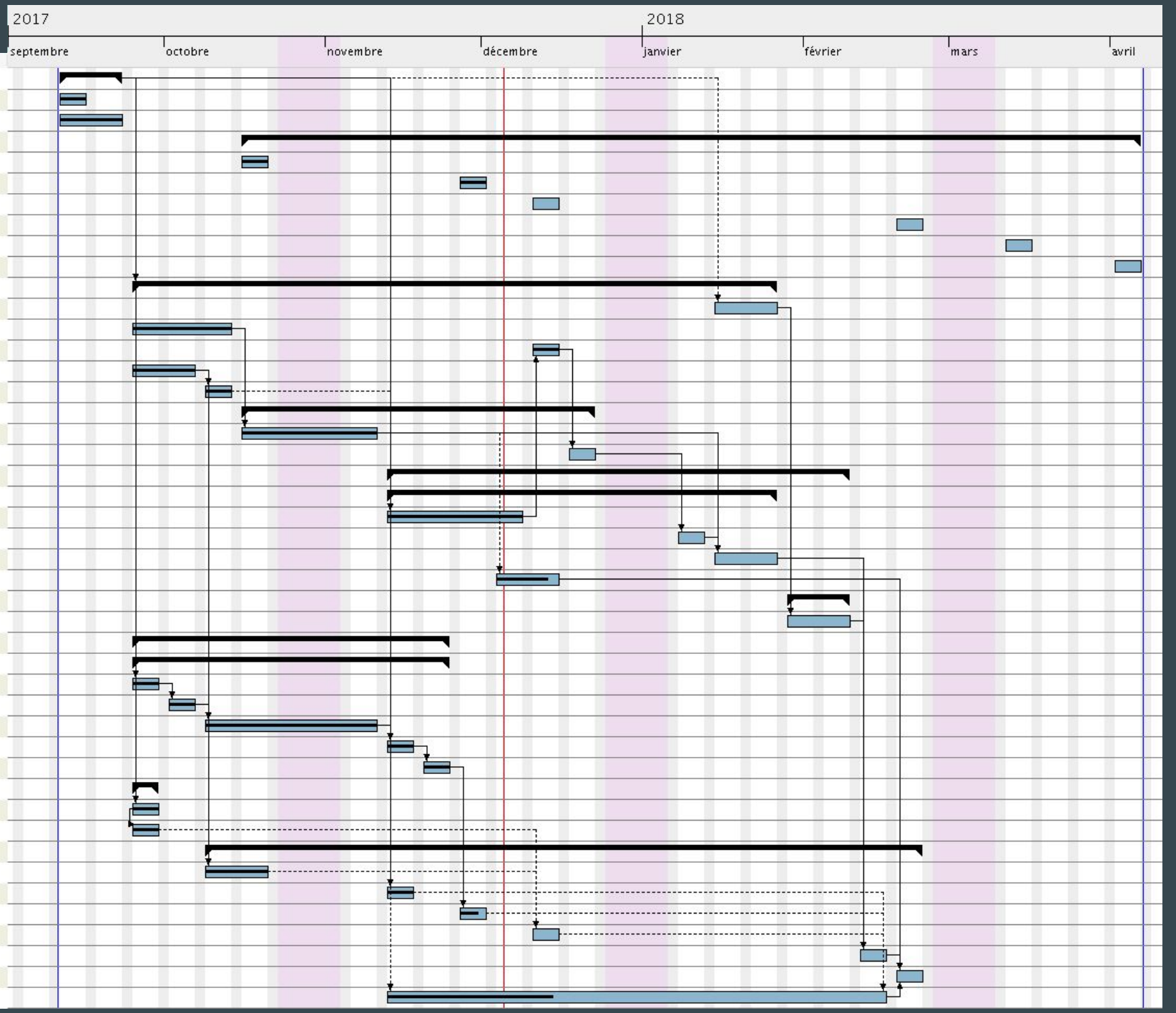


Diagramme prévisionnel

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,Thomas,Alexandre
Rechercher un vérin adapté	Thomas,Alexandre
Rechercher une carte pour piloter le vérin	Thomas,Alexandre
Choisir un langage de programmation et ses bibliothèques	Gilles
Établir un design pour l'interface graphique	Gilles
Commande de matériel	
Commander le vérin	Thomas,Alexandre
Commander la carte de pilotage du vérin	Thomas,Alexandre
Positionnement de la parabole	
Site	
Tester et analyser le fonctionnement du vérin	Thomas,Alexandre
Tester le vérin avec sa carte de pilotage	Thomas,Alexandre
Tester le pilotage du vérin depuis le Raspberry Pi	Gilles,Thomas,Alexandre
Réaliser un support pour le vérin	Thomas,Alexandre
Azimut	
Tester le pilotage du rotor depuis le Raspberry Pi	Gilles,Thomas,Alexandre
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,Thomas,Alexandre
Programmer l'interface graphique	Gilles



Répartition des tâches

Tâches	Lise **Chauvin**	Gilles **Devillers**	Thomas **Grageon**	Alexandre **Minot**
Récupération de la qualité du signal	X	X		
Essais du GPS	X			
Mesures et dimensionnement du vérin			X	X
Configuration du Raspberry Pi		X		
Programmation du GPS	X			
Programmation de l'interface graphique		X		
Commande du rotor et du vérin			X	X
Câblage			X	X
Programmation finale	X	X	X	X

Budget actuel

Désignation	Prix TTC
Raspberry Pi 3	35,00 €
Alimentation Raspberry Pi	10,00 €
Carte MicroSD Kingston 16 Go	8,00 €
ERC V4 USB	110,00 €
Écran tactile TFT 3,5”	37,00 €
Rotor Yaesu G-1000DX & Contrôleur	490,00 €
Câble Ethernet RJ45	5,00 €
Kit satellite Tooway (parabole, modem, support, activation)	375,00 €
Abonnement Tooway 25	120 €/mois
Vérin	45,00 €
L298N	6,00 €
Total	**1541 €**

Travail réalisé

Récupération de la puissance du signal du satellite

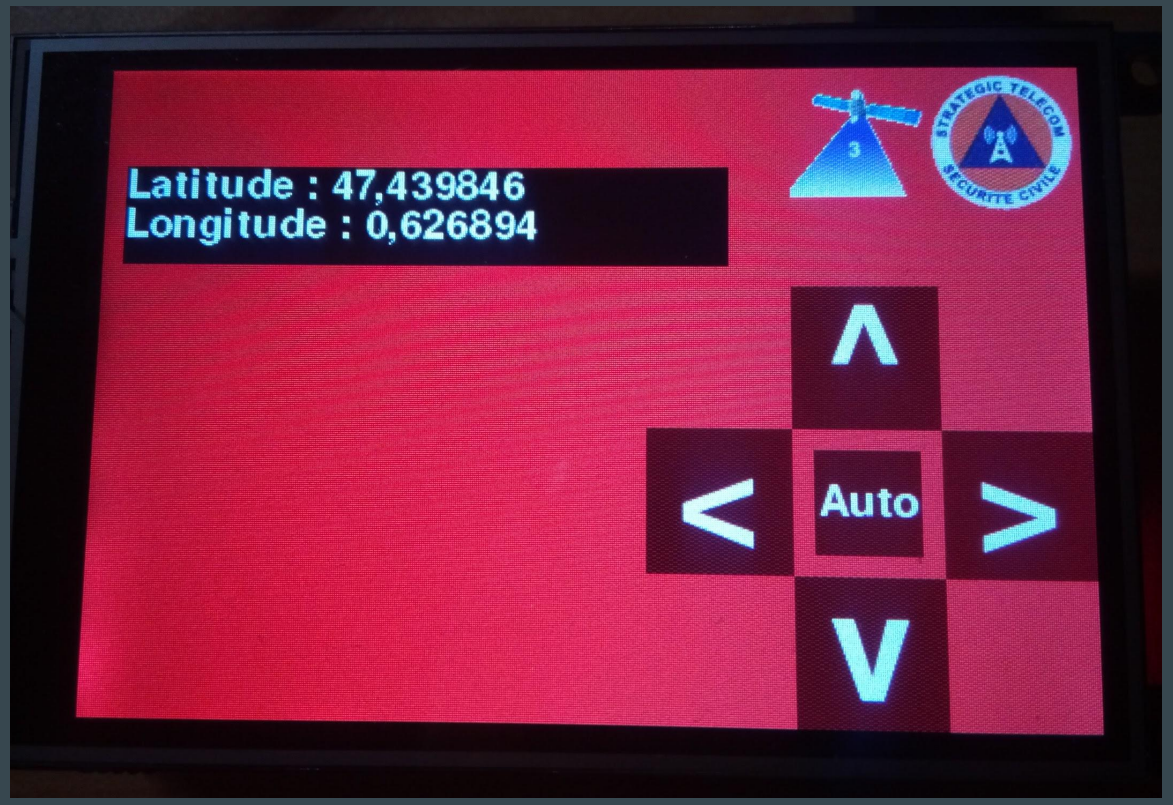
- Programme en Python
- Bibliothèque Requests pour les requêtes HTTP
- Puissance du signal reçu et bruit accessibles depuis une page web hébergée sur le routeur

```python
import requests
from requests_toolbelt.utils import dump

data = []
def getData():
    global data

    r    = requests.get("http://192.168.100.1/index.cgi?page=modemStatusData")
    data = r.text.split("##")

getData()
rxPower = data[14]
rxSNR   = data[11]

print("Rx Power : " + rxPower + " dBm\tRx SNR : " + rxSNR + " dB")
```

Travail réalisé

Réalisation de l'interface graphique

- Bibliothèque Pygame
- Design simple à un ou deux écrans
- Potentiellement une bibliothèque GUI en plus



Informations affichées :

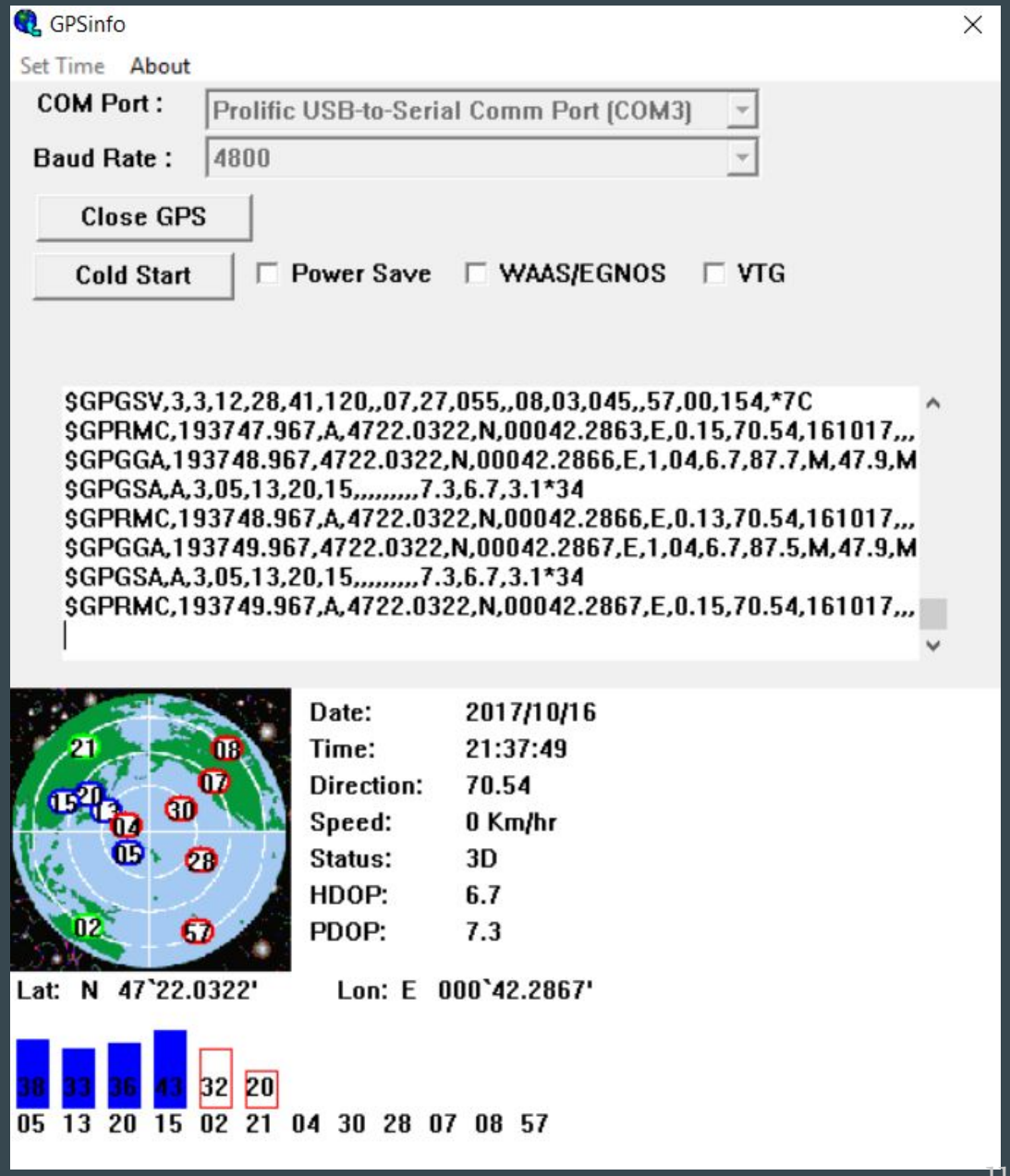
- Position (latitude & longitude)
- Nombre de satellites connectés
- Zone dans laquelle l'utilisateur est situé (1 à 4)
- Qualité de la communication avec le satellite

Contrôles :

- Démarrage du positionnement automatique
- Positionnement manuel de la parabole
- Sélection manuel de la zone

Travail réalisé

Analyse des trames GPS NMEA



Travail réalisé

Récupération des coordonnées GPS

- Connexion du GPS au Raspberry Pi par USB
- Communication série 4800 baud
- Bibliothèque Pyserial pour communiquer avec le GPS
- Bibliothèque Pynmea2 pour décoder et exploiter les trames NMEA

```
3  import serial
4  import pynmea2
5  import string
6
7  gps = serial.Serial('/dev/ttyUSB0', 4800)
8  print(gps.name)
9
10 while True:
11     nmea = gps.readline()
12
13     if nmea[0:6] == "$GPGGA":
14         gpsData = pynmea2.parse(nmea)
15         print("Latitude : " + "%02d°%02d'%07.4f'" % (gpsData.latitude, gpsData.latitude_minutes, gpsData.latitude_seconds))
16         print("Longitude : " + "%02d°%02d'%07.4f'" % (gpsData.longitude, gpsData.longitude_minutes, gpsData.longitude_seconds))
17         print("Fix" if gpsData.gps_qual else "No fix")
18         print("Sats : " + gpsData.num_sats)
19         print("")
20
21 gps.close()
22
```

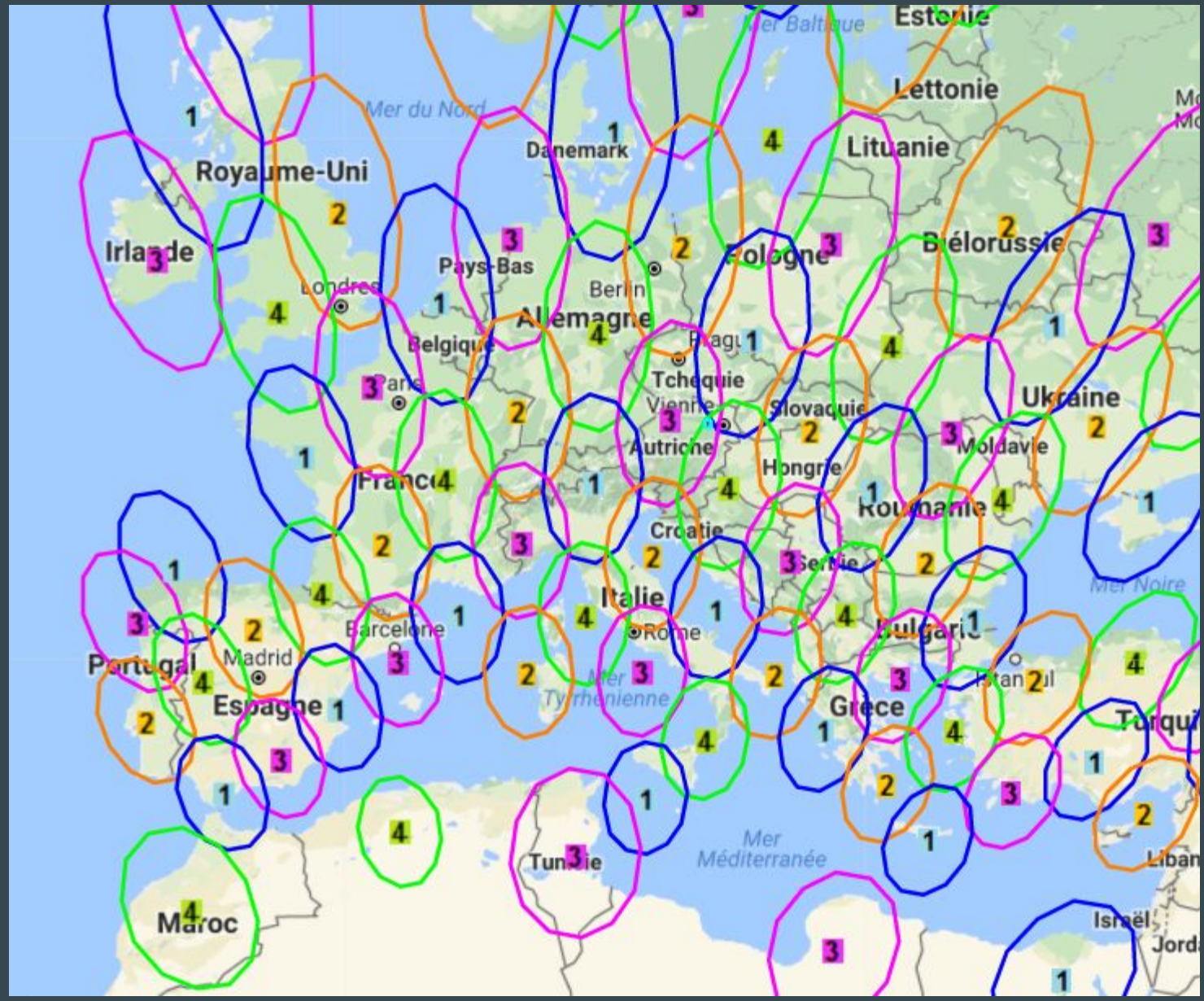
Travail réalisé

Détermination de la zone

Carte des satellites Ka-Sat en Europe

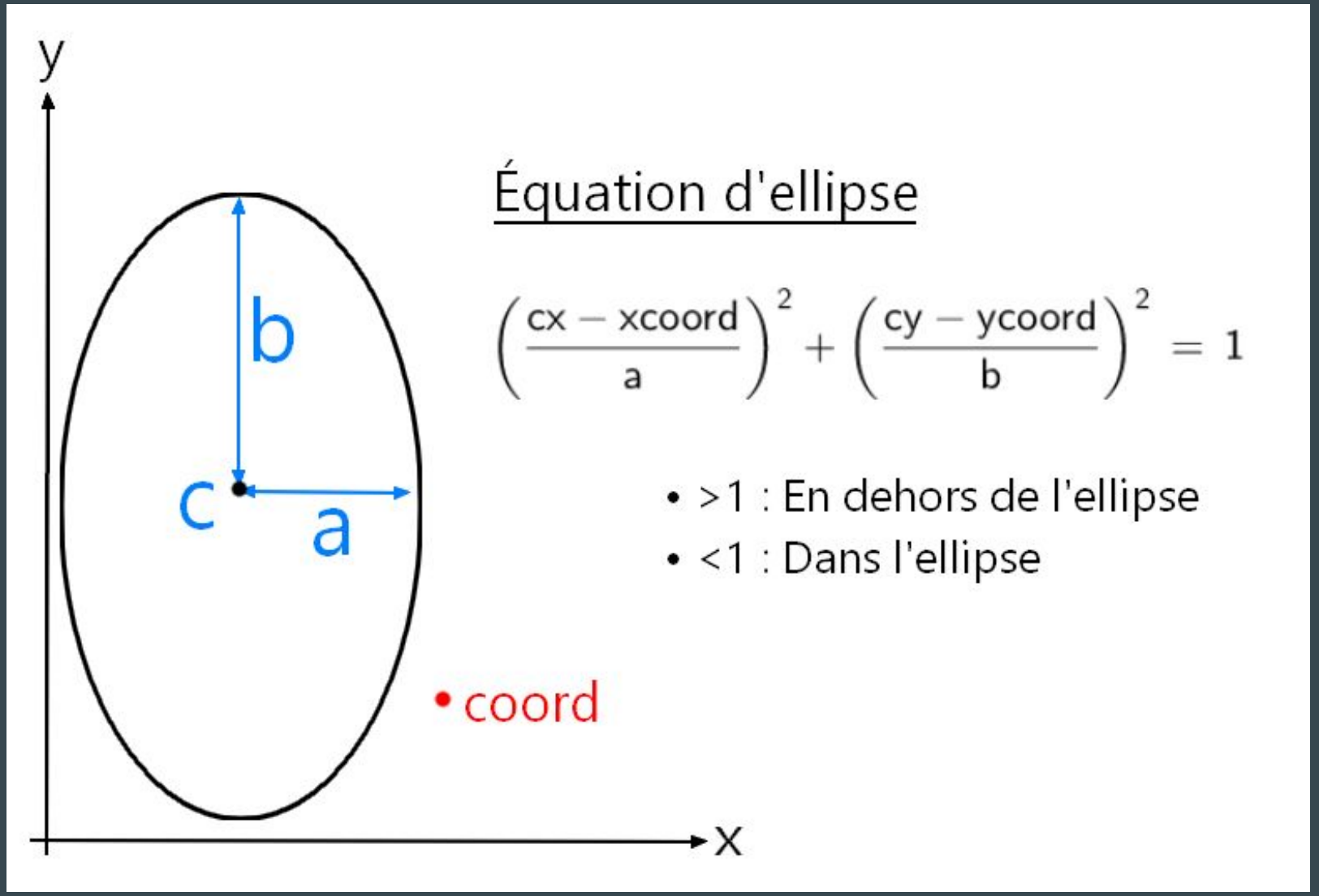
- 4 Zones
- 82 Spots
- Identifier le spot et la zone

Capture d'écran du site satsig.net



Travail réalisé

Modèle géométrique de l'ellipse



Travail réalisé

Démonstration visuelle

$C = (2, 40)$

$a : \left(\frac{2 - x}{2.5}\right)^2 + \left(\frac{40 - y}{4}\right)^2 = 1$

$Sat = (1, 41)$

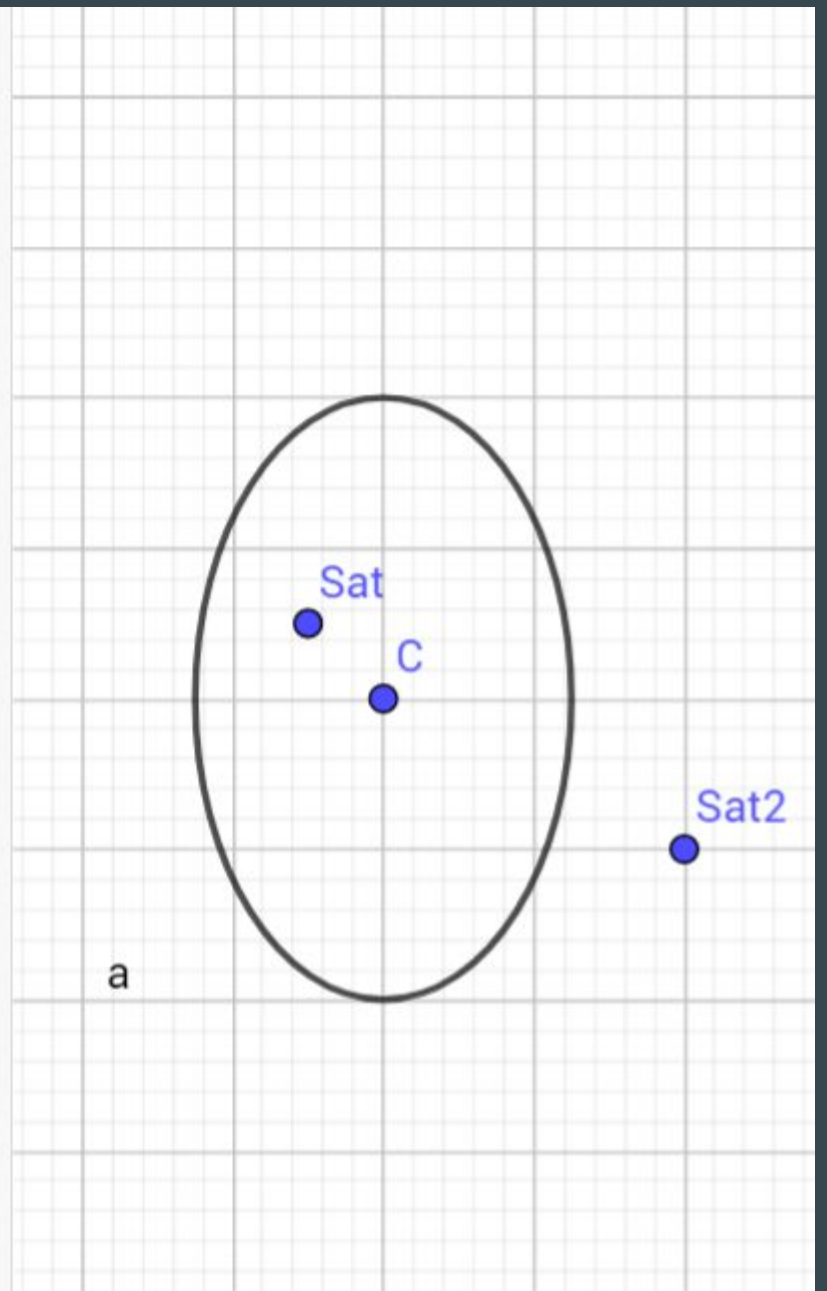
$b = \left(\frac{2 - 1}{2.5}\right)^2 + \left(\frac{40 - 41}{4}\right)^2$

$\rightarrow 0.22$ > 1 donc dans l'ellipse

$Sat2 = (6, 38)$

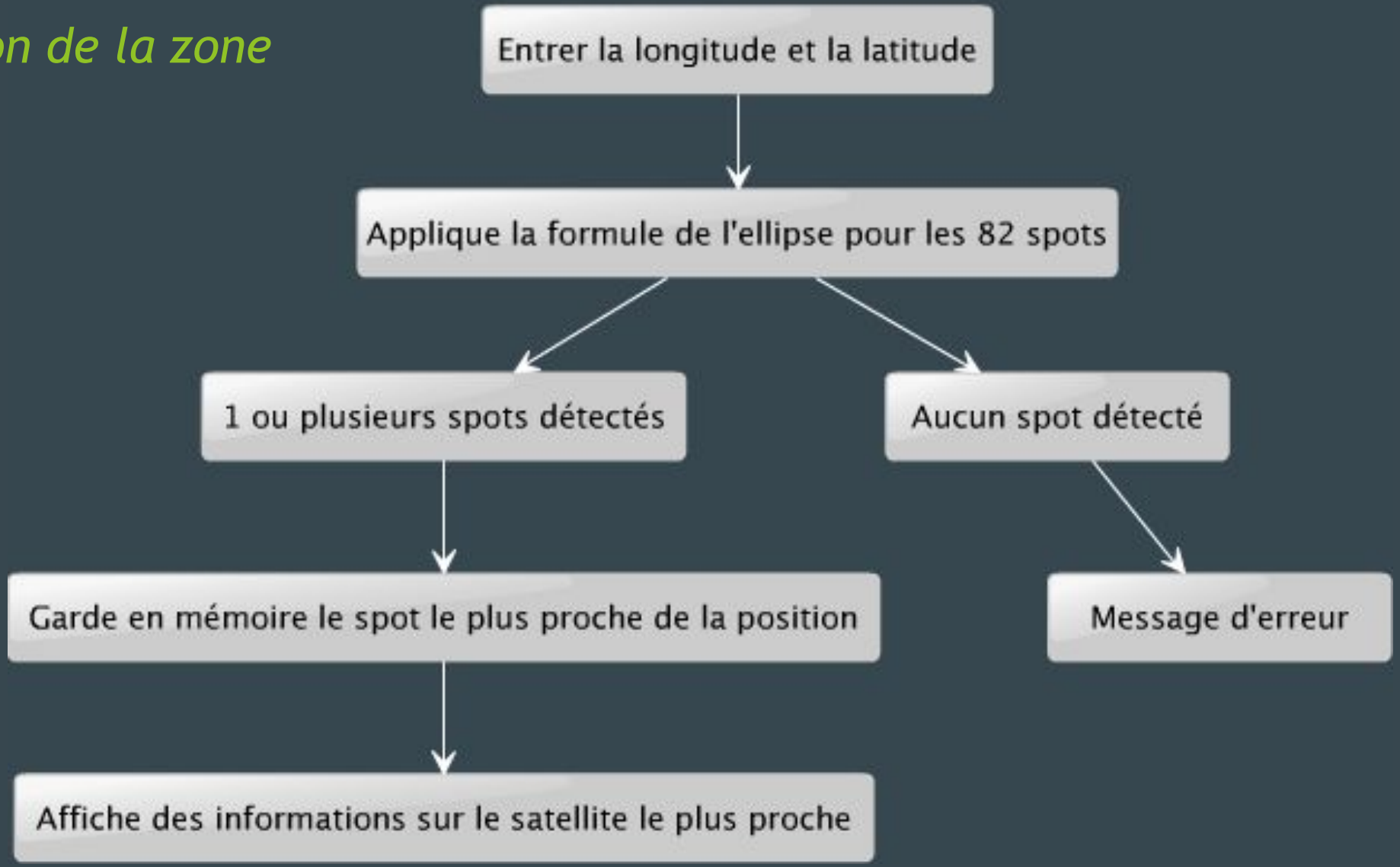
$c = \left(\frac{2 - 6}{2.5}\right)^2 + \left(\frac{40 - 38}{4}\right)^2$

≈ 2.81 < 1 donc en dehors de l'ellipse



Travail réalisé

Détection de la zone

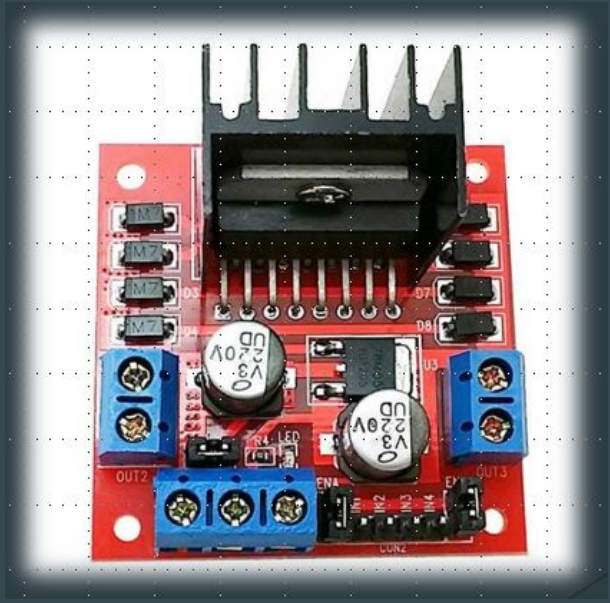


Travail réalisé

Vérin superjack III

Caractéristiques
- 12 Pouces
- Charge statique: 225 kg
- Charge dynamique: 135 kg
- 76 impulsions par pouces
- Alimenté en 36 volts

Carte interface vérin: L298



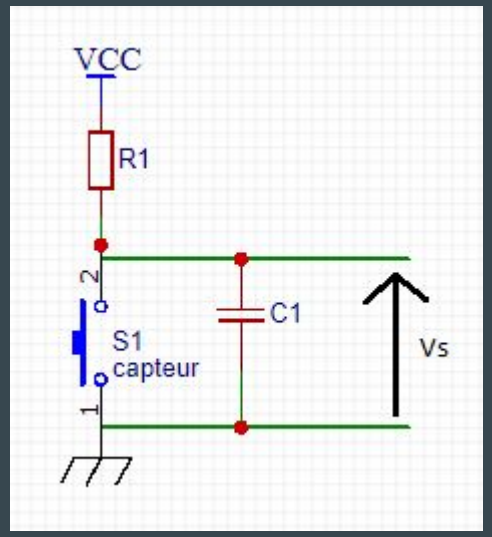
Caractéristiques:
Tension de conduction: 5V à 35V
Fournis jusqu'à 2A

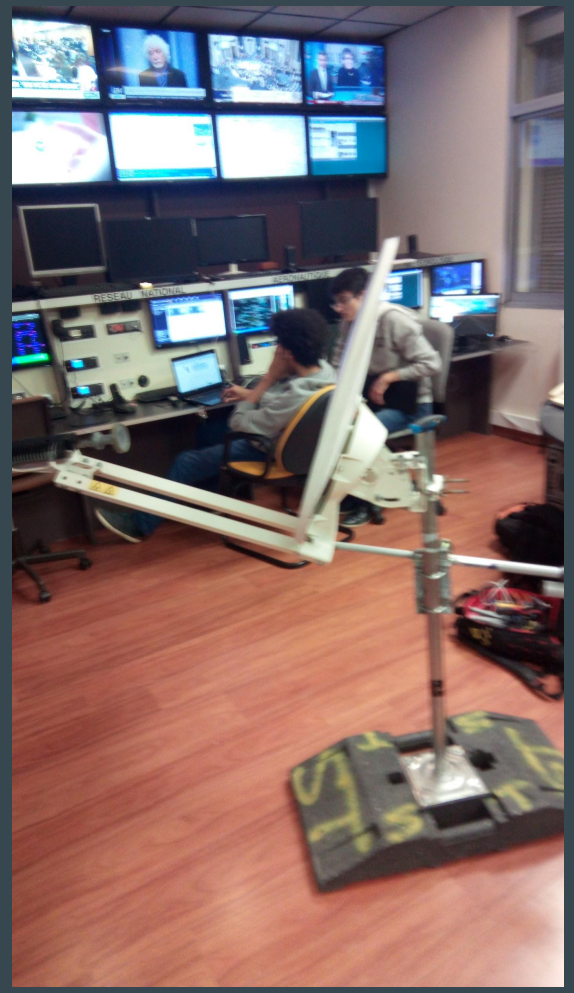
Travail réalisé

Essais du vérin

Tests réalisés :

- Sur le capteur de position
  - Capteur de type interrupteur à lame souple





- Sur le vérin

-Montée en 24V en ~2min

-Montée en 36V en ~1min

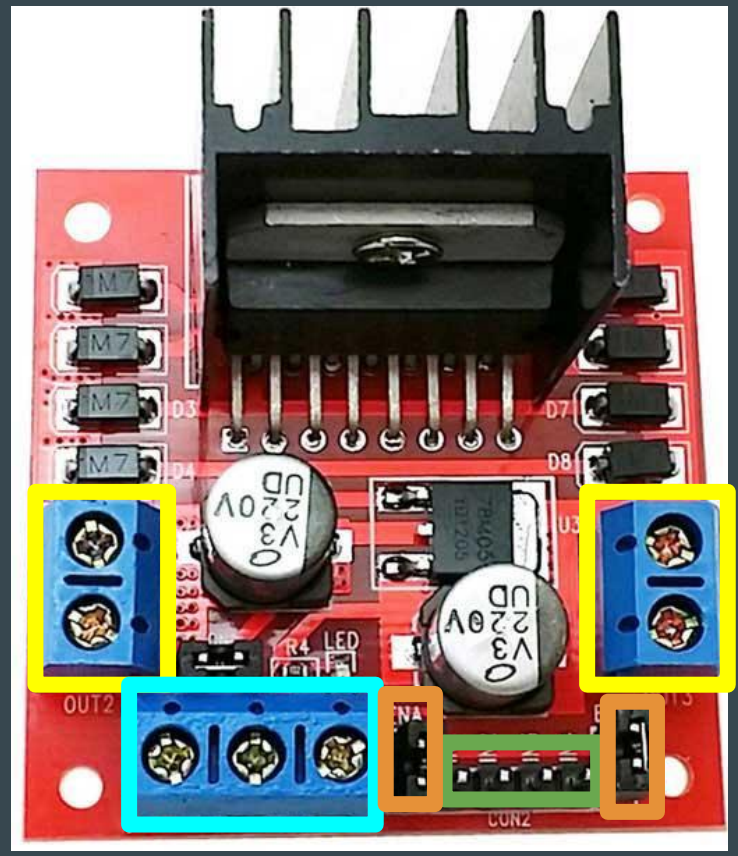
Travail réalisé

Essais du vérin

Travail réalisé

Étude de la carte de commande du vérin

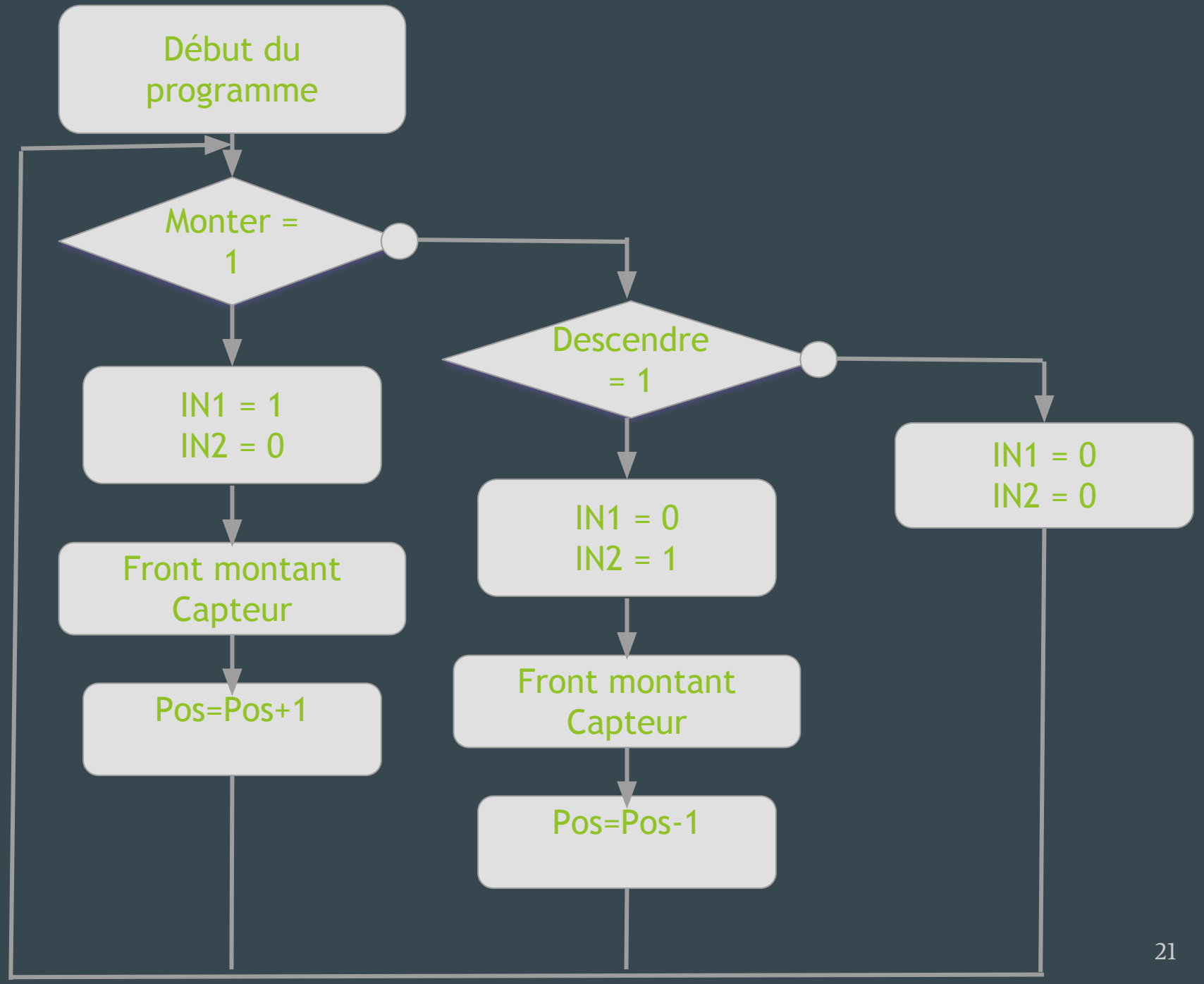
L298N



Broche	Fonction
OUT1, OUT2	Alimentation moteur 1
OUT3, OUT4	Alimentation moteur 2
IN1, IN2	Choix polarité du moteur 1
IN3, IN4	Choix polarité du moteur 2
ENA	Signal PWM de commande de vitesse moteur 1
ENB	Signal PWM de commande de vitesse moteur 2
+ et GND	Alimentation des moteur
5v	Sortie 5volt

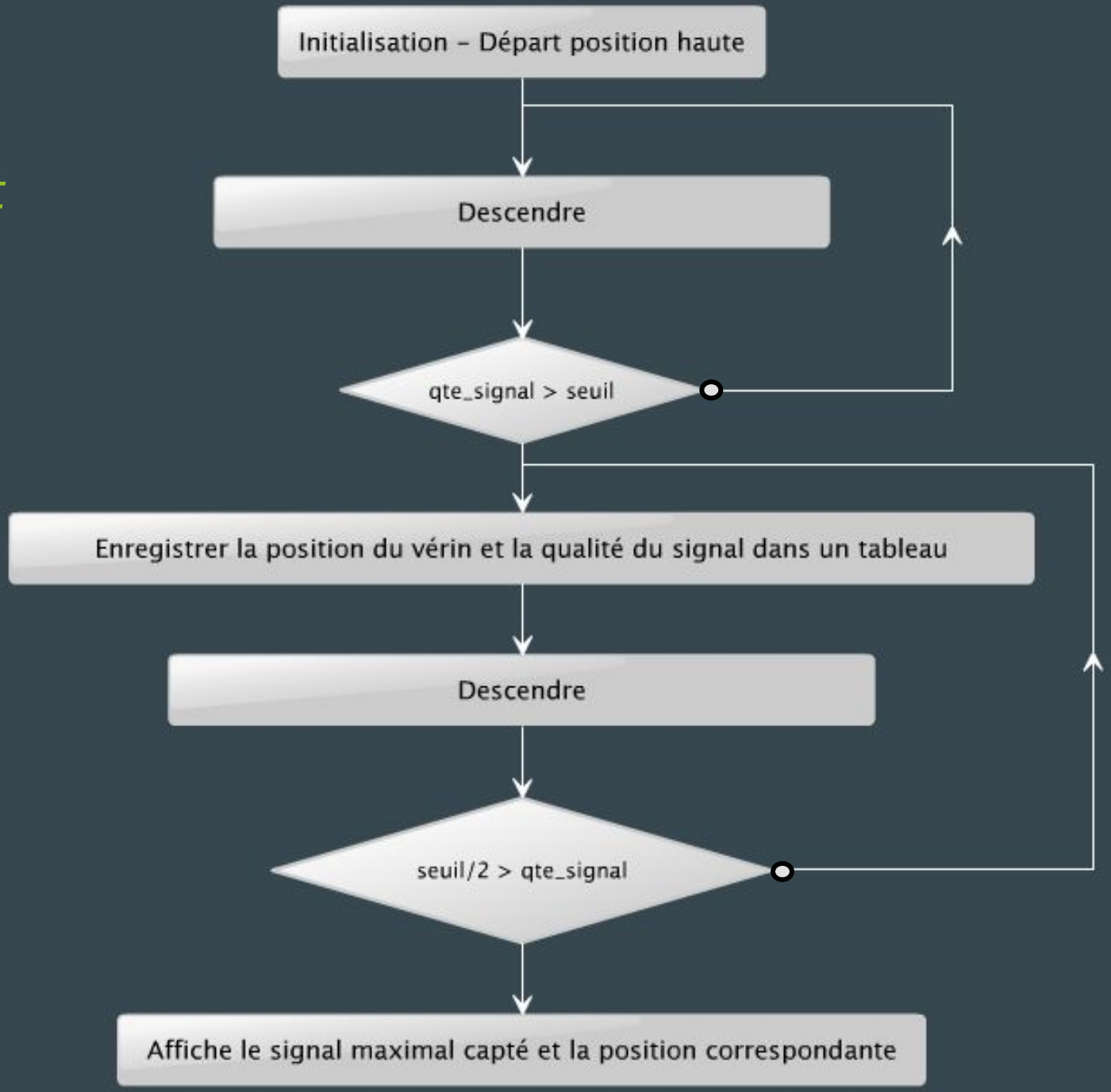
Travail réalisé

Algorithme de commande de la parabole en site



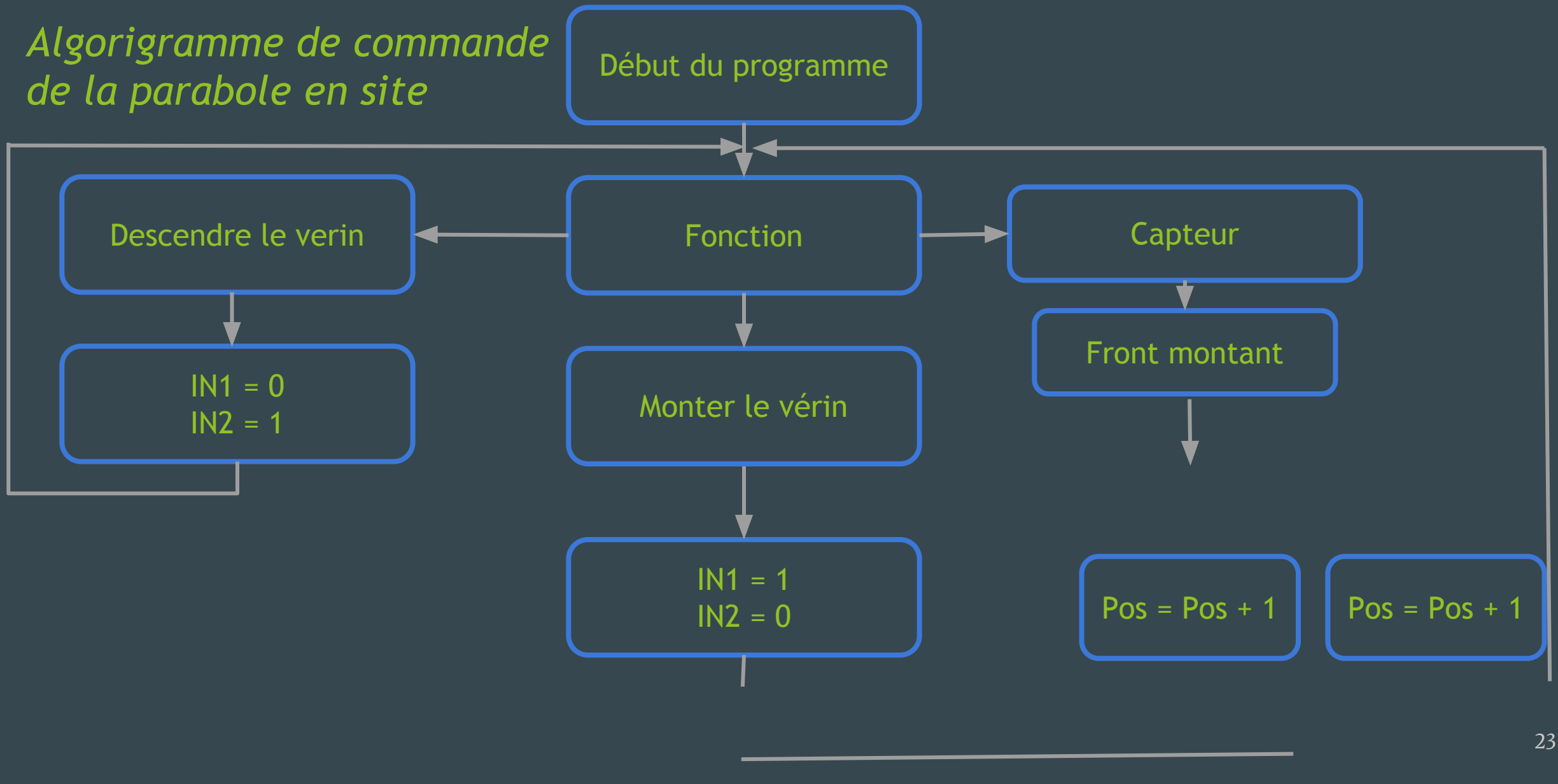
Travail réalisé

Algorigramme de positionnement de la parabole en site



Travail réalisé

Algorigramme de commande de la parabole en site



Techniques de présentation orale						
Qualité du diaporama	**0**	**1**	**2**			
Respect du timing de présentation	**0**	**1**				
Avancement du projet						
Suivi de projet – fiches du classeur	**0**	**1**	**2**	**3**		
Redéfinition efficace du planning	**0**	**1**				
Présentation détaillée des solutions techniques, justification des choix opérés	**0**	**1**	**2**	**3**	**4**	**5**
Résultats des premiers essais	**0**	**1**	**2**	**3**	**4**	
Remise en cause, réflexion						
Prise en compte des remarques de l'oral A	**0**	**1**	**2**			
Pertinence des échanges et réponses	**0**	**1**	**2**			
Note finale	**/20**					

Conclusion

- Développement de l’interface graphique
- Amélioration du code de positionnement du vérin
- Mise en commun avec la partie commande
- Réaliser le code de pilotage du rotor

Questions

GPS

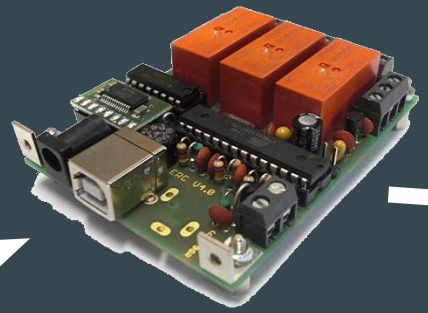
Position

Qualité du signal

Modem

Parabole

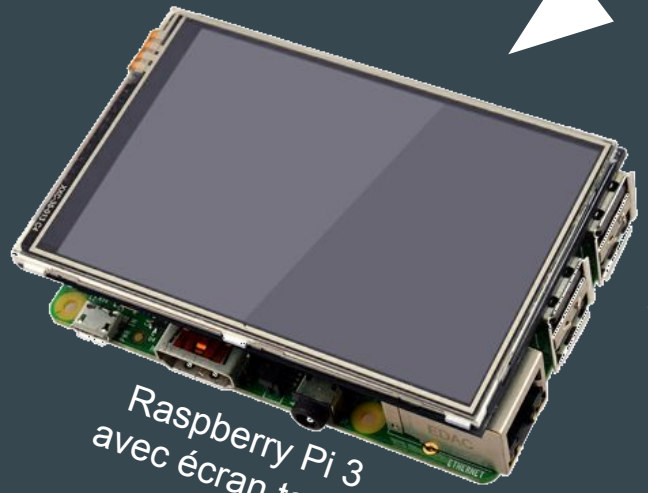
Orientation azimut



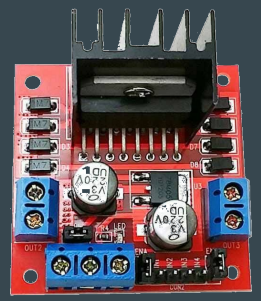
Carte interface
ERC V4 USB

Rotor & Contrôleur
Yaesu G1000DX

Orientation site



Raspberry Pi 3
avec écran tactile



Carte alimentation vérin

Vérin