logo.png

Valentin Pouce – Jan Barillec – Antonin Lyaët – Alrick Giry

coach : M. Vincent grimaud

2019-2020

Automatisation du placement d’une parabole et d’une antenne râteau pour la station spatiale internationale ISS

D4R4 – Groupe 303

logo.png

Valentin Pouce – Jan Barillec – Antonin Lyaët – Alrick Giry

coach : M. Vincent grimaud

2019-2020

Automatisation du placement d’une parabole et d’une antenne râteau pour la station spatiale internationale ISS

D4R4 – Groupe 303

Table des matières

[1 Introduction 6](#_Toc33009647)

[2 Diagramme de Gantt 7](#_Toc33009648)

[3 Carte mentale 9](#_Toc33009649)

[4 Budget 10](#_Toc33009650)

[5 Projet 11](#_Toc33009651)

[5.1 Matériel 11](#_Toc33009652)

[5.2 Programme 14](#_Toc33009653)

[6 Notre travail 17](#_Toc33009654)

[6.1 Correction et Amélioration 17](#_Toc33009655)

[6.2 Asservissement 21](#_Toc33009656)

[6.3 Documentation 24](#_Toc33009657)

[6.4 Communication avec l’ISS 24](#_Toc33009658)

[7 Bilan 26](#_Toc33009659)

[8 Conclusion 27](#_Toc33009660)

[9 Mots clés 29](#_Toc33009661)

[10 Tables des illustrations 30](#_Toc33009662)

[11 Annexes 31](#_Toc33009663)

[11.1 Protocole Ethernet (TCP/IP) : Câblage, trame, adressage et communication | Valentin Pouce 31](#_Toc33009664)

[11.2 Les différents écrans tactiles | Alrick Giry 35](#_Toc33009665)

[11.3 Annexe 2 | Python | Jan BARILLEC 37](#_Toc33009666)

[11.4 Les technologies de géolocalisation | Antonin Lyaët 40](#_Toc33009667)

[11.5 Annexe 4 43](#_Toc33009668)

[11.6 Annexe 5 44](#_Toc33009669)

[11.7 Annexe 6 45](#_Toc33009670)

# Introduction

Dans le cadre de notre projet de deuxième année, nous reprenons un projet déjà travaillé en collaboration avec Christophe Tailliez de Strategic Telecom Sécurité Civile.

L’objectif premier de ce projet est la création d’un système permettant le pilotage d’une antenne parabole, placée sur un camion, de façon à diriger l’antenne vers le satellite géostationnaire EUTELSAT 9A. Le système doit pour interagir par le biais d’un écran tactile codée en Python à l’aide d’un Raspberry Pi avec la parabole. Ceci doit présenter deux modes de contrôle, le mode Manuel et mode Automatique.

Le but de cette année est de conclure cette partie en élaborant un algorithme permettant l’asservissement de l’orientation de la parabole, autrement dit la création du mode automatique permettant l’orientation autonome de la parabole ainsi qu’une documentation technique permettant une compréhension claire et précise du fonctionnement du projet.

Ce projet est doté d’une seconde partie, qui consiste à la conception d’un système permettant la communication par satellite avec l’ISS (International Space Station) qui servira à des écoliers ou radioamateurs pour entrer en contact avec la station. Cette partie devait être étudiée d’avantage suite l’aboutissement de la première partie.

Cependant, à la suite d’une qualité de médiocre présenté dans les années précédentes et un délai qui ne permettant pas la bonne étude de cette deuxième partie, nous avons décidé de refaire la carte électronique et le boiter afin d’offrir au client un livrable de bonne qualité. Cette partie comprend l’ajout de connecteur dans le boiter afin de faciliter la connexion avec le système, re conception de la carte électronique afin qu’elle soit plus adaptée au produit final.

Pour la réalisation de ce projet, notre groupe, le groupe 303 est composé de :

* Valentin Pouce
* Alrick Giry
* Antonin Lyaët
* Jan Barillec

Par la supervision de M. GRIMAUD Vincent.

Dans ce rapport, nous allons tout d’abord expliquer et définir l’existant du projet quand nous l’avions commencé, afin de voir ce qui a déjà été pensé et définir les points à améliorer ou bien à faire. Ensuite, dans une seconde partie, nous allons présenter ce que nous avons fait mais aussi ce que nous allons effectuer afin de clore ce projet dans le temps imposé.

# Diagramme de Gantt

Le planning de ce projet était composé en deux parties qui représentaient les parties du projet, c’est-à-direle pilotage de la parabole jusqu’à mi-décembre puis dans un deuxième temps, l’antenne râteau avec l’ISS.

Lors du déroulement du projet, nous avons pris du retard sur ce planning prévisionnel suite à la confrontation avec quelques problèmes techniques qui seront détaillés dans le développement. En effet, nous avons dû renoncer au commencement de la deuxième partie de ce projet, la communication avec antennes râteaux. En revanche, nous avons opter pour rendre un produit de meilleure qualité à Monsieur Tailliez, ce qui implique de nouvelle tâche. Ainsi, début Mars nous avons divisé le groupe en deux afin de finir l’asservissement de la parabole et pouvoir avancer de manière efficace sur la qualité du livrable.

Pour ce projet, nous nous sommes réparti les tâches de la façon suivante afin d’optimiser au mieux le temps et les capacités de chacun :

* Antonin : Conception de l’algorithme permettant l’asservissement de la parabole.
* Alrick : Modification de l’IHM afin de la rendre plus intuitive pour l’utilisateur. Calibrage de l’écran tactile.
* Jan et Valentin : Etude du programme existant et vérification du bon fonctionnement de ceci. Amélioration et correction des disfonctionnements de la partie existante.
* Groupe : Rédaction de la documentation technique afin de rendre aisé la compréhension, la reproduction ou bien la reprise du projet.

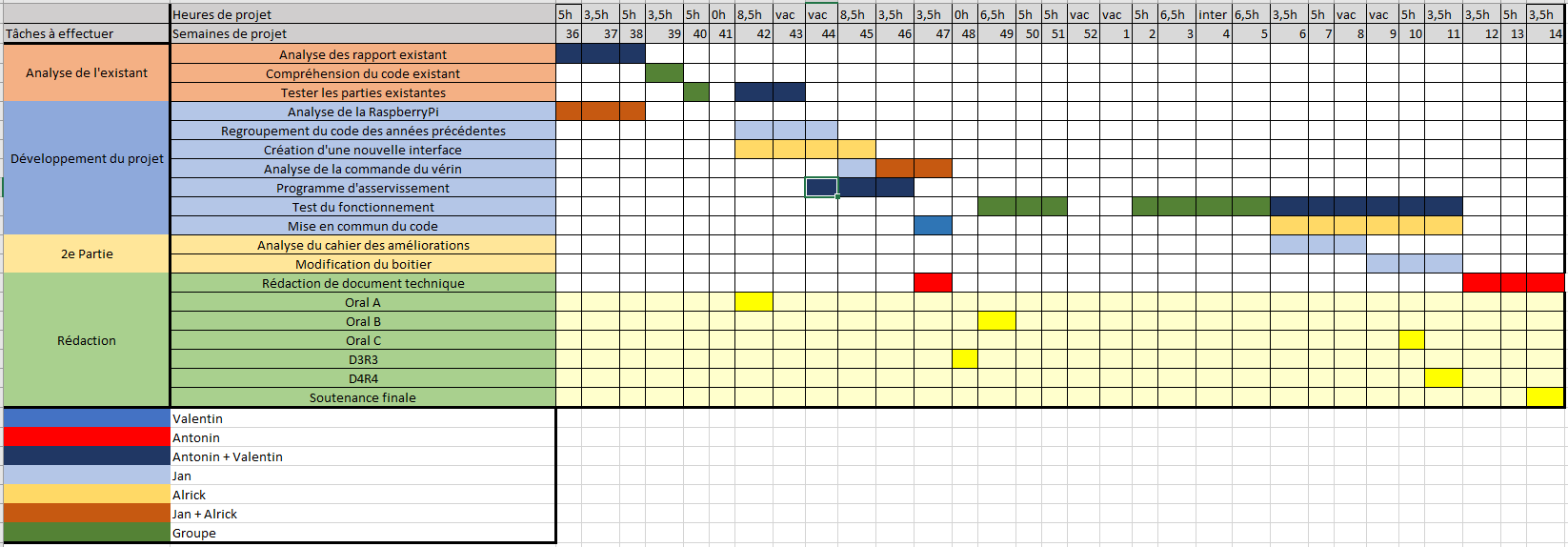


Figure Diagramme de Gantt

# Carte mentale

Le projet initial se déroule en cinq grandes parties pour arriver à un but, l’automatisation du placement de la parabole. (Voir Annexe)

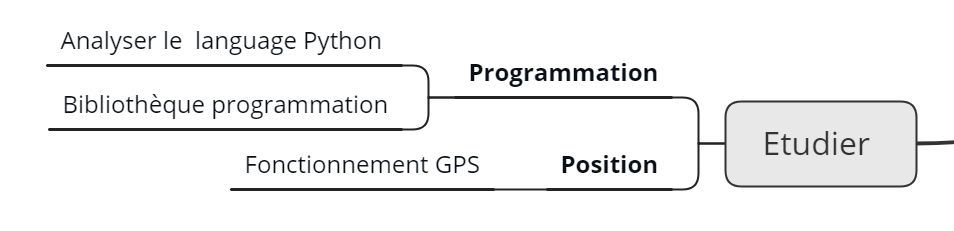


Figure Etudier Carte Mentale

La première partie étant l’étude de l’existant. Il était nécessaire d’analyser et de connaître ce qui avait déjà été effectué par les deux groupes précédents. Ainsi, l’étude s’est portée sur le langage Python qui nous étaient peu ou pas familier puisqu’il n’est pas enseigné dans le DUT, mais aussi sur les bibliothèques utilisées et les fonctions qui avaient été réalisées.

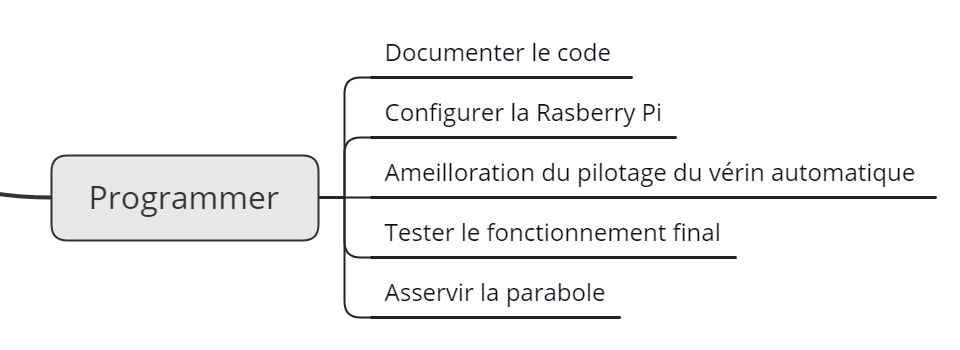


Figure Programmer Carte Mentale

Cette partie consiste à la réalisation des fonctions manquantes, corrigés ce qui n’étaient pas fonctionnels et demandées dans le cahier des charges. Puis l’ajout d’une documentation claires et précises afin de faciliter la compréhension de programme et des configurations nécessaires.

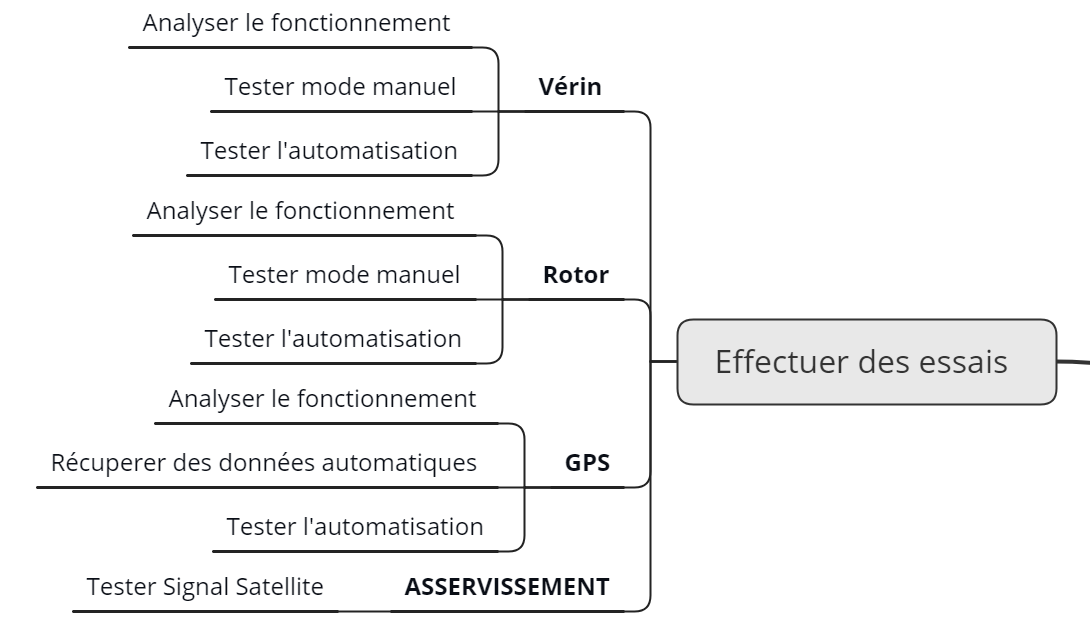


Figure Essais Carte Mentale

Les essais son très importants dans ce projet car ils nous ont permis dans un premier lieu, de voir ce qui étaient à corriger des parties des années précédentes ou non afin de partir sur un projet bien fonctionnel, de savoir ce qu’il était bon de rajouter d’un point de vue utilisateur, et dans un second lieu de valider nos réalisations.

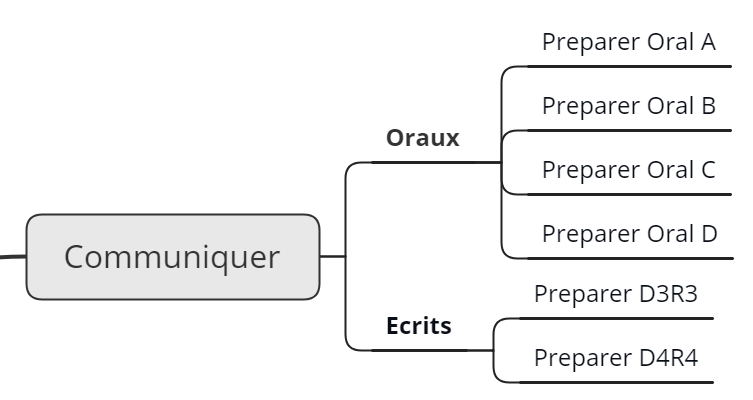


Figure Communiquer Carte Mentale

La communication consiste à la rédaction et préparations de rapport ou d’oral de présentation afin de faire un point sur l’avancement du projet. Cette rédaction permet de rendre un compte rendu technique sur le fonctionnement du système et du déroulement du projet.

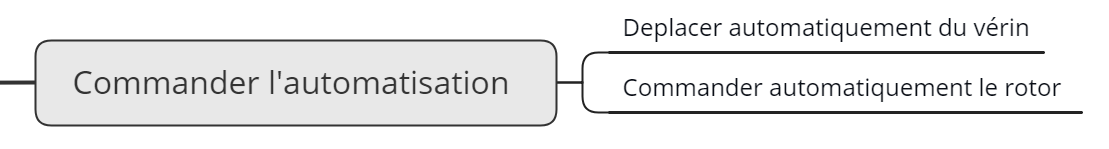


Figure Automatisation Carte Mentale

Ce projet a pour but de permettre l’automatisation de la position de la parabole par l’intermédiaire du pilotage d’un rotor et d’un vérin.

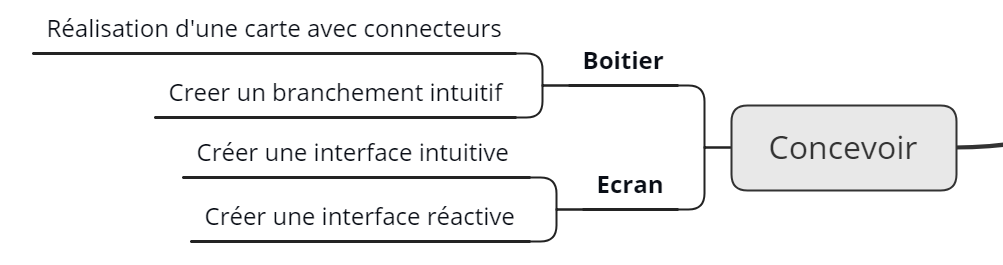


Figure Concevoir Carte Mentale

Pour une question de temps, nous avons opter pour rendre un livrable de meilleure qualité avec des finitions de qualités. Ce qui implique de nouvelles tâches et ainsi refaire le boitier et la carte électrique.

# Budget

La totalité du matériel nous a été fournie par Strategic Telecom, les précédents groupes qui ont travaillé sur le projet (2 au total) ont acheté le nécessaire. Cependant du matériel qui a été acheté et nous est inutile comme la Boussole CMP12 puisque son utilisation n’est plus envisagée.

Ce changement de cahier des charges concernant la deuxième partie de ce projet implique de nouveaux achats. Pour ce faire nous avons eu besoin de refaire la carte électrique, rajouter des connecteurs et d’autres. Le coût de cette opération s’estime à environ 28€.



Matériel fournit de l’année dernière

Matériel acheté cette année

Figure Budget du projet

Budget final du projet

# Projet

Avant de nous plonger dans notre avancement sur le projet, faisons un point sur le projet et l’existant, c’est-à-dire étudier le choix des composants, du matériel et du programme provenant des années précédentes.

Une image contenant intérieur, carte, texte

Description générée automatiquement

Figure Schéma de la structure globale du système

## Matériel

Voici le matériel dont nous disposons pour effectuer ce projet. Ce matériel pour la plupart, a été déterminé par les groupes des années précédentes selon les critères établis par le cahier des charges.

### Le vérin

Le choix du vérin a été fait sur deux critères par le groupe de l’année 2016-2017 :

* Avec une Parabole d’environ 15 kilogrammes, le vérin se doit d’avoir une poussée au minimum équivalente afin de pouvoir l’orienter convenablement.
* Le vérin est installé sur le camion, alors la seule alimentation possible est le 24 V des batteries présentes dans le camion.

Le vérin qui a été retenu est le Super Jack III de chez Jaeger avec les caractéristiques techniques suivantes :

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

D’après les informations qui nous ont été fournis par les rapports des années précédentes, le vérin prévu avec une alimentation de 36V DC fonctionne correctement avec une tension inférieure qui est celle du camion.

Cependant, ce vérin nécessite une carte d’interface afin d’interagir avec pour un contrôle plus aisé.

#### Carte d’interface

Une image contenant équipement électronique, circuit

Description générée automatiquementAfin de piloter le moteur du vérin, nous avons besoin d’une carte d’interface capable de délivrer la tension d’alimentation nécessaire pour le bon fonctionnement des moteurs. La Raspberry Pi n’étant pas capable de fournir une telle tension, il en est alors plus que nécessaire. Pour accomplir cette tâche, il s’agit du module L298N qui a été sélectionnée par les groupes des années précédentes pour sa compatibilité avec la Raspberry et les 24V de l’alimentation.

Une image contenant capture d’écran

Description générée automatiquement

**GND**

**24V**

Figure Carte d'interface L298N

Figure Schéma cablâge du vérin

Cette carte permet également le pilotage du moteur dans les deux sens de rotation par l’intermédiaire des entrées IN1 et IN2.

### Le rotor

Pour la rotation de la parabole en azimut, nous avons besoin d’un rotor. Ce rotor a déjà été choisi par le premier groupe. Il s’agit du YASEU G-1000DS.

Ce rotor a pour caractéristiques :

* 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

La Raspberry Pi communique avec le rotor par le biais d’un ERC Mini qui permet une liaison entre la Raspberry et le contrôleur du rotor.

### Le modem Tooway



Figure Parabole et module Tooway

Un module Tooway est installé dans le camion et permet de recevoir des données internet par satellite à très haut débit.

Le signal est capté par une parabole qui est montée sur le toit du camion. Il est aussi possible d’envoyer des données par le même module.

Ce module est développé par les sociétés Eutelsat et ViaSat.



Figure Zones de récupération du signal

Il est possible de récupérer des données sur l’entièreté de l’Europe grâce à un abonnement.

Les données sont ensuite transférées par une liaison Ethernet à la Raspberry Pi.

Afin de pouvoir positionner la parabole, le programme récupère la position du camion et grâce à celles-ci et à un site internet, il est possible de connaitre l’élévation et l’azimuth à lui donner.

### GPS

Le module GPS GSTAR IV a été fourni par la sécurité civile, il permet de géolocaliser le camion.

Le module est connecté par USB à la Raspberry Pi 3 et permet la réception de trames qui contiennent la latitude, longitude et l’altitude, ainsi que l’heure. Les trames suivent le protocole standardisé NMEA 0183 (National Electronics Association). Les différentes trames reçues fournissent chacune des informations différentes.

Dans notre cas, nous utilisons la trame GGA qui fournit l’heure, la latitude et longitude, l’altitude et le nombre de satellites trouvés.



Figure GPS GSTAR IV

Les groupes des années précédentes ont créé une classe GPS qui permet de lire directement la valeur de la latitude et longitude de la trame GGA ainsi que de déterminer l’azimut et l’élévation que la parabole doit avoir pour recevoir correctement le signal du satellite.

### Raspberry Pi

Pour relier et faire cohabiter tous les éléments du projet il faut un dispositif qui permette d’accueillir des périphériques USB, un écran tactile, des ports GPIO (General Purpose Input Output).

Figure Raspberry Pi 3

Les groupes qui ont travaillés sur le projet précédemment ont fait le choix d’utiliser un Raspberry Pi 3, car il s’agit d’un système linux embarqué peu onéreux et compact, mais aussi puissant.

L’écran tactile permettra de réaliser l’IHM, les ports GPIO permettront de contrôler le vérin et le port USB servira à recevoir les trames.

## Programme

Pour utiliser ce matériel, il a fallu mettre en place différents algorithmes afin d’utiliser finement ce dont on avait à disposition. La programmation est la partie clé de ce projet, c’est elle qui détermine le fonctionnement de ce système. Pour cela, les groupes des deux années précédentes élaborer quelques programmes qui permettent le fonctionnement du GPS, du rotor ou bien du vérin.

### Classe Rotor



Figure Classe Rotor

Cette classe permet de piloter le rotor. Chaque fonction a un rôle bien défini qui envoie une trame au rotor en fonction de ce que l’on veut qu’il fasse.

Liste des fonctions :

* « \_\_init\_\_(self, chemin) : créer un objet Rotor, c’est le constructeur. Si le rotor n’est pas connecté à la Raspberry, elle envoie un message d’erreur ;
* « tourner(self, angle) : permet de faire tourner le rotor jusqu’à un certains angle ;
* « angle(self) » : permet de lire l’angle sur lequel le rotor est actuellement ;
* « tournerHoraire(self) » : fait tourner le rotor dans le sens horaire jusqu’à qu’on lui dise de se stopper ;
* « tournerAntiHoraire(self) » : fait tourner le rotor dans le sens anti-horaire jusqu’à qu’on lui dise de se stopper ;
* « stop(self) » : stop la rotation du rotor ;
* « config(self) » : calibre et configure le rotor. Cette fonction est appelée dans le constructeur.

### Classe Vérin

Figure Classe Vérin

Cette classe permet de piloter le vérin. Chaque fonction à un rôle bien définie qui envoie une trame au vérin en fonction de ce que l’on veut qu’il fasse.

Liste des fonctions :

* « \_\_int\_\_(self, pinPush, pinPull, pinInterrupt) » : Constructeur du vérin qui définit les GPIOs pour monter ou descendre ainsi qu’un GPIO d’interruption ;
* « impulsion(self, channel) » : fonction de débogage. Écrit le nombre d’impulsion envoyé au vérin ;
* « monter(self) » : Changer les pins actifs et inactifs pour faire monter le vérin ;
* « descendre(self) » : Changer les GPIOs actifs et inactifs pour faire descendre le vérin ;
* « arreter(self) » : arrête le mouvement du vérin. Mets donc les deux pins en inactif

### Fonction GPS

La fonction GPS permet de faciliter l’utilisation du module GPS. Deux processus sont à l’œuvre dans la fonction GPS, la lecture de la trame GGA et la conversion de la latitude et de la longitude en azimut et élévation.

Trames GPS

Fonction GPS

Conversion

Latitude

Longitude

Longitude

Azimut

Lecture

GGA

Latitude

Elévation

La lecture de trame GGA se fait à l’aide de la bibliothèque **pynmea2,** qui permet d’extraire les variables Latitude et Longitude de la trame. La conversion est réalisée à l’aide de relations trigonométrique qui ont été déduite par le groupe de l’année dernière.

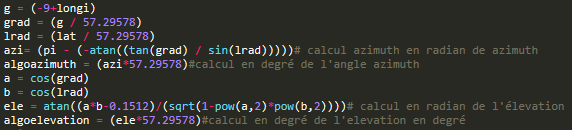
 La relation ci-dessus a été trouvé par le groupe de l’année dernière lors de l’analyse du Javascript fourni par le constructeur.

Figure Relations trigonomiques

# Notre travail

Dans cette partie, nous allons voir le travail qui est à effectuer et ce que nous avons pu faire à partir de l’existant. Cela va concerner l’ajout de certaine fonctionnalité comme demandé par le cahier des charges ou bien même la correction de fonctions défaillantes, qui ne seraient pas opérationnelles.

## Correction et Amélioration

Lors de la reprise du projet nous avons dû faire face à de nombreux disfonctionnements, qui nous ont permis de prendre conscience que le code avait besoins d’améliorations mais surtout d’une documentation complète pour permettre à une tierce personne de reprendre le projet.

### Modifications et corrections

Afin de rendre le projet utilisable nous avons donc dû, finir et corriger l’ensemble des erreurs du code pour pouvoir l’exploiter convenablement et y apporter des modifications, de plus nous l’avons amélioré pour rendre l’expérience utilisateur plus agréable.

#### Ajout du nord magnétique

Il a fallu intégrer le nord magnétique dans le programme afin de remplacer la boussole qui ne pouvait fonctionner dû à son environnement parasite et orienter convenablement le rotor.

#### Gestion des ports de communication USB

La gestion des ports de communication a été la première chose à comprendre car, deux de nos périphériques sont connectés par USB (le rotor et le GPS) et lorsqu’une erreur survenait, cela entrainait la coupure du logiciel. Pour obtenir une erreur il fallait brancher les équipements dans l’ordre que le code l’autorisait, en effet avec la Raspberry Pi le numéro de port de communication est attribué en fonction du nombre de périphérique branché précédemment.

On peut apercevoir sur l’image ci-dessus que le chemin d’accès au port (‘/dev/ttyUSB1’) est lié à l’objet gps de façon définitive, cela oblige donc l’utilisateur de notre système de brancher le gps en deuxième (car le premier port USB0 est occupé par le rotor).

Figure Chemin d'accès pour le GPS

Afin de rendre l’ordre de branchement totalement indépendant, nous avons imaginé et mis en place la solution suivante :

Figure Schéma fonctionnel de la correction de l'adresse des périphériques

Connection rotor établie

* PortUSBRotor = (‘/dev/ttyUSB1’)
* PortUSBGPS = (‘/dev/ttyUSB0’)

Initialisation :

* PortUSBRotor = (‘/dev/ttyUSB0’)
* PortUSBGPS = (‘/dev/ttyUSB1’)

Nous utilisons deux variables globales qui ont chacune des chemins d’adresse différent, et on réalise un test sur la connexion du rotor et si, la connexion est effectuée le reste du programme se déroule sinon, on inverse les ports.

#### Trames GPS

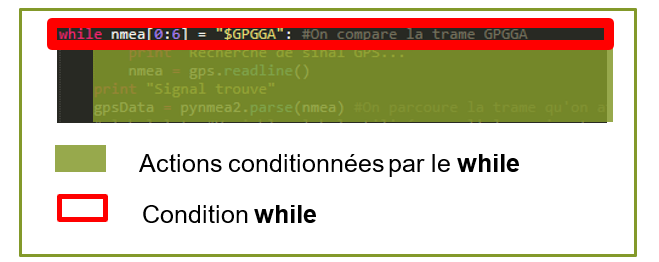
Lors des essais nous avons remarqué que le GPS ne reçois pas forcément les trames dès la première tentative, or tel que le programme a été conçus des arrêts imprévus du logiciel interviennent. Nous avons donc eu l’idée de ne démarrer le reste du programme qu’une fois que les trames ont bien été reçues.

Figure Correction de la classe GPS

Comme l’illustre l’extrait de code de la classe GPS ci-dessus, l’ensemble des actions sont conditionnées par une boucle while. La boucle while s’exécute tant que la condition est vraie.

Ici la condition devient vraie lorsque qu’une trame contient les caractères clés lié à la trame GGA.

#### Fonction tourner dans la classe Rotor

La fonction tourner permet de commander l’angle du rotor. Le rotor est commandé par l’envoi de mots sur l’ERC Mini. Le programme de l’année dernière est fonctionnel cependant lorsque l’angle envoyé était inférieur à 10 degrés, le programme cessait de fonctionner. Nous avons fixé le problème et nous avons pris en compte tous les angles possibles entre 0° et 360°

#### Vérin

Il est nécessaire de corriger le fonctionnement du vérin à partir d’une valeur entrée puisque la fonction qui régit sa sortie établie par les années précédentes n’est pas fonctionnelle.



Signal d’interruption envoyé à la broche 12

Figure Schéma de fonctionnement des Interruptions

Il y a une visse sans fin qui permet au vérin de rentrer ou de sortir. A chaque fois que la visse sans fin effectue un tour complet, une interruption est détectée par la RasberryPi.

Dans un but de simplification nous ne sommes pas préoccupés par la longueur que le vérin parcourt mais par l’angle que celui-ci donne à la parabole, puisque c’est le paramètre qui nous est utile dans le programme.

Pour comprendre la relation entre le nombre d’impulsions et l’angle de la parabole nous relevons pour plusieurs valeurs d’angles, le nombre d’impulsion et nous obtenons les résultats suivants :

Figure Linéarisation du vérin en Montée

On remarque que les points reportés sur un graphique forment une droite pour la montée mais aussi la descente. Afin de trouver la droite passe le plus précisément par l’ensemble, nous utilisons la fonction ‘courbe de tendance linéaire’ du tableur.

Figure Linéarisation du vérin en Descente

Grace a la courbe de tendance on remarque que le coefficient directeur est très proche en monté (0.0568) et en descente (0.0587), pour simplifier nous prenons dans la suite des calculs le même coefficient en montée et en descente (0.057). Nous pouvons également remarquer la présence d’une constante dans la courbe de tendance, celle présente dans la montée du vérin est dû au fait que le vérin commence à un angle approximativement 13°,et pour la descente le vérin démarre avec un angle de 45°.

Pour le placement approximatif on pourra donc utiliser la relation suivante car nous n’utilisons que la fonction monter du vérin :

On est donc capable de connaître l’angle de la parabole en fonction du nombre d’impulsions.

### L’IHM

#### Les objectifs

L’IHM ou l’Interface Homme Machine est un outil qui permet à l’utilisateur d’interagir avec le système qu’il souhaite utiliser. Dans notre cas, nous voulons pouvoir contrôler l’élévation et l’azimut de la parabole. Il faut aussi pouvoir activer le mode placement automatique et rentrer l’orientation du camion par rapport au nord magnétique. L’écran devra afficher les coordonnées GPS, les valeurs d’inclinaison de la parabole et la puissance du signal.



Nous repartirons avec la base de la première IHM car celle-ci est plus ergonomique. Toute l’interface a été codé en python avec la bibliothèque Pygame pour un écran tactile LCD de dimension 320x480.

Figure Écran utilisé (320x480)



Figure Logo de Pygame

Figure Logo de Python

En entrée utilisateur nous avons 4 boutons de direction et un bouton « auto » et un pavé numérique pour rentrer la valeur du nord magnétique.

L’IHM devra appeler les fonctions :

* rotor.tournerAntihoraire() / rotor.tournerHoraire()
* verin.monter() / verin.descendre()
* rotor.stop()
* verin.arreter
* positionnementauto()
* calibration()

#### Les modifications

Nous pourrons envisager de faire des menus pour visualiser les valeurs et en rentrer d’autres afin d’avoir une interface plus claire et lisible.



Figure IHM 2017

Ceci est la première interface réalisée pour ce projet (2017)



Figure IHM 2018

Ceci est la deuxième version de l’interface (2018)

Certaines fonctions avaient du mal à fonctionner nous les avons donc corrigés. Comme par exemple l’arrêt des fonctions d’élévations et de rotation au relâchement des boutons.

L’interface n’étant pas très pratique, ni ergonomique, nous avons décidé de remanier les éléments de changer les couleurs et rajouter un changement de couleur quand l’utilisateur appui et relâche un bouton (d’où la création d’une fonction actionVide() qui permet de ne rien faire quand on relâche un bouton).

Voici donc la dernière interface à ce jour :



Figure Esquisse IHM 2019

Nous avons rajouté un bouton calibration qui fait disparaitre tous les boutons sauf ceux pour faire tourner le rotor afin de le calibrer.

Un message apparait indiquant qu’il faut le placer à 0° puis à 360° puis l’interface de base réapparait.

#### Paramétrage de l’écran tactile

Quand nous avons récupéré les cartes SD nous n’arrivions pas à faire fonctionner l’écran tactile nous avons dû reformater une carte pour faire fonctionner celui-ci.

* Télécharger « balenaEtcher » pour le formatage/flash ainsi que la version « Buster with desktop and recommended software” de Raspbian.
* Flasher l’image de raspbian sur la carte SD avec le logiciel téléchargé précédemment.
* Mettre la carte SD dans une Raspberry, brancher la Raspberry et connecteur un écran en HDMI dessus.
* Suivre le tutoriel « <https://learn.adafruit.com/adafruit-pitft-3-dot-5-touch-screen-for-raspberry-pi/easy-install-2> » à partir de « Installer script ».
* Dans la rubrique « Configuring what shows where » suivez la partie « PiTFT as HDMI Mirror (Best for Raspbian Full/PIXEL) ».
* L’écran est maintenant configuré

### La Boite

Lors de l’utilisation de la boite réalisé par les années précédentes, nous avons remarqué que celle-ci était peu pratique à l’utilisation car les fils se déconnectaient. Nous avons donc pris l’initiative de faciliter l’utilisation de notre système en réalisant une boite et une nouvelle carte avec des branchement intuitifs.

Figure Shéma de principe de la carte

Routeur

(Ethernet)

GPS

(USB)

Rotor

(USB)

5V

0V

Rasperry Pi

&

Écran LCD tactile

Connecteur

nappe

L298N

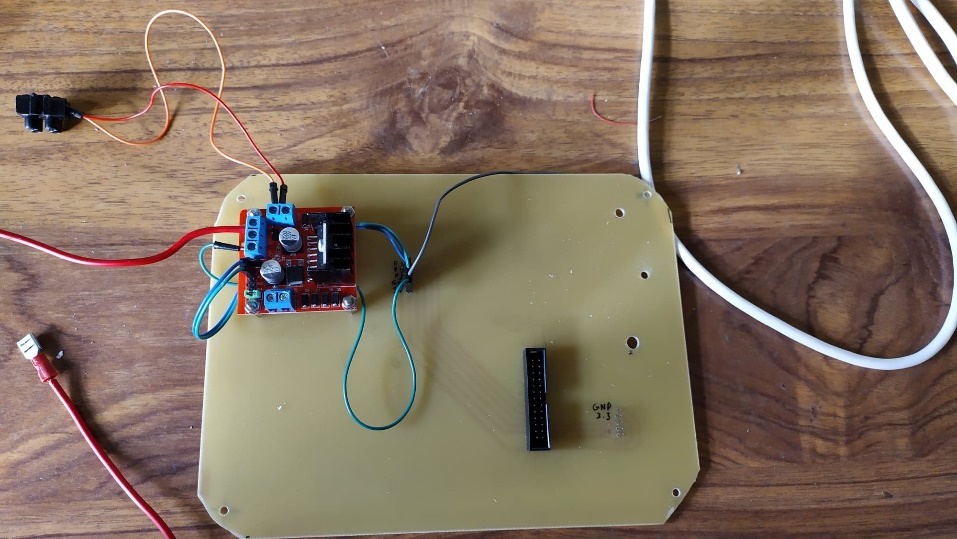
Contrôleur du vérin

ERC V4 Mini

Contrôleur du rotor

5v & GPIO x 2

#### L’ancienne carte



Branchement

vers le vérin

Branchement

vers la batterie

Branchement

vers la RaspberryPi

Figure Ancienne Carte

## Asservissement

Dans notre projet, nous devons mettre en place un mode automatique de la parabole pour qu’elle se calibre sans intervention humaine sur le satellite de communication voulue à un degré près. Pour effectuer ce mode automatique, on a dû mettre en place un algorithme qui permet à la parabole si elle doit monter ou descendre et aller à gauche ou à droite.

Pour effectuer un asservissement correct, on a dû mettre en place un algorithme permettant savoir par où le satellite est en fonction de la puissance du signal qu’on reçoit.

### Partie théorique

Pour simplifier le code et l’algorithme, nous avons choisis de faire fonctionner le rotor et le vérin chacun leurs tours dans une boucle.

Grâce à la fonction « getData », créée par le groupe d’il y a deux ans que l’on expliquera dans la partie pratique, on peut récupérer la puissance et le bruit que reçoit la parabole. On vient donc récupérer la puissance du signal à chaque fois que la boucle recommence et enregistrer l’ancienne puissance pour comparer.

En étudiant ce que devait faire l’asservissement, nous nous sommes retrouvés face à 4 cas lorsque le rotor fonctionne et 4 cas lorsque le vérin fonctionne.

Voici les différents cas :

#### Différents cas pour le rotor



Voici les 4 cas pour le rotor.

-Le rond rouge correspond à l’ancienne puissance ;

-Le rond bleu correspond à la nouvelle puissance ;

-La flèche correspond au sens de rotation du rotor ;

-Le rond avec un H à une plus grande puissance que le rond avec un B en dessous.

Si on prend cas par cas :

Dans le cas numéro 1, on voit que la puissance de la nouvelle mesure est plus puissante que celle de l’ancienne mesure. Le rotor part à l’opposé de l’ancienne mesure donc le rotor tourne dans le bon sens.

Dans le cas numéro 2, on voit que la puissance de la nouvelle mesure est moins puissante que celle de l’ancienne mesure. On en déduit donc que l’on a soit passé la puissance maximale soit on est parti dans le mauvais sens au début. On fait donc un changement de sens, le rotor va donc tourner vers l’ancienne mesure.

Dans les cas numéro 3 et 4, ils sont respectivement identiques au cas numéro 2 et 1 mais dans l’autre sens.

Avec ce fonctionnement, le rotor va faire un changement de sens lors du démarrage s’il est parti dans le mauvais sens puis en fera une infinité lorsqu’il s’approchera de la puissance maximale si on ne l’arrête.

#### Différents cas pour le vérin

Une image contenant capture d’écran, graphiques vectoriels

Description générée automatiquement

Voici les 4 cas pour le rotor.

-Le rond rouge correspond à l’ancienne puissance ;

-Le rond bleu correspond à la nouvelle puissance ;

-La flèche correspond au sens d’élévation du vérin ;

-Le rond avec un H à une plus grande puissance que le rond avec un B en dessous.

Le fonctionnement de l’asservissement pour le vérin reste le même que celui du rotor mais l’on gère l’élévation de la parabole.

Dans les cas numéros 1 et 3, il n’y a pas de changement de sens car la nouvelle mesure de puissance est plus élevée que la précédente.

Cependant pour les cas numéros 2 et 4, la nouvelle mesure de puissance est plus faible que la précédente donc le vérin change de sens d’élévation

### Partie pratique

Une fois les différents cas différenciés et qu’on a réfléchis au fonctionnement de l’algorithme que l’on devait mettre en place. On a pu commencer à coder l’algorithme.

#### La fonction « getData »

La fonction « getData » est une fonction créée par le groupe d’il y a deux ans. Elle permet de récupérer la puissance reçus par la parabole. Cette fonction utilise la page de configuration du modem Tooway où l’on peut retrouver la puissance et le bruit.

Figure Reception des données du modem

Les étudiants d’il y a deux ans avaient réussi à récupérer ces informations via une bibliothèque python « Request ». Cette bibliothèque vient faire une requête HTTP et reçoit une série de valeurs séparées par des « ## ». Par analyse de ces données, ils en ont déduit que la puissance était la 15 valeur (14 dans le tableau) et le bruit la 12 valeur (11 dans le tableau).

Figure Fonction getData()

Cette fonction vient nous renvoyer dans l’ordre : la puissance et le bruit.

#### Le code de l’asservissement

Pour l’asservissement, on choisit d’établir quelques règles pour que l’algorithme ne dure pas trop longtemps mais que la précision soit aussi présente. 3 règles pour l’asservissement ont été mise en place :

* Le pilotage du rotor et vérin se fait chacun leur tour ;
* On change d’élément pilotant la parabole (rotor->vérin ou inversement) lorsque l’objet piloté à fait 4 changements de sens depuis son cycle ;
* L’asservissement se finit après avoir changé 4 fois d’objets.

Avec ces 3 règles, on s’assure que le rotor et vérin soient bien calibrés pour que la parabole pointe vers le satellite.

Voici l’asservissement : (voir Annexe 5)

En résumé, voici un schéma bloc du fonctionnement de l’asservissement : (voir Annexe 6)

## Documentation

Suite à la reprise de ce projet, nous avons pu comprendre l’importance d’une bonne documentation pour mieux appréhender le projet et comprendre son fonctionnement, pour faciliter l’éventuelle reprise du projets par d’autres personnes nous avons choisi de réaliser une documentation à l’aide de l’outil Doxygen, qui permettra de créer une page HTML à partir des informations fournis afin de donner une vision graphique du code et rendre la compréhension plus aisées du code pour les prochains utilisateurs

De plus l’outil Doxygen permet d’associer le code a une bonne documentation, qui est essentiel dans notre projet puisque le langage Python n’est pas enseigné dans le cadre du DUT GEII.

## Communication avec l’ISS

Ce projet est composé de 2 parties. La première partie consistait à finir l’automatisation d’un système déjà réfléchi par des étudiants d’années précédentes. La seconde partie commencera seulement quand la première partie sera terminée.

Une image contenant satellite, transport, extérieur, clôture

Description générée automatiquementCette partie consiste au début du développement d’une nouvelle application pour quatre antennes râteaux. En effet, ce système devra communiquer par l’intermédiaire de ces antennes, avec la Station Spatiale International, autrement dit l’ISS. Ce projet servira par la suite à des écoles ou bien à des radios amateurs, pour entrer en contact avec la station.

Figure ISS

Pour des raisons techniques et que le cahier des charges le permet, nous pourrons utiliser des algorithmes déjà élaborer dans la première partie afin d’accomplir le projet. On peut prendre pour exemple, le pilotage du vérin qui pourra être similaire.

A l’heure actuelle, cette partie n’est pas encore démarré puisque la première partie du projet en est encore à l’état d’expérimentation.

# Bilan

Dans l’état actuel du projet, le système est fonctionnel et de nouvelles fonctionnalités on même pu être rajoutées telle que la Calibration possible à partir de l’interface tactile. De plus, une amélioration de la qualité du système est prévue pour fournir un livrable de meilleure qualité. Notre projet s’est entièrement articulé autour du système d’asservissement en positions de la parabole.

* Nous avons tout d’abord étudié l’existant des deux premières années afin de comprendre l’existant et son fonctionnement. Par la suite nous avons pu tester les différentes parties du programme pour savoir ce qui est fonctionnel ou non pour corriger les défauts et ainsi partir d’un programme fonctionnel. La difficulté de cette partie était l’analyse de ce qui avait été fait car parfois peu explicite ou peu de documentation pour le programme par exemple ce qui a entrainé de retard dans le planning prévisionnel. Notre erreur a été de se concentrer principalement sur l’existant de l’année précédente ce qui nous parfois fait partir dans de mauvaise direction.
* Une fois le projet compris et fonctionnel, nous sommes concentrés sur l’asservissement, le point principal du projet tout en rajoutant certaine fonctionnalité à ce système pour y faciliter son utilisation (ajout de la calibration, nouvelle Interface Homme Machine plus intuitive, …).
* Une fois les tests et les ajouts validé par l’intermédiaire de différents tests sur le site de Larcay, il ne nous restait peu de temps pour s’attaque à la deuxième partie de ce projet. Ainsi, nous avons décidé de refaire le boîtier et la carte électronique qui étaient peu adaptés et peu qualitative. Ce changement de cahier des charges a entrainé des frais supplémentaires et donc une révision du budget.

# Conclusion

Ce projet consistait à l’automatisation d’un placement de parabole afin de pointer un satellite plus rapidement et surtout sans intervention humaine. Un sujet complet et à la fois qui sort de l’ordinaire. En effet, il a pu pour chacun d’entre nous nous faire sortir de nos précieux acquis afin de nous confronter à de l’analyse de documentation, de rapport (D3R3, D4R4 d’années précédentes) mais également à de la programmation effectuée par d’autre personne. Ce projet assez complet dans les compétences requises, nous a permis de conforter ce qui a été vu en cours tant dans la technique que dans la communication.

Conclusion personnelle Jan :

Le projet est pour moi très intéressant car il me permet de découvrir un nouveau langage, le python et un nouveau système linux embarqué, la Raspberry Pi. De plus récupérer un projet déjà commencé est une première pour moi et, cela est très instructif par la difficulté de compréhension des travaux d’une autre personne, mais aussi très utile puisqu’il permet de rendre compte qu’un projet mal documenté, est très dur à comprendre par une personne extérieure au projet.

Conclusion personnelle Antonin :

Ce projet m’apporte beaucoup et est donc très intéressant pour moi. En réalisant ce projet, j’ai pu découvrir la programmation sur Raspberry et donc en même temps un nouveau langage : le python qui est pour moi une première pour ce type de langage car c’est le premier langage de niveau haut que j’utilise.

De plus dans le projet, je me suis occupé de la programmation de l’asservissement. J’ai donc dû mettre en place un algorithme ce qui est aussi une première pour moi. L’algorithmie est très importante dans le monde de l’informatique donc mettre en place un algorithme qui est assez simple mais complet pour commencer m’aider énormément.

Malgré la déception de retrouver un projet non-documenté et non-fonctionnelle, j’ai pu me rendre compte que la documentation que j’oublie très régulièrement ou que je ne fais pas par fainéantise est en fait très importante pour quelqu’un d’extérieur au projet.

Conclusion personnelle Alrick :

Travailler sur ce projet est très intéressant pour moi car il me permet d’approfondir mes connaissances en Python et d’utiliser pour la première fois une Rapsberry Pi munie d’un écran tactile. Développer l’IHM depuis une IHM déjà existante est très instructif car cela me permet de voir qu’une interface mal faite ne permet pas une bonne utilisation du système. De plus reprendre un code sans aucunes explications dessus est un défi à relever et cela me fait comprendre qu’il faut vraiment bien commenter son code pour que d’autres personnes puissent l’exploiter facilement. Le travail avec le groupe ne pose pas de problème ce qui permet au projet de bien de concrétiser.

Conclusion personnelle Valentin :

Ce projet m’a permis de sortir de ma zone de confort à pratiquer sur des cartes, des langages que je connaissais. Ainsi, j’ai pu expérimenter et en apprendre davantage sur le langage Python par exemple ou bien sur le fonctionnement d’une Raspberry Pi, ce qui m’étaient totalement inconnu jusqu’à maintenant. Le reprise d’un projet ne fût pas une tâche facile, puisqu’il a fallu comprendre le fonctionnement de ce qui a déjà été effectuer, savoir ce qui fonctionne, ce qui nous montre l’importance d’une bonne documentation technique.

Malgré tout, ce projet reste néanmoins très enrichissant tant dans son niveau technique que dans son sujet. Il nous a permis d’innover et de réfléchir à l’aspect pratique du système pour nous pousser à aller plus loin qu’établi dans le cahier des charges.

# Mots clés

automatisation, 23

**azimut**, 11, 12, 18

classe, 12, 14, 17, 18, 28, 35, 36

**élévation**, 12, 18

Ethernet, 12, 29, 30, 31, 32

fonction, 14, 15, 16, 18, 20, 22, 23, 30, 35, 36, 39

GGA, 12

GPS, 12, 13, 18

**IHM**, 13, 18, 20, 34

latitude, 12

parabole, 6, 7, 8, 11, 12, 18

pilotage, 6, 7, 11, 24

python, 18

Raspberry, 6, 11, 12, 13

vérin, 10, 11, 13, 24

# Tables des illustrations

[Figure 1 Diagramme de Gantt 7](file:///C:\Users\valen\Desktop\GEII\Projets\PROJET%20S3\ISS-PROJECT\D3R3%20cette%20année%20(1).docx#_Toc26464849)

[Figure 2 Carte Mentale du projet 8](file:///C:\Users\valen\Desktop\GEII\Projets\PROJET%20S3\ISS-PROJECT\D3R3%20cette%20année%20(1).docx#_Toc26464850)

[Figure 3 Budget du projet 9](file:///C:\Users\valen\Desktop\GEII\Projets\PROJET%20S3\ISS-PROJECT\D3R3%20cette%20année%20(1).docx#_Toc26464851)

[Figure 4 Schéma de la structure globale du système 10](#_Toc26464852)

[Figure 5 Carte d'interface L298N 11](file:///C:\Users\valen\Desktop\GEII\Projets\PROJET%20S3\ISS-PROJECT\D3R3%20cette%20année%20(1).docx#_Toc26464853)

[Figure 6 Schéma cablâge du vérin 11](#_Toc26464854)

[Figure 7 Parabole et module Tooway 12](file:///C:\Users\valen\Desktop\GEII\Projets\PROJET%20S3\ISS-PROJECT\D3R3%20cette%20année%20(1).docx#_Toc26464855)

[Figure 8 Zones de récupération du signal 12](file:///C:\Users\valen\Desktop\GEII\Projets\PROJET%20S3\ISS-PROJECT\D3R3%20cette%20année%20(1).docx#_Toc26464856)

[Figure 9 GPS GSTAR IV 13](file:///C:\Users\valen\Desktop\GEII\Projets\PROJET%20S3\ISS-PROJECT\D3R3%20cette%20année%20(1).docx#_Toc26464857)

[Figure 10 Raspberry Pi 3 13](file:///C:\Users\valen\Desktop\GEII\Projets\PROJET%20S3\ISS-PROJECT\D3R3%20cette%20année%20(1).docx#_Toc26464858)

[Figure 11 Classe Rotor 14](file:///C:\Users\valen\Desktop\GEII\Projets\PROJET%20S3\ISS-PROJECT\D3R3%20cette%20année%20(1).docx#_Toc26464859)

[Figure 12 Classe Vérin 15](file:///C:\Users\valen\Desktop\GEII\Projets\PROJET%20S3\ISS-PROJECT\D3R3%20cette%20année%20(1).docx#_Toc26464860)

[Figure 13 Chemin d'accès pour le GPS 17](file:///C:\Users\valen\Desktop\GEII\Projets\PROJET%20S3\ISS-PROJECT\D3R3%20cette%20année%20(1).docx#_Toc26464861)

[Figure 14 Schéma fonctionnel de la correction de l'adresse des périphériques 18](#_Toc26464862)

[Figure 15 Correction de la classe GPS 19](file:///C:\Users\valen\Desktop\GEII\Projets\PROJET%20S3\ISS-PROJECT\D3R3%20cette%20année%20(1).docx#_Toc26464863)

[Figure 16 Écran utilisé (320x480) 20](file:///C:\Users\valen\Desktop\GEII\Projets\PROJET%20S3\ISS-PROJECT\D3R3%20cette%20année%20(1).docx#_Toc26464864)

[Figure 17 Logo de Pygame 20](file:///C:\Users\valen\Desktop\GEII\Projets\PROJET%20S3\ISS-PROJECT\D3R3%20cette%20année%20(1).docx#_Toc26464865)

[Figure 18 Logo de Python 20](file:///C:\Users\valen\Desktop\GEII\Projets\PROJET%20S3\ISS-PROJECT\D3R3%20cette%20année%20(1).docx#_Toc26464866)

[Figure 19 IHM 2017 20](file:///C:\Users\valen\Desktop\GEII\Projets\PROJET%20S3\ISS-PROJECT\D3R3%20cette%20année%20(1).docx#_Toc26464867)

[Figure 20 IHM 2018 21](file:///C:\Users\valen\Desktop\GEII\Projets\PROJET%20S3\ISS-PROJECT\D3R3%20cette%20année%20(1).docx#_Toc26464868)

[Figure 21 Esquisse IHM 2019 21](file:///C:\Users\valen\Desktop\GEII\Projets\PROJET%20S3\ISS-PROJECT\D3R3%20cette%20année%20(1).docx#_Toc26464869)

[Figure 22 Fonction getData() 24](file:///C:\Users\valen\Desktop\GEII\Projets\PROJET%20S3\ISS-PROJECT\D3R3%20cette%20année%20(1).docx#_Toc26464870)

[Figure 23 ISS 26](file:///C:\Users\valen\Desktop\GEII\Projets\PROJET%20S3\ISS-PROJECT\D3R3%20cette%20année%20(1).docx#_Toc26464871)

[Figure 24 Table d'adresse MAC 32](#_Toc26464872)

[Figure 25 Aperçu document les différents types d'écrans tactiles 36](file:///C:\Users\valen\Desktop\GEII\Projets\PROJET%20S3\ISS-PROJECT\D3R3%20cette%20année%20(1).docx#_Toc26464873)

# Annexes

## Protocole Ethernet (TCP/IP) : Câblage, trame, adressage et communication | Valentin Pouce

### Introduction

Ce document va me permettre d’analyser et de comprendre le fonctionnement de la communication par Ethernet et son protocole. Il retrace l’évolution du l’Ethernet, son fonctionnement et des différentes normes en vigueur concernant cette communication et de ses avantages.

### Indentification

Ce document est un cours en ligne sur le protocole Ethernet et ses spécificités sur le site coursreseaux.com. Ce cours s’intitule Protocole Ethernet (TCP/IP) : Câblage, trame, adressage et communication. Ce cours permet de répondre aux questions Qu’est ce que le protocole Ethernet ? À quoi sert-il ? et aussi Comment fonctionne-t-il ?

Lien : https://coursreseaux.com/protocole-ethernet/

### Évaluation du contenu

|  |  |  |  |
| --- | --- | --- | --- |
| Critères | Faible | Moyen | Bien |
| Fiabilité |  |  |  |
| Illustration |  |  |  |
| Lisibilité |  |  |  |
| Pertinence |  |  |  |
| Accessibilité |  |  |  |

### Liste des mots clés

|  |  |
| --- | --- |
| Protocole | Ethernet |
| Communication | Adressage |
| Trame[[1]](#footnote-1) | TCP[[2]](#footnote-2) |
| IP[[3]](#footnote-3) | Norme |
| Mbps[[4]](#footnote-4) | Réseau |
| Câble droit | Câble croisé |
| Technologie |  |

### Compte rendu

#### Evolution vers la communication des trames (Ethernet Switching)

* Introduction du câblage 10BASE-T et de UTP 🡪 Popularisation des Hub en tant que moyen économique et facile d’utilisation
* Limitations des hubs avec les collisions de trames sur un même segment
* Ponts Ethernet pour résoudre les problèmes dans les réseaux locaux partagés 🡪 Amélioration des performances du réseau et réduction de trafic inutile
* Apparition des commutateurs 🡪 Augmentation des interfaces et augmentation du débit
  + Bande passante dédiées à chaque port
  + Environnement sans collision
  + Full Duplex

#### Logique de commutation des trames

* Commutateur Ethernet transfère les trames individuelles d’un port source vers un nœud ou un port de destination
* Connexion logique en fonction des adresses MAC sources et destinations via l’en-tête de la trame
* Le Commutateur décide :
  + Quand transférer ou filtrer une trame en fonction de l’adresse MAC de destination
  + Examine l’adresse de la source
  + Créer un environnement sans boucle avec les autres Commutateurs via le protocole Spinning Tree (STP)
* Pour le filtrage, il utilise une table d’adresse MAC inscrite dynamiquement dans la RAM et compare avec la destination

Une image contenant capture d’écran

Description générée automatiquement

Figure Table d'adresse MAC

#### Vue d’ensemble de la technologie Ethernet

* 802.3 est la norme IEEE pour l’Ethernet 🡪 Désigne une famille de normes définissant l’ensemble des couches physiques et la liaison de données de la technologie LAN en place
* L’Ethernet sépare la couche liaison du modèle OSI en deux couches distinctes :
  + Logical Link Control (LCL) établi dans la norme 802.2
  + Media Access Control (MAC) établi dans la norme 802.3
* LCL gère la connexion entre la couche MAC et celle réseau
* MAC encapsule les données :
  + Assemblage des trames avant transmission
  + Analyse de la trame lors de la réception
  + Contrôle l’accès aux médias 🡪 Par la méthode Carrier Sense Multiple Access with Collision Detection (CSMA/CD) lors du Half Duplex

#### Technologie Ethernet Traditionnelle et actuelles

* Spécifications 10BASE5 et 10BASE2 connectés sur un bus
* Arrivé UTP 🡪 Développement des HUB avec la topologie Etoile et nécessite le CSMA/CD
* Topologie étoile permet le Full Duplex ainsi désactive le CSMA/CD
* Ethernet 🡪 10Mbps sur cuivre
* Fast Ethernet 🡪 100Mbps sur cuivre
* GigaBit Ethernet 🡪 1000Mbps par fibre

#### Câblage UTP

* Utilisation courante de 2 ou 4 paires de fils 🡪 Généralement utilisé avec un connecteur RJ-45 (8 fils)
* Telecommunication Industry Association (TIA) 🡪 Norme des couleurs des fils
* Appareil utilisant paire de broche opposées pour transmettre 🡪 câble droit
* Appareil utilisant la même paire de broche pour transmettre 🡪 câble croisé

#### Adressage Ethernet

* IEEE définit un format d’attribution d’adresse LAN
* Adresse MAC unique :
  + 1ère moitié provient du fabricant de la carte 🡪 Identificateur unique de l’organisation (OUI)
  + Seconde moitié est donnée par le fabricant de la carte et n’est jamais utilisée par une autre carte avec le même OUI
* Ethernet possède des adresses de groupe :
  + Adresse de diffusion : Tous les appareils doivent traiter la trame suivante FFFF.FFFF.FFFF.FFFF
  + Adresse multidiffusion : Permet à un sous ensemble d’appareils sur un même réseau local de communiquer

#### Ethernet Framing

* « Framing » représente la définition des informations supposées se situer dans les données reçues

Une image contenant capture d’écran

Description générée automatiquement

* Préambule : Synchronisation des données
* SFD (Délimiteur du début de trame) : Marque le début du champ MAC de destination
* Longueur : Taille du champ de données de la trame
* FCS (Séquence de contrôle de trame) : Méthode pour la carte réseau de déterminer la présence d’erreur

### Avis critique du document

Je donnerai la note de 8/10 à ce cours sur le protocole Ethernet. En effet, ce cours a bien ciblé mes attentes en termes d’informations pour mieux comprendre cette technologie pour mon projet comme par exemple avec le détail de la trame Ethernet.

Ce cours a d’abord traité des moyens de communique par l’Ethernet avec le fonctionnement des Hub et des Commutateurs. Puis, le développement des fonctionnements des émissions de trames avec cette technologie. Enfin, un brief vu d’ensemble des différentes évolutions et technologie de l’Ethernet compléter avec le câblage adéquat pour la communication. Cours intéressant en termes d’informations et d’images qui facilitent la compréhension du sujet bien que par moment il y est difficile de bien cerner les notions évoquées.

## Les différents écrans tactiles | Alrick Giry

### Identification

* Auteurs : Pierre-Sylvain ROOS, Laurent ACHARD
* Titre : Le tactile, troisième géné utilisateurs
* Collection : Techno sans frontière
* Parution : Mai-Juin 2014

### Évaluation du contenu

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Critères d'évaluation** | **Faible** | **Moyen** | **Bien** | **Très bien** |
| Degré de fiabilité de l'information |  |  |  |  |
| Utilité et accessibilité, pertinence de ces informations |  |  |  |  |
| Qualité, utilité des illustrations |  |  |  |  |
| Lisibilité du document… |  |  |  |  |

### Liste des mots clefs pour ce document

* Capteur
* Ergonomie
* Informatique
* Logiciel
* Multitouch

### Compte rendu

* Interfaces homme machine : lignes de commandes (CLI) puis clavier souris (GUI) puis écran tactile
* Premier écran tactile né en 1953, introduit dans notre vie quotidienne en 2007 avec l’iPhone
* Dalle tactile n’a pas d’écran contrairement aux écrans tactiles
* Écrans tactiles communiquent par USB, données interprétées par driver et librairies et affiché sur écran grâce à la carte graphique.
* Deux types de tactile : électriques et optiques
  + Résistif (électrique) : résistant, utilisable avec n’importe quel pointeur, pas de pression simultanée
  + Capacitif (électrique) : utilisable sur surface parfaitement lisse et par pointeur tactiles seulement, apte au multitouch.
  + Quadrillage de faisceaux infra-rouge et capteurs (optique) : détectent plusieurs doigts (jusqu’à 40), résistant.
  + FTIR (frustrated total internal reflection), DI (diffused illumination), PSD (planar scatter detectection), optical imaging et DSI (diffused screen illumination) : projection infra-rouges captés par caméra.
* Prix : 1500€ un écran capacitif 25 pouces, 80 000€ mur tactile 7m²
* Écrans classiques 6 à 10 points
* Écrans plus avancés 32 à 64 points
  + Permet le travail collaboratif
* Les grands écrans tactiles permettent donc de gagner en rapidité et en organisation
* Recherche pour transformer n’importe quelle surface en surface tactile
* Recherche écrans simulant objets en 3D
* Recherche impression d’écrans tactiles

### Pertinence des informations

Je donnerai la note de 8/10, à ce document. Il correspond à ma partie dans notre projet car il traite des écrans tactiles et des IHM. L’information est fiable car les deux auteurs sont soit ingénieur ou chercheurs et elle provient d’un magazine spécialisé dans la technologie. Les sources sont citées en fin de document ce qui permet de vérifier l’information. Les images sont très claires et permettent de bien comprendre le fonctionnement de chaque technologie. Il est parfois difficile de suivre le cours de sa phrase quand la ligne change de page et qu’une illustration avec sa description est positionnée entre les deux pages. Je reproche aussi à l’article de rester trop vague sur certains points et en même temps d’aller trop loin dans certains détails. Le début est aussi un peu confus car l’auteur commence à expliquer comment fonctionnent les écrans tactiles puis fait un historique des découvertes et ensuite revient au fonctionnement.

### Aperçu du document

Figure Aperçu document les différents types d'écrans tactiles

## Annexe 2 | Python | Jan BARILLEC

### Identification

|  |  |
| --- | --- |
| Auteur : | Pierre Giraud |
| Titre : | Apprendre à programmer en Python |
| URL : | https://www.pierre-giraud.com/python-apprendre-programmer-cours/ |
| Date de 1ère mise en ligne : | 2019 |

Etude du chapitre « Python Orienté Objet »

### Liste des mots clés

* Classe
* Objets
* Attributs
* Variables
* Arguments
* Fonction
* Encapsulation
* Dérivation
* Héritage

### Compte rendu

* Qu’est-ce qu’un objet en programmation ?
  + Les types str,int,list sont des objets
* Un objet est une entité qui regroupe un ensemble de fonctionnalités
* Les attributs désignent les variables de la classe et les fonctions de celle-ci sont appelées méthodes
* Comment crée-t-on un objet ? Présentation des classes
  + Pour créer un objet il faut utiliser des entités que l’on appelle classes, c’est quelque peu le plan de création d’un objet.
  + Les objets permettent de regrouper un ensemble d’opérations pour permettre au code d’être plus clair et concis.
  + Les objets sont plus faciles à manipuler car le programmeur n’a pas besoins de savoir comment l’objet est codé mais seulement les actions qu’il permet de faire.
  + Les objets permettent à tout programmeur de réutiliser et de créer de nouveau objets à partir d’anciens.
  + Il est possible de créer une classe enfant à partir d’un autre parent, ainsi les enfants héritent des propriétés du parent, c’est le principe de dérivation.
* Création d’une première classe et objet en Python
  + En Python on déclare une nouvelle classe avec le mot clé « class »
  + Pour créer un objet on réalise la chose suivante : NomObjet = NomClasse()
  + Les attributs sont des variables contenues dans l’objet sont accessibles dans l’ensemble des méthodes de l’objet.
  + Il est possible en python de créer un attribut spécifique à un objet en particulier.
* Différence entre variable et attribut de données et entre fonction et méthode
  + Variables et fonctions sont totalement indépendantes de l’objet tandis que, les attributs et méthodes sont locales à l’objet.
  + Pour accéder à un attribut ou une méthode il faut donc toujours spécifier de quel objet celui-ci provient.
* Les classes et le principe d’encapsulation
  + Le principe d’encapsulation réside dans le fait que l’ensemble des variables de la classe n’est pas accessible du reste du programme.
  + Cela permet de limiter l’usage des variable globales et de rentre le code universel et potentiellement réutilisable dans d’autres situations.
* Initialiser des objets avec \_\_init\_\_()
  + \_\_init\_\_() est une fonction qui s’exécute à la création de la classe, elle permet d’initialiser des variables. Les arguments qui lui sont fournis doivent être précédé du mot self.
  + Self permet de désigner que la ou les variables sont liés à la classe dans laquelle la fonction est créé.
* Variable de classe et attributs de données d’un objet
  + Initialiser des variables est utile que si, à la création de l’objet les valeurs sont constantes.
  + Il faut éviter de créer des variables avec des données qui peuvent être modifié par le reste du programme (Dictionnaire par exemple).

### Évaluation du contenu

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Critère d’évaluation | Faible | Moyen | Bien | Très bien |
| Degré de fiabilité de l’information |  |  |  | X |
| Utilité, accessibilité, pertinence des informations |  |  |  | X |
| Utilité des illustrations |  |  | X |  |
| Qualité, nombre d’illustrations |  |  | X |  |

### Pertinence des informations

Le cours est accessible à tous puisqu’il reprend l’ensemble des notions de bases de la programmation, de la création de variable jusqu’à des notions plus complexes comme la programmation orienté objet.

Chaque notion est contenue dans un chapitre détaillé avec une partie de théorie et un exemple afin d’illustrer le propos. Cependant des illustrations en couleurs auraient permis une meilleure lisibilité du code.

Mon étude s’est portée sur le chapitre du cours qui présente les objets, car malgré le fait que nous avons eu des cours sur la programmation orienté objet en C++, ce livre m’a permis d’identifier les différences entre le C++ et le Python et de découvrir les spécificités du Python comme l’usage du mot self ou encore l’utilisation de la fonction \_\_init\_\_(). Dans le cadre du projet, ces connaissances nous ont été très utile puisque la quasi-totalité du code issu du projet est en python et, nous avons quelques classes à l’intérieur tel que les classes IHM,Rotor,Vérin,Button il était donc primordial d’être à l’aise avec la programmation orienté objet de python.

**Note : 8/10**

### Illustration du cours étudié



## Les technologies de géolocalisation | Antonin Lyaët

### Introduction

Aujourd’hui, une des technologies les utilisés par tous est la géolocalisation. Même si elle utilisée par tous, peu de personnes ne connaissent le fonctionnement de cette technologie ainsi que les différents moyens de géolocalisation.

### Identification

Ce document est un article qui présente les principales technologies de géolocalisation, leurs avantages et leurs inconvénients. Cet article, posté sur « Organilog.com », s’intitule « Fonctionnement des systèmes de géolocalisation des smartphones » et vient énoncer les différents types de géolocalisation, les avantages et les inconvénients. Ce guide vient donc nous aider à répondre à la question suivante : « Quelles est la meilleure technologie de géolocalisation et comment fonctionnent-elles ? »

Lien : <https://fr.organilog.com/454-fonctionnement-geolocalisation-mobile/>

### Évaluation du contenu

|  |  |  |  |
| --- | --- | --- | --- |
| Critères | Faible | Moyen | Bien |
| Fiabilité |  |  |  |
| Illustration |  |  |  |
| Lisibilité |  |  |  |
| Pertinence |  |  |  |
| Accessibilité |  |  |  |

### Liste des mots clés

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| GPS | Géolocalisation | Satellite | Wifi | Technologie | GSM |
| Prix | Consommation | Précision | Fonctionnement | Intérieur | Extérieur |
| Vitesse | Matériel | Rural | Urbain | Batterie | Solution |

### Compte rendu

#### La géolocalisation

La géolocalisation est un système se situant dans beaucoup de technologie d’aujourd’hui comme les smartphones, les voitures, les drones, … Ce système est un concept très utilisé par tous mais peu de personne ne connaissent son fonctionnement. Mais connaitre les différents procédés de géolocalisation et leurs fonctionnements, cela permet une optimisation de la batterie pour les appareils qui utilisent ce système. Les 3 procédés les plus connus sont : le GPS (Global Positioning System), le GSM (Global System for Mobile) et le Wifi.

#### Géolocalisation par GPS

Ce système de géolocalisation doit disposer de deux éléments essentiels : un vue sur le ciel pour communiquer avec des satellites de géolocalisation ainsi qu’une puce GPS. Ce procédé géolocalise une puce GPS grâce à des signaux émis par un réseaux de satellites. Une fois que la puce intercepte un des signaux, elle renvoie ce signal au satellite qui l’a émis. Puis par triangulation des signaux renvoyés par la puce, on peut déterminer les coordonnées GPS de la puce.

Cette géolocalisation permet de géolocaliser une puce avec précision de 15 à 100 mètres mais des satellites européens, stellite Galileo, viennent d’être lancés et permettent une précision de 1 mètres.

Cette puce a un néanmoins deux gros défauts, elle consomme énormément de batterie. C’est pour cela que les concepteurs d’appareils équipés de ce système ont mis en place un moyen de l’activer ou non. Son deuxième gros défaut est que la réception et/ou le renvoie de signaux ne se fait pas correctement en intérieur

#### Géolocalisation par GSM

La géolocalisation par GSM fonctionne grâce au antennes GSM qui sont utilisés principalement pour la transmission de données aux téléphones ou voiture. Ce système fonctionne donc comme le GPS, avec le temps de réception en fonction des antennes qui les reçoivent ou émettent. Un algorithme de triangulation peut alors déterminer l’emplacement de l’appareil.

La précision dépend fortement de la densité des antennes présentent dans les environs. Elle est variée donc de 200 mètres à plusieurs kilomètres. Elle fonctionne donc beaucoup plus dans les milieux urbains que rural.

#### Géolocalisation par Wifi

Cette géolocalisation assez récente utilise des points d’accès Wifi connue pour déterminer là où l’appareil se trouve. C’est-à-dire à quelques dizaines de mètres. Il faut néanmoins que l’appareil en question soit muni d’une puce Wifi.

Cette puce consomme très peu et à une très bonne précision. Mais il faut être proche d’un accès Wifi connue pour que la géolocalisation par ce procédé se face. Ce concept vient à se répandre de plus en plus car Google vient référencer de nouveau point d’accès Wifi lorsque ses voitures utilisées par Google Street View passent devant de nouveau point.

#### Conclusion

D’après ce document, aucune des technologies de géolocalisation n’est mieux que les autres car elles dépendent toutes de là où on se situe et de comment on l’utilise. Pour avoir la meilleure géolocalisation possible, il faut combiner ces trois concepts de géolocalisation. Grâce à un algorithme prenant en compte ces système, on peut donc déterminer très facilement une géolocalisation très précise.

### Avis critique du document

Si je devais noter ce document, je lui mettrais la note de 7/10. Ce document est très complet, il nous apprend le fonctionnement des différents moyens de géolocalisation. Mais ce document est très court et manque d’illustration/schéma. Malgré ce gros défaut, il va tout de même droit au but et on comprend très bien comment fonctionne ces systèmes, leurs avantages et leurs inconvénients. Sa conclusion, rapide mais efficace, répond parfaitement à nos attentes même si nous n’avons pas de classement des meilleurs moyens de géolocalisation.

## Annexe 4

## Annexe 5



Figure Classe Algorithme

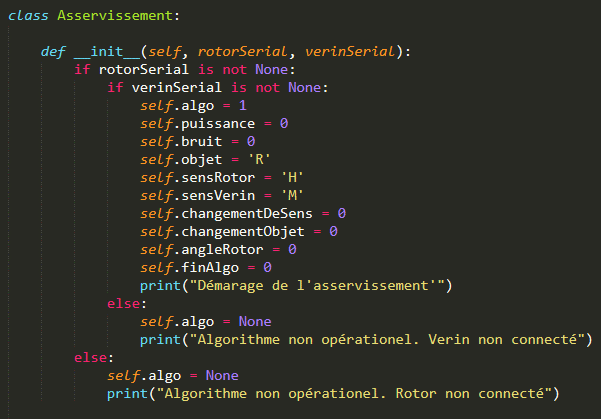


Figure Classe Asservissement

## Annexe 6

Figure Schéma de l'asservissement général

1. Série de bits organisés de façon à établir une communication [↑](#footnote-ref-1)
2. Transmission Control Protocol [↑](#footnote-ref-2)
3. Internet Protocol [↑](#footnote-ref-3)
4. Megabit [↑](#footnote-ref-4)