

Université François Rabelais

Institut Universitaire technologique de Tours

Département Génie Electrique et Informatique Industriel

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

ISS



D3R3

Thomas BARRIER
Olivier BERTHIAU
Augustin DECAUX
Grégoire REBOUL
Groupe 302
2017/2019

Projet tutoré 2^{ème} année
Mme GUIBLIN Nathalie
Mme AUGER Véronique
Mme VALLERY Sylvie
Coach : Mr GRIMAUD Vincent

Table des matières

Introduction.....	5
1. Diagramme de Gantt	6
2. Carte mentale.....	7
3. Budget.....	8
4. Partie mécanique	9
4.1 Choix du vérin.....	9
4.2 Choix de la carte d'interface du vérin	10
4.3 Rotor	11
4.4 Choix de la boussole	12
4.5 Position GPS du camion	13
4.6 Raspberry Pi 3b.....	14
5. Programmation	15
5.1 Documentation du code.....	15
5.2 Algorithme et programmation	16
Bilan.....	19
Conclusion.....	19
Table des illustrations.....	20
Annexe 1 Initiation à la programmation avec Python et C++ Olivier BERTHIAU	21
Identification	21
Liste des mots clés.....	21
Compte rendu	21
Évaluation du contenu.....	23
Pertinence des informations	23

Illustration du livre étudié	24
Annexe 2 Le Bus I2C (PDF) Thomas BARRIER.....	25
Identification	25
Liste des mots clés.....	25
Compte-rendu	25
Évaluation du contenu.....	27
Pertinence des informations	27
Annexe 3 Algorithmie Gregoire REBOUL.....	28
Identification	28
Liste des mots clés.....	28
Compte-rendu	28
Vocabulaire.....	29
Évaluation du contenu.....	30
Pertinence des informations	30
Illustration du livre étudié	31
Annexe 4 Télécommunication et infrastructure Augustin DECAUX.....	32
Identification	32
Liste des mots clés.....	32
Compte-rendu	32
Évaluation du contenu.....	34
Pertinence des informations	34
Illustration du livre étudié	35



Introduction

Dans le cadre du projet Super Pouet qui s'étend sur les semestres 3 et 4, nous devons dans un premier temps mettre en place l'automatisation du placement d'une parabole pour Strategic Telecom Sécurité Civile. Puis, dans un second temps, nous exploiterons la première partie pour une antenne râteau qui communiquera avec l'ISS (International Space Station).

Cette parabole sera située sur un camion, nous devrons alors mettre en place un algorithme permettant le positionnement automatique de la parabole quelle que soit sa position géographique.

Le système est équipé d'un vérin permettant de contrôler la parabole verticalement, mais aussi d'un rotor pouvant diriger la parabole horizontalement. Le code effectué sera programmé en langage python et exécuté sous Raspberry Pi.

Pour le moment, le système est fonctionnel manuellement. On peut donc piloter à l'aide de l'écran tactile la parabole en azimut et avoir un retour quant au degré d'inclinaison de celle-ci.

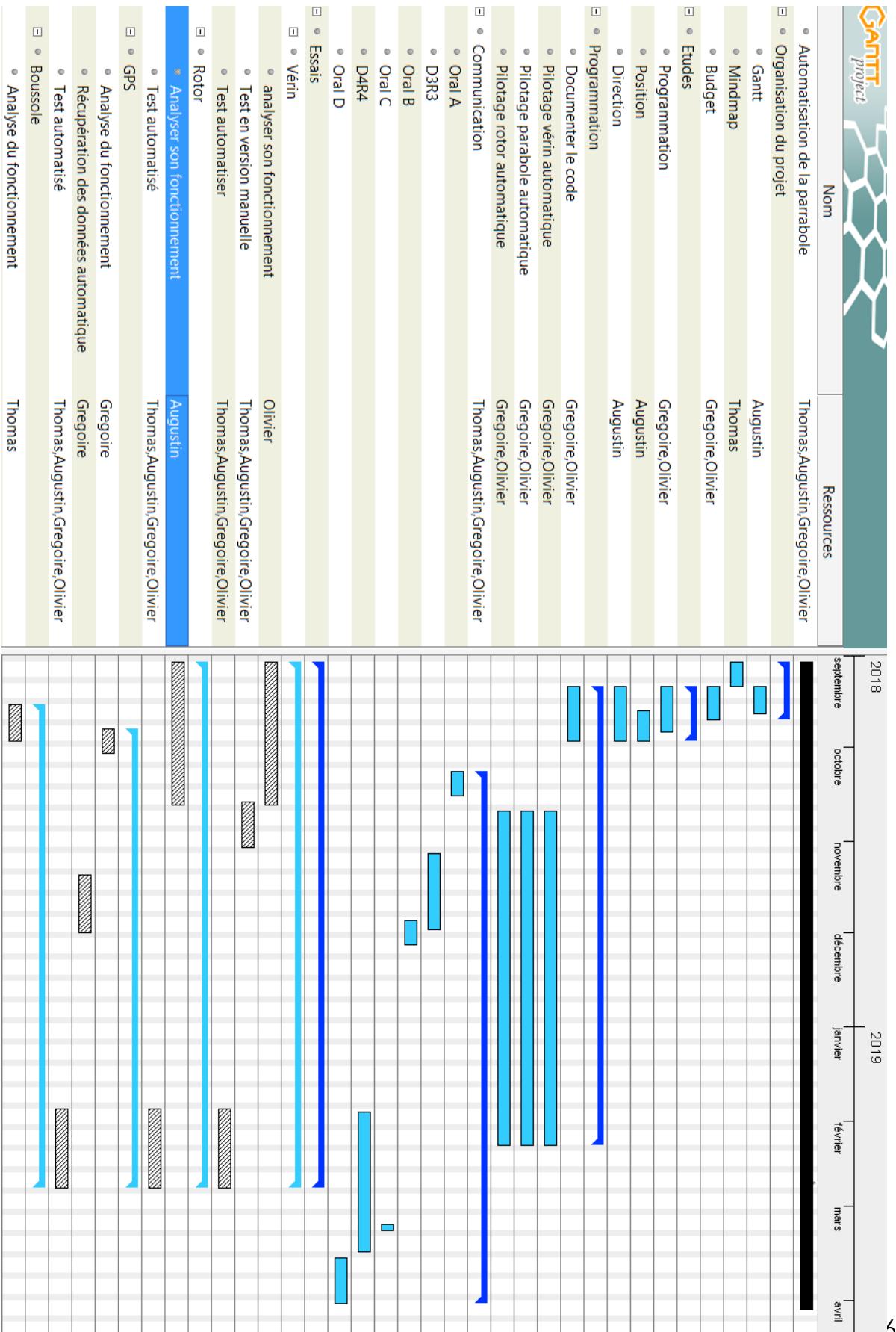
Nous sommes le groupe 302 composé de :

- **Thomas Barrrier**
- **Olivier Berthiau**
- **Augustin Decaux**
- **Grégoire Reboul**

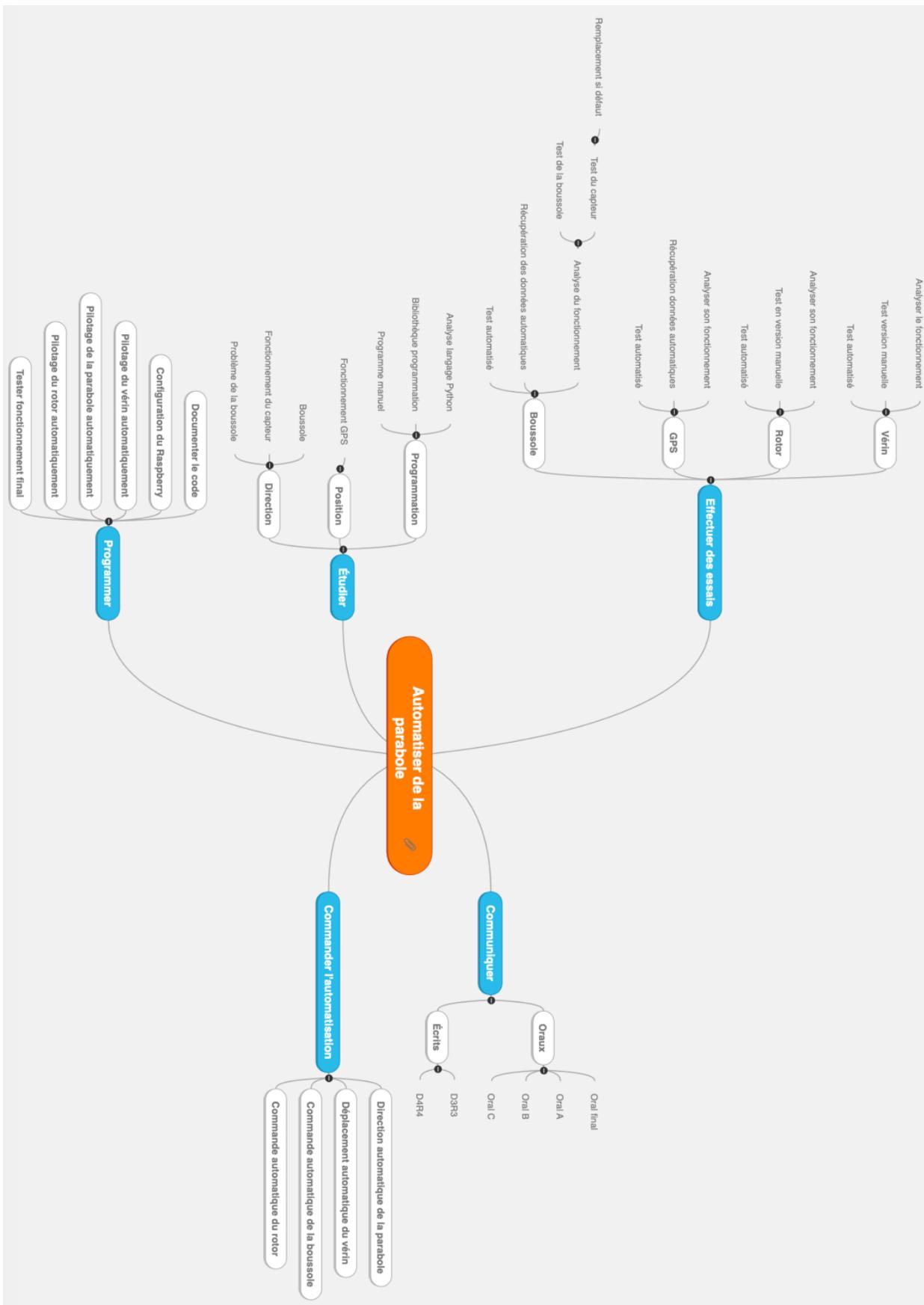


Nous sommes coachés par Mr GRIMAUD Vincent.

1. Diagramme de Gantt



2. Carte mentale



3. Budget

Fiche Budget Prévisionnel Projet

Groupe : 302

Matériel Fourni (*imposé dans le projet ou provenant de projets précédents*)

Désignation	Prix unitaire HT	Quantité	Prix total
Raspberry Pin B+	35,00	1,00	35,00
Alimentation Raspberry Pi	10,00	1,00	10,00
Carte micro SD 16Go Kingstone	8,00	1,00	8,00
Controleur de Rotor ERC-MINI V4 USB	110,00	1,00	110,00
Ecran tactile TFT 3,5"	37,00	1,00	37,00
Rotor Yaesu G-1000DX & controleur	490,00	1,00	490,00
Cable RJ45	5,00	1,00	5,00
Kit satellite Tooway	375,00	1,00	375,00
Abonnement Tooway	120,00	1,00	120,00
Vérin super Jack	45,00	1,00	45,00
Controleur de vérin	6,00	1,00	6,00
			1 241,00
			0,00

Matériel magasin

Matériel à commander

SOMME TOTAL DU PROJET 1 340 10

COUNT TOTAL 2 + TOTAL 3 99.10

4. Partie mécanique

Avant de nous consacrer pleinement dans l'automatisation de la parabole, nous avons d'abord dû étudier le choix des composants qui a été fait par le groupe précédent.

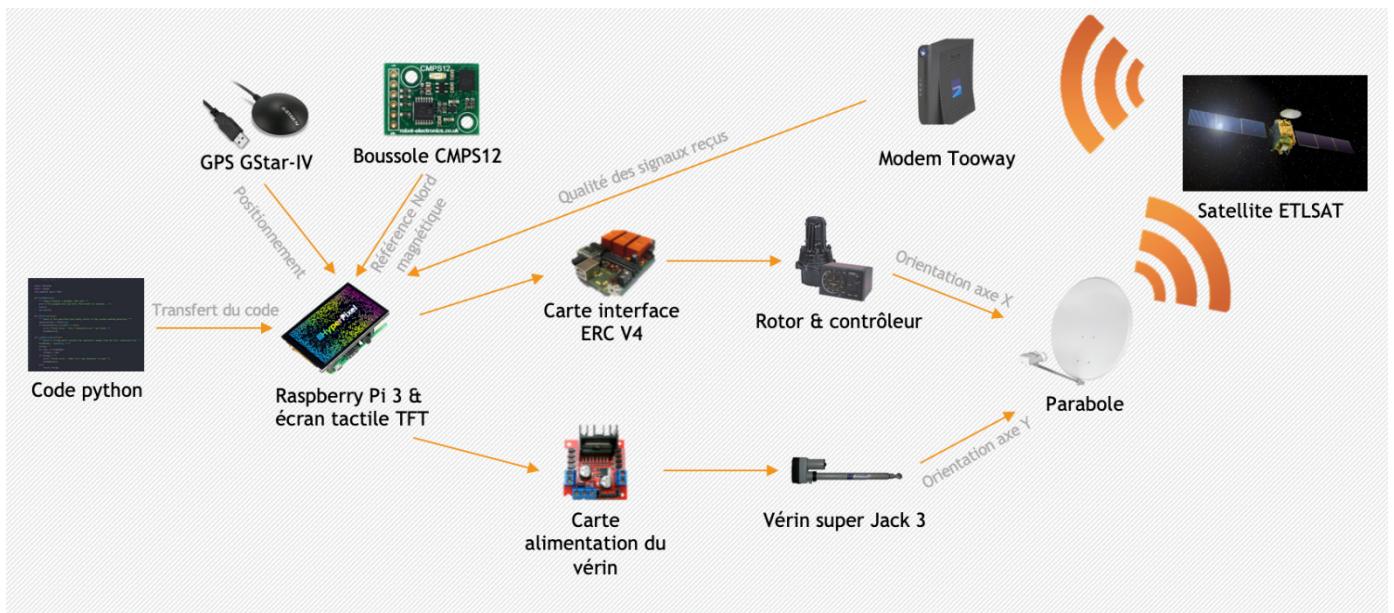


Figure 1 : Schéma structurel du système

4.1 Choix du vérin

Le vérin doit avoir une poussée suffisante pour soulever la parabole qui a une masse d'environ 15 kilogrammes. De plus, comme la parabole sera installée sur le camion, il faut que le vérin soit alimenté avec la tension pouvant être fournie par le camion soit 24 volts maximum.

Ainsi, le vérin qui a été retenu est le Super Jack III de chez Jaeger qui a les caractéristiques 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



Figure 2 : Photographie du Super Jack III

Nous avons effectué des tests pour confirmer ce qui a été conclu par le groupe précédent. Le vérin fonctionne bien avec une tension de 24 volts et émet 76 impulsions par pouce comme indiqué sur les caractéristiques du vérin.

4.2 Choix de la carte d'interface du vérin

Le vérin est contrôlé par une carte d'interface, le modèle utilisé pour le projet est le L298N.

Voici les caractéristiques de cette dernière :

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

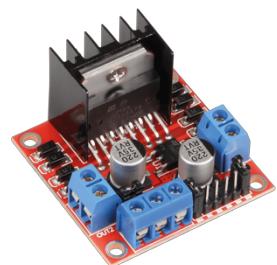


Figure 3 : Carte d'interface L298N

Le schéma de câblage du vérin sur la carte d'interface est représenté ci-dessous :

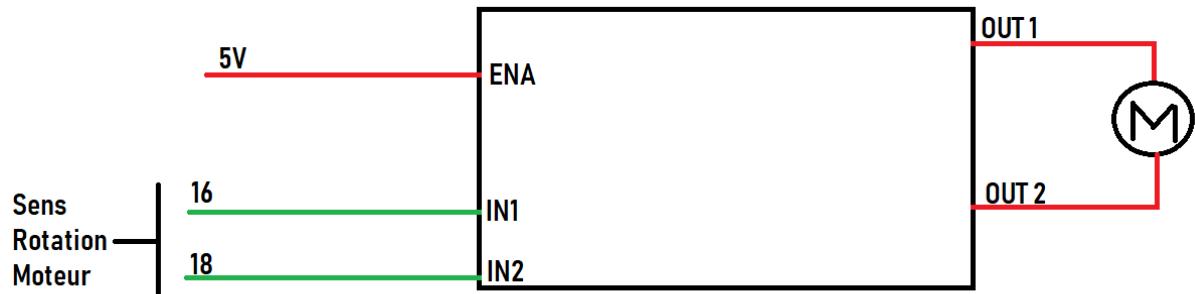


Figure 4 : Schéma de câblage du vérin

Cette carte répond parfaitement à nos attentes car on peut facilement changer le sens de rotation du moteur en envoyant un signal soit sur IN1 (sens horaire) soit sur IN2 (sens antihoraire).

4.3 Rotor

Le rotor qui est utilisé pour le projet est le YAESU G-1000DS. Ces caractéristiques sont les 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



Figure 5 : Rotor YAESU G-1000DS

La Raspberry Pi pourra communiquer en USB avec ce rotor grâce à une carte d'interface ERC version 4 qui est fournie. Les commandes se feront via le protocole YAESU GS-232A ou GS-232B.

4.4 Choix de la boussole

L'année dernière la boussole QMC5883 avait été utilisée pour connaître l'orientation du camion en fonction du Nord. Or, après un entretien avec le professionnel, il s'est avéré que la boussole avait un défaut de fonctionnement. En effet, celle-ci donnait de fausses informations sur l'orientation du camion et ne fonctionnait pas quand le camion était en pente . Nous avons donc décidé de changer la boussole pour répondre au cahier des charges du professionnel.

Le choix de la boussole s'est porté sur le modèle CMPS12.

Voici ses caractéristiques :

- Boussoles sur 3 axes (X,Y,Z)
- Gyroscope sur 3 axes
- Accéléromètre sur 3 axes
- Une précision de 99%
- Elle est alimentée en 3,3V en consommant environ 18mA. L'interface de connexion utilisée est un BUS I2C.

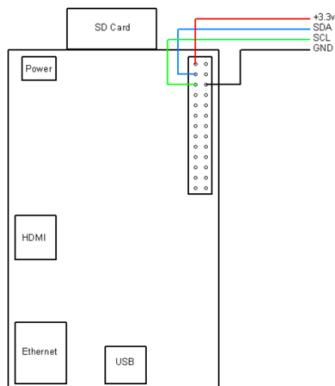


Figure 6 : Schéma d'implantation sur la Raspberry Pi

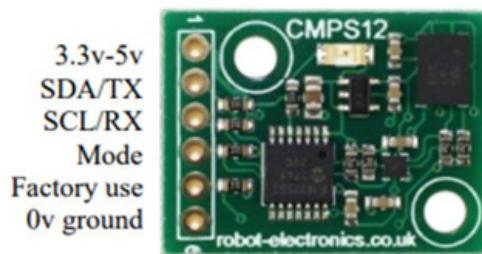


Figure 7 : Boussole CMPS12

Cette boussole est donc un excellent choix tant pour sa précision que pour ses options. Le gyroscope et l'accéléromètre nous permettront d'accroître la précision des informations renvoyées quelle que soit l'inclinaison du camion. Cela n'était malheureusement pas possible sur la boussole QMC5883.

4.5 Position GPS du camion

Un module GPS GSTAR IV a été fourni par la sécurité civile. Celui-ci sert à géolocaliser de façon précise le camion. Ce module est branché en USB au Raspberry Pi 3 afin d'émettre des trames selon un protocole standardisé (National Marine Electronics Association 0183).

Ces trames nous permettent d'obtenir des informations sur l'heure, les coordonnées en longitude et latitude, l'altitude du camion, la qualité du signal des satellites ainsi que la précision et la vitesse de réception.

Pour notre projet, nous utilisons la trame GGA pour obtenir les coordonnées, l'altitude, l'heure et le nombre de satellites captés.

Une fois la trame captée et exploitée, nous pouvons cartographier avec précision la position du camion en France pour ensuite nous connecter au satellite ayant le meilleur signal. Celui-ci est déterminé grâce à la carte des ellipses ci-dessous.

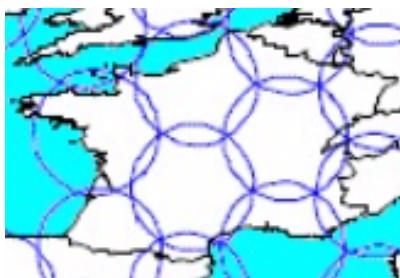


Figure 8 : Carte des ellipses satellites en France



Figure 9 : Module GSAR IV

Ensuite nous créerons un programme capable de déterminer le meilleur satellite en fonction des signaux reçus. Nous allons donc compléter le programme réalisé par le groupe précédent.

4.6 Raspberry Pi 3b

En reprenant la suite du projet de l'année dernière sur l'automatisation du positionnement d'une parabole, le choix du Raspberry Pi 3b nous a été imposé. La sélection du microcontrôleur s'est faite par rapport à sa puissance et ses caractéristiques présentent ci-dessous :

Raspberry Pi 3b
Processeur : 64-bit quad-core ARM Cortex-A53
Cadence : 1200MHz
RAM : 1024Mo
Bluetooth : Oui, 4.1
Alimentation : 5v 2.5A
Stockage : Carte MicroSD
Ports USB : 4
USB 2.5A : Oui

Ce microcontrôleur correspond à nos besoins car il est très peu couteux et fournit une puissance suffisante. Nous avons branché au Raspberry une dalle tactile pour contrôler la parabole. Le microcontrôleur communique avec le vérin et le rotor grâce aux GPIO de la Raspberry. Le GPS, lui, est branché en USB sur le Raspberry via une connexion série standard.



Figure 11 : Dalle tactile PiTFT



Figure 10 : Raspberry Pi 3b

5. Programmation

Dans cette partie nous allons détailler les différentes étapes de l'automatisation. Cette étude se décompose en deux parties : l'une traitant de la compréhension et de la documentation du code développé l'année dernière ; l'autre partie concerne les solutions trouvées pour l'algorithme.



Figure 12 : Logo Python

5.1 Documentation du code

Nous avons hérité de nombreuses lignes de code écrites dans le langage Python. Le langage utilisé n'est pas étudié à l'IUT mais il reste simple à comprendre et à utiliser. Pour documenter le code, nous avons chacun étudié minutieusement le D4R4 de l'an passé afin de comprendre comment tout cela fonctionnait. Pour continuer, nous avons regardé les programmes laissés par le groupe précédent pour ainsi le commenter ligne par ligne.

```
# ----- GPS ----- #
class GPS(object): # Création de la classe GPS
    def __init__(self, path): # Argument pour les fonctions utilisés par la class GPS (constructeur)
        try:
            self.gps = serial.Serial(path, 4800) # Liaison 4800 bauds série
            self.gps.readline() # On se place au début de la ligne d'une trame pour lire
        except serial.SerialException: # Le GPS n'est pas connecté
            self.gps = None # donc gps n'est pas connecté alors on retourne rien

        self.data = None # Si gps n'est pas connecté aucune data trouvé donc aucun retour

    def __del__(self): # creation de la destruction ( destructeur)
        if self.gps is not None: # Si gps est connecté alors on ferme le fichier gps
            self.gps.close() # fermeture de la trame gps

    def getInfos(self): # récupération les informations dans la chaîne de caractère
        if self.gps is not None: # Gps connecté
            while self.gps.in_waiting > 0: # Tant que le gps est en attente il lis la trame
                serialBuffer = self.gps.readline() # La mémoire série = à la chaîne de caractère lu sur la ligne
                if serialBuffer[0:6] == "$GPGGA": # compare la trame au la chaîne de caractère constitué de 7 caractères ($GPGGA la trame interprété par pynmea2)
                    self.data = pynmea2.parse(serialBuffer) # pynmea2 gère les trames ,parse = analyser la mémoire ( serial Buffer)

            return self.data # return les valeurs chaines de caractères
        else:
            return None # retourne rien si l'objet gps retourne rien
```

Figure 13 : Exemple d'un code commenté

Nous avons appris les normes et les spécificités du Python qui permettent de documenter le code plus facilement. Nous utiliserons ultérieurement l'outil Doxygen pour avoir une meilleure visibilité du code. Ce dernier permet de créer automatiquement des pages web (HTML, HyperText) pour de la

documentation logicielle. Chaque partie du programme aura sa propre documentation HTML ce qui facilitera la poursuite éventuelle du projet par d'autres étudiants.

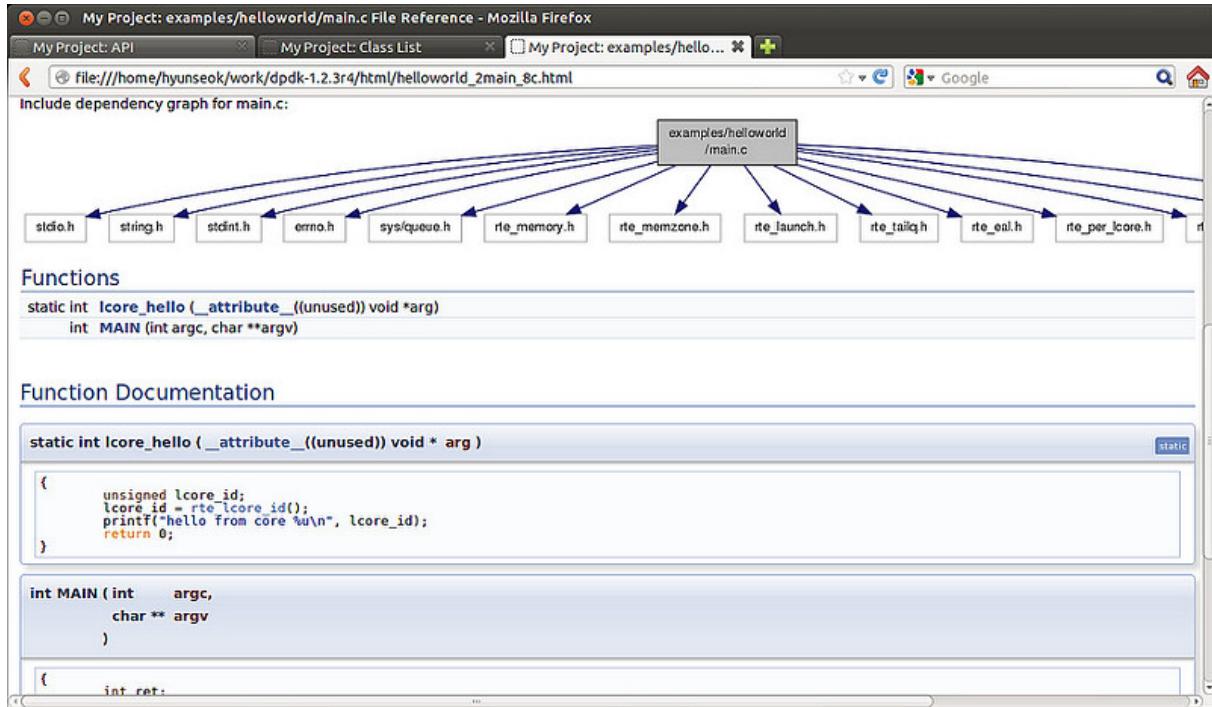


Figure 14 : Exemple de page générée par Doxygen

5.2 Algorithme et programmation

Nous devons automatiser le positionnement de la parabole à l'aide d'une interface graphique. Le contrôle manuel de la parabole a déjà été réalisé par le groupe précédent, nous devons automatiser son positionnement. Cette tâche est particulièrement complexe par la précision qu'elle nécessite (quelques degrés rendent la connexion au satellite impossible). Pour répondre à cette spécification, nous avons donc soumis des ordinogrammes de test comme celui-ci-dessous :

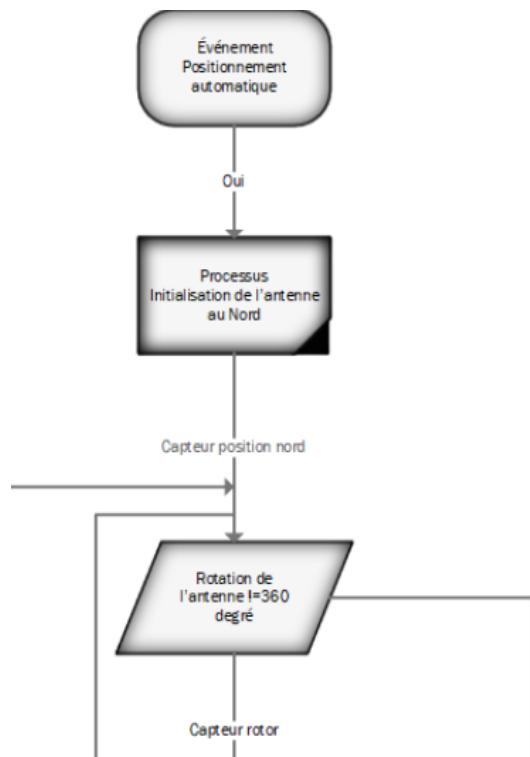


Figure 15 : Extrait d'un ordinogramme de test

A l'avenir nous allons utiliser un site web pour calculer l'élévation et l'azimut de notre antenne en fonction de la position géographique. Il permet donc à un utilisateur de savoir où diriger son antenne pour capter un satellite. Ce site nous aidera pour l'automatisation du placement de la parabole.

Ka-Sat Finder

L'application KA-SAT finder est destinée aux installateurs d'antennes satellites. Elle permet de déterminer les paramètres de pointage du satellite KA-SAT en fonction de la position géographique de l'antenne. KA-SAT finder détermine la couleur du spot de configuration et les angles d'azimut et d'élévation.

Carte **Coordonnées**

Plan **Satellite**

Google

Données cartographiques ©2018 Google, INEGI, ORION-ME Conditions d'utilisation

Résultats

Elévation

Azimut géographique

Configuration du spot

Cette zone n'est pas couverte par le satellite

Imprimer

Figure 16 : Site pour obtenir les coordonnées

Nous utilisons des bibliothèques qui effectuent les liens entre le Raspberry Pi et les différentes parties de notre système. La bibliothèque PySerial permet de faire le lien entre les GPIO de la carte et la partie commande (vérins...).



Figure 17 : Logo de la bibliothèque
PySerial

Pour la gestion des différentes trames GPS, nous continuerons d'utiliser la bibliothèque choisie par l'ancien groupe puisqu'elle facilite grandement le traitement des données envoyées dans ces trames. Cette bibliothèque se nomme pynmea2.

Enfin pour la partie communication avec l'utilisateur, nous conserverons la partie graphique déjà existante puisqu'elle est fonctionnelle et agréable. Pour cela le groupe précédent a utilisé la librairie PyGame. Cette dernière permet de créer une interface graphique simple d'utilisation. Dans notre cas le bouton d'automatisation étant déjà prévu sur l'écran nous n'avons pas besoin de toucher à cette partie.



Figure 18 : Logo de la bibliothèque Pygame

Bilan

A ce stade d'avancement du projet, nous avons étudié le choix de chaque composant et en avons déduit que la boussole qui était utilisée ne fonctionnait pas correctement. Nous avons donc remplacé cette dernière par une CMPS12 présentée précédemment. Par la suite, le code effectué par le groupe de l'année dernière a été commenté par nos soins afin d'avoir une meilleure compréhension du sujet. De plus, des recherches ont été menées afin d'optimiser l'automatisation de la parabole. Pour cela, nous avons exploité l'algorithme d'une page internet nous permettant d'avoir avec précision l'azimut et l'élévation nécessaire pour la connexion de l'antenne à un satellite. Enfin, des tests ont été réalisés après câblage du système afin de vérifier le bon fonctionnement du mode manuel.

De ce fait, il nous reste à intégrer la boussole dans le système et à tester l'algorithme permettant de connecter la parabole automatiquement au satellite.

Conclusion

La réalisation de cet avant-projet nous a fait prendre conscience de l'importance d'avoir un schéma électrique détaillé et une documentation technique précise. Le langage de programmation Python nous a été imposé, nous devions donc apprendre de manière autodidacte celui-ci. Nous avons également dû nous familiariser avec le Raspberry Pi 3 et avec le bus de communication utilisé pour interagir avec les différents composants du système. Ce projet nous a également permis d'approfondir nos compétences dans le travail d'équipe et dans le respect du temps.

Table des illustrations

Figure 1 : Schéma structurel du système	9
Figure 2 : Photographie du Super Jack III.....	10
Figure 3 : Carte d'interface L298N.....	10
Figure 4 : Schéma de câblage du vérin	11
Figure 5 : Rotor YAESU G-1000DS	11
Figure 6 : Schéma d'implantation sur la Raspberry PI	12
Figure 7 : Boussole CMPS12.....	12
Figure 8 : Carte des ellipses satellites en France	13
Figure 9 : Module GSAR IV	13
Figure 10 : Raspberry Pi 3b	14
Figure 11 : Dalle tactile PiTFT	14
Figure 12 : Logo Python	15
Figure 13 : Exemple d'un code commenté	15
Figure 14 : Exemple de page générée par Doxygen.....	16
Figure 15 : Extrait d'un ordinogramme de test.....	17
Figure 16 : Site pour obtenir les coordonnées	17
Figure 17 : Logo de la bibliothèque PySerial.....	18
Figure 18 : Logo de la bibliothèque Pygame	18

Annexe 1 | Initiation à la programmation avec Python et C++ | Olivier BERTHIAU

Identification

Auteur : Yves Bailly

Titre : Initiation à la programmation avec Python et C++

Edition : Pearson

Date de 1^{ère} parution : 26 mai 2008

Etude du chapitre 3 de la page 29 à la page 38

Liste des mots clés

- Fonction/procédure
- Instructions
- Incrémentation
- Définition de fonction
- Blocs d'instruction
- Déclaration de variable
- Portée d'une variable
- Valeur de retour

Compte rendu



- Définition
 - Une fonction permet de regrouper une suite d'instructions qui peut être exécutée à la demande de l'utilisateur
- Les fonctions permettent d'éviter de répéter plusieurs fois les mêmes instructions et avoir une meilleure compréhension du programme
- Pour créer une fonction, il faut d'abord écrire sa définition :
 - En Python : on utilise le mot `def` (pour `define`) suivi de parenthèses
 - En C++ : La définition de la fonction est différente selon ce que réalise la fonction

- Le corps de la fonction :
 - En Python : Les instructions de la fonction doivent absolument être indentées
 - En C++ : Les instructions de la fonction sont comprises entre 2 crochets et l'indentation est purement esthétique
- En C++, il est obligatoire d'avoir une fonction « main » pour exécuter les instructions lorsque le programme se lance
- Paramètres d'une fonction :
 - Les paramètres dans une fonction permettent de faire évoluer un programme beaucoup plus rapidement et simplement
 - Pour passer une variable en paramètre, il faut l'ajouter entre les parenthèses dans la définition de la fonction
- Blocs d'instructions :
 - Les blocs d'instructions sont utilisés pour exécuter un ensemble d'instructions se distinguant du programme
 - Une variable déclarée dans un bloc d'instruction n'est pas utilisable en dehors de ce dernier
- Retourner une valeur
 - On peut créer des fonctions qui renvoient une variable contenant une information qui peut être réutilisée dans d'autres éléments du programme
 - Cela permet de distinguer la partie interface qui communique avec l'utilisateur du noyau qui va effectuer les calculs
 - En C++ : on doit modifier la définition de fonction en fonction du type de variable qu'on envoie
 - En Python : la définition de la fonction ne change jamais quel que soit le type de variable renvoyé par celle-ci

Évaluation du contenu

Critère d'évaluation	Faible	Moyen	Bien	Très bien
• Degré de fiabilité de l'information				
• Utilité, accessibilité, pertinence des informations				
• Utilité des illustrations				
• Qualité, nombre d'illustrations				

Pertinence des informations

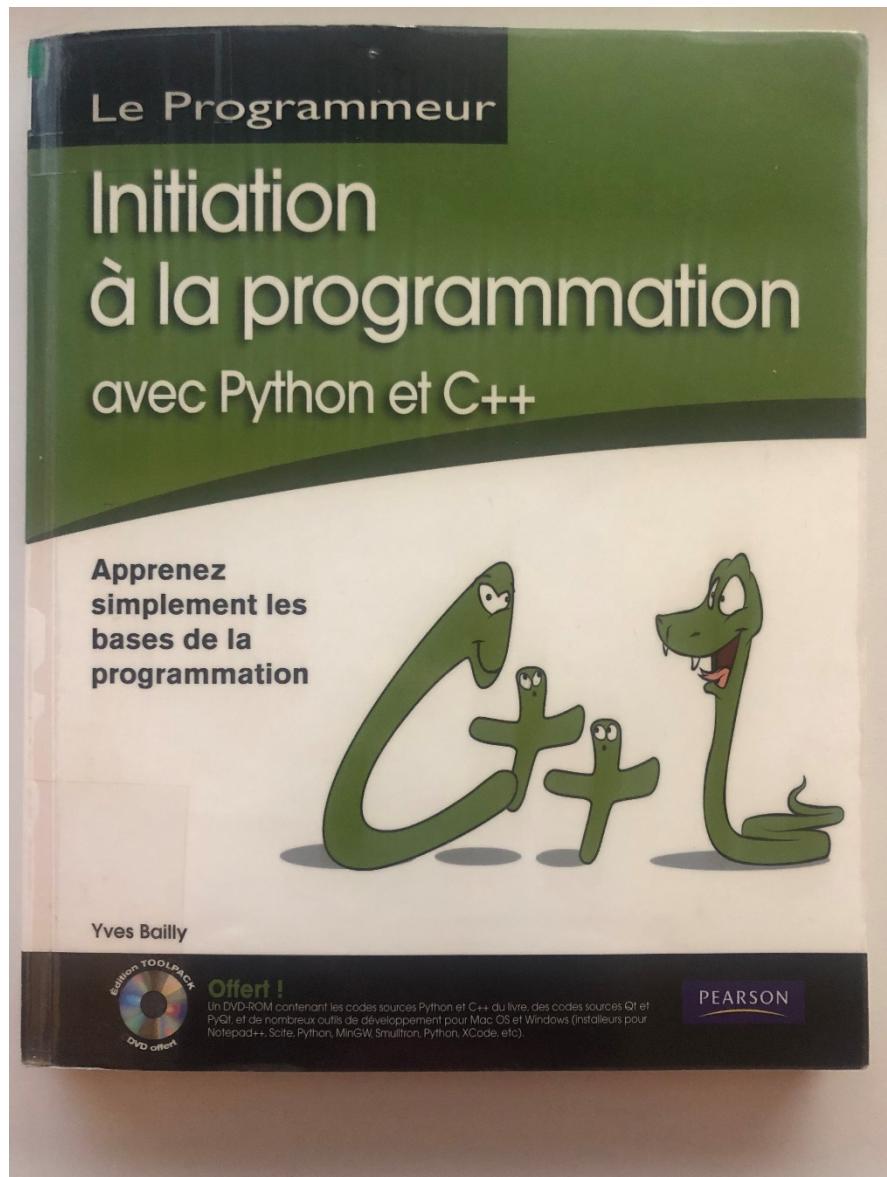
L'étude du livre s'est faite sur le chapitre 3 qui présente la notion de fonction dans un programme informatique. Les informations sont très claires et bien structurées à travers plusieurs sous parties.

Chaque idée est explicitée simplement et illustrée avec un exemple concret. Mon choix s'est porté sur ce livre car il montre les différences et les similitudes entre le langage Python et le langage C++.

Le livre date de 2008 mais cela ne pose aucun problème car ces 2 langages de programmation n'ont pas eu d'évolution majeure ces dernières années. Ainsi, cette étude était particulièrement intéressante car j'ai pu facilement me familiariser avec le langage Python qui m'était totalement inconnu auparavant grâce à mes connaissances solides en C++.

Note attribuée : 9/10

Illustration du livre étudié



Annexe 2 | Le Bus I2C (PDF) | Thomas BARRIER

Identification

Auteur : Inconnu

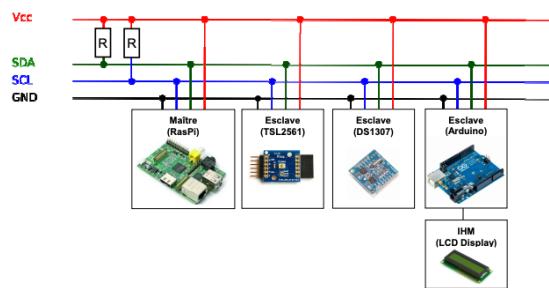
Titre : Transmission série : BUS I2C

Adresse du site : http://electroniqueveynes.free.fr/IMG/pdf/Bus_I2C.pdf

Date de 1^{ère} parution : 20 mai 2008

Liste des mots clés

-  Connexions
-  SCL (Serial Clock Line)
-  SDA (Serial Data Line)
-  Périphérique maître
-  Périphériques esclaves
-  Acquittement



Compte-rendu

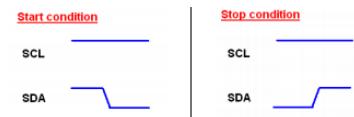


Définition du Bus I2C

- Deux connexions SCL et SDA → disponibles sur certaines broches
- Sur Raspberry PI 3B → broches GPIO2 et GPIO3 pour SDA et SCL
- Bus I2C → Périphérique maître
 - Coordonner des informations avec les autres périphériques
- Maître contrôle « Esclaves »
- Esclaves possèdent un numéro d'adresse
- Débit transfert de données (100kbts/s à 400kbts/s)
- Point fort du Bus I2C
 - Série bifilaire → ligne de données SDA, Horloge SCL
 - Bus multi maîtres
 - Acquittement* à chaque octet
 - Utilisation de circuit CMOS, TTL ou autres

Protocole du BUS I2C

- SDA et SCL niveau haut au repos (1) → possibilité de forçage à 0
- SDA et SCL niveau bas (0) → Impossibilité de forçage à 1
- Donnée valide quand SCL est au niveau haut
- Information début transmission : START
- Information fin de transmission : STOP
- Condition START :
 - SDA état bas et SCL état haut
- Condition STOP :
 - SDA état haut et SCL état bas
- Données → paquet de 8 bits, 1^{er} Bit = Bit de poids fort
- Chaque octet (8 bits) suivi d'un acquittement



Échange sur le BUS I2C

- 1^{er} Partie : Initier le dialogue
 - 1 - Maitre émet condition START
 - 2 - Maitre envoie adresse de l'esclave
 - 3 - Esclave répond
 - 4 - Maitre envoie adresse du registre
 - 5 - Esclave répond
- 2^{ème} partie : Transfert de données
 - 1 – Maitre émet condition START
 - 2 – Maitre envoie adresse de l'esclave
 - 3 – L'esclave répond.
 - 4 – L'esclave envoie contenu du registre
 - 5 – Le maitre répond
 - 6 – Le maitre émet condition STOP

Évaluation du contenu

Critère d'évaluation	Faible	Moyen	Bien	Très bien
• Degré de fiabilité de l'information				
• Utilité, accessibilité, pertinence des informations				
• Utilité des illustrations				
• Qualité, nombre d'illustrations				

Pertinence des informations

Ce document aborde avec technicité les fondamentaux du BUS I2C. Celui-ci est beaucoup utilisé dans le milieu informatique pour pouvoir connecter et faire dialoguer des composants entre eux.

Ce document est pertinent car il permet de compléter mes connaissances dans le domaine de l'informatique et ainsi permettre d'accroître mes recherches dans mon projet. En effet l'utilisation du BUS I2C sera utile lors de l'implantation de la nouvelle boussole au système. Celle-ci devra remplacer l'ancienne et transmettre ses informations au Raspberry Pi 3.

Cependant ce document n'est pas facilement accessible car il requiert des connaissances en terme d'informatique. Ces connaissances aident grandement à la compréhension de ce dernier.

Note attribuée : 7/10

Annexe 3 | Algorithmie | Gregoire REBOUL

Identification

Auteur : Thomas H.Cormen

Titre : Algorithmes

Edition : DUNOD

Date de 1^{ère} parution : 2013

Etude du chapitre 3 de la page 34 à la page 35

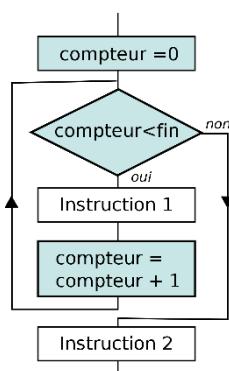
Liste des mots clés

- Pseudo code
- Algorithmes
- Alvéole

Compte-rendu



- Explication d'un algorithme type
 - Ouverture sur le tri par sélection → qui est le plus simple
 - Le plus souvent utilisé / moins rapide
 - Problématique tri alphabétique dans une librairie
 - En fonction du nom de l'auteur
 - Dans notre cas le premier est écrit par Louise May Alcott
 - Création d'un tableau à plusieurs cases
 - Chaque case contenant un chiffre (1...n)
 - On met Alcott dans cette première case
 - Rebalaye l'étagère pour trouver le deuxième auteur
 - Mise en place d'un nouvel élément dans le tableau
 - Répété cette action jusqu'à la fin du tri

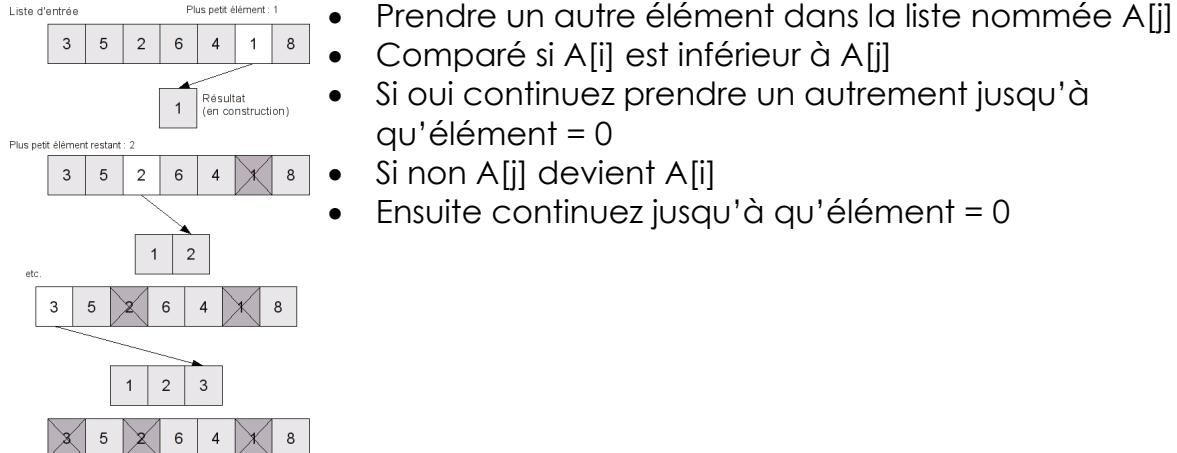


Conversion du langage en pseudo code

- Création d'une fonction comportant 2 éléments
 - 2 éléments un tableau est un les éléments du tableau
 - Tableau = étagère & éléments du tableau = ouvrage
 - A correspondant au Tableau et N éléments du tableau
- N est trier par ordre croissant dans le tableau A
- Explication de la boucle for (pour)
 - Dans cette boucle il y'a un i valant 1
 - Pour chaque tri il y'a N – i
 - On trie jusqu'à que N = 0

Inversion du pseudo code

- On cherche maintenant le plus petit élément
- Même type de raisonnement
- Balayer le tableau
 - Prendre un élément quelconque le nommer A[i]
 - Prendre un autre élément dans la liste nommée A[j]
 - Comparé si A[i] est inférieur à A[j]
 - Si oui continuez prendre un autrement jusqu'à qu'élément = 0
 - Si non A[j] devient A[i]
 - Ensuite continuez jusqu'à qu'élément = 0



Vocabulaire

Tableau : Est une structure de données qui consiste en un ensemble d'éléments ordonnés accessibles par leur indice (ou index). C'est une structure de données de base que l'on retrouve dans chaque langage de programmation.

Algorithme : Un algorithme est une suite finie et non ambiguë d'opérations ou d'instructions permettant de résoudre un problème

For(pour) : La boucle for est en informatique une structure de contrôle de programmation permettant de réaliser une boucle associée à une variable entière ou un pointeur qui sera incrémentée à chaque itération.

Évaluation du contenu

Critère d'évaluation	Faible	Moyen	Bien	Très bien
• Degré de fiabilité de l'information				
• Utilité, accessibilité, pertinence des informations				
• Utilité des illustrations				
• Qualité, nombre d'illustrations				

Pertinence des informations

Dans ce livre l'algorithmie y est présentée comme un cours. On y retrouve des formules, des exemples et même des exercices. L'aspect scientifique y est très présent comme le montre les notations.

Ce document explique la façon la plus simple de tri. L'explication est faite de différentes façons que cela soit par le langage français, le pseudo code ou même graphique par le dessin de fin page. L'avantage est que ce livre combine aussi bien l'aspect mathématique que pratique, en utilisant de nombreuses notations abstraites tout en gardant des exemples simples de la vie quotidienne (tri des livres).

Malheureusement ce livre n'est pas simple de compréhension car il demande des acquis dans de nombreux domaines que cela soit les mathématiques ou la programmation. Cela est accentué par la première de couverture et la préface mentionnant le niveau requis pour une bonne compréhension.

Note attribuée : 8/10

Illustration du livre étudié



Algorithme

Notions de base

Licence, IUT
Écoles d'ingénieurs

DUNOD

Annexe 4 | Télécommunication et infrastructure |

Augustin DECAUX

Identification

Auteur : Gérard Barué

Titre : Télécommunication et infrastructure. Liaison hertziennes spatiales optique

Edition : Ellipses

Date de parution : 2003

Page 247-251 Les liaisons spatiales

Liste des mots clés

-  Liaison spatiale
-  Satellite
-  Azimut
-  Propagation
-  Emission/Réception
-  Antenne



Compte-rendu

Eléments d'ingénierie d'une liaison par Satellite

- Généralités
 - Systèmes de télécommunication qui empruntent les satellites géostationnaires
- Géométrie d'un satellite avec un satellite géostationnaire
 - Schéma expliquant la liaison Terre/satellite
 - Schéma avec azimut, élévation, distance Terre-satellite

- Calcule de la distance de géosynchronisme
- Détermination de l'azimut, de la distance et de l'élévation
 - Calcule de l'angle dans le plan horizontal et verticale et de la distance entre la Terre et le satellite
 - Application numérique du schéma précédent

Architecture d'une liaison satellite

- Liaison par satellite
 - Schéma de cette liaison avec l'explication sur les bandes de fréquence plus élevé entre Terre et satellite qu'à la réception du signal
- Caractéristique générale du répéteur
 - Clef de voute du système (situé sur le satellite) → caractéristique du répéteur
 - L'opérateur du satellite renvoie des informations grâce au répéteur
 - Zone de couverture : espace que recouvre l'émission et la réception depuis le satellite
 - Bandes des fréquences montantes et descendantes
 - Répéteur constitué de plusieurs éléments
 - Amplificateur
 - Oscillateur

Évaluation du contenu

Critère d'évaluation	Faible	Moyen	Bien	Très bien
• Degré de fiabilité de l'information				
• Utilité, accessibilité, pertinence des informations				
• Utilité des illustrations				
• Qualité, nombre d'illustrations				

Pertinence des informations

Ce livre m'a permis d'en apprendre plus sur les liaisons entre la Terre et les satellites. Il nous informe sur plusieurs éléments importants de notre projet. La première partie du document donne certaines informations quant à l'élaboration d'un algorithme permettant une liaison satellitaire. Afin d'apporter des connaissances à notre étude, les définitions et propriétés qui y sont écrites sont pertinentes. Les informations globales du document sont accessibles pour toutes personnes travaillant dans le domaine de la physique.

La note de 9 a été attribuée. Cette note est justifiée notamment parce que l'auteur du livre est un ingénieur spécialisé dans le domaine des liaisons hertziennes et optiques. Les schémas sont accessibles et ils illustrent bien les explications apportées.

Note attribuée : 9/10

Illustration du livre étudié

