Cahier de Bord projet CSF Yann Brault

1er séance, 11/03/19 : Aujourd’hui, première séance du projet, avec mon binôme Antoine Cousson nous avons décidé de réaliser un ballon météo pour notre projet. Il devra récupérer différentes données telle que la température, la lumière, l’humidité, la pression barométrique ou encore la vitesse du vent. Ces données seront transmises via le protocole LoRa à un serveur d’application qui lui aura une API Tweeter pour tweeter la météo tous les jours. Ce système sera alimenté sur batterie, et celle-ci sera rechargée par énergie solaire.

2nd séance, 22/03/19 : Aujourd’hui, seconde séance du projet, nous avons créé un Git, mis en ligne sur GitHub, qui sera notre classeur pour le projet, tous les plannings de répartitions, présentations et autres documents tels que les codes y seront stockés. Une fois cela fait nous avons démarré nos recherches de composants qui a été fructueuse puisque nous avons trouvé une carte embarquant les capteurs pour toutes les données que nous voulons récupérer. Il nous faut rajouter dessus un anémomètre pour la vitesse du vent.

3 -ème séance, 25/03/19 : Aujourd’hui troisième séance du projet, on s’est entrainé à la modélisation 3D en modélisant un jeton de cadis, puis nous avons fait une première ébauche de la nacelle. Nous n’avons pas encore décidé du matériaux de la nacelle, on ne trouve pas de matériaux léger et suffisamment résistant. Il va également falloir percer les parois de la nacelle pour pouvoir récupérer les données. Pour donner suite à la première modélisation, j’ai découpé des parois test pour simuler la boite et prévoir l’agencement. Le professeur M.Ferrero nous a également trouvé une nouvelle carte, pour contrôler et réguler l’énergie. Le liPo Rider Pro permet de contrôler un flux d’énergie arrivant d’un panneau solaire et de la rediriger vers une batterie et/ou la carte arduino. Aucun code n’est nécessaire pour cette carte c’est très pratique.

4-ème séance, 05/04/19 : Aujourd’hui quatrième séance du projet, Antoine a produit un premier bon modèle 3D de la nacelle, les parois sont percées par des hexagones afin d’avoir le meilleur rendement possible en élasticité/résistance. Nous avons également trouvé le matériaux de la boite, un dérivé du polystyrène, beaucoup moins friable et plus résistant, et il peut être découpé au laser. En parlant de code, Antoine a produit un premier code mais qui ne fonctionne que pour l’arduino puisque nous n’avons pas encore le sparkfun Weather Shield le code ne peut pas encore être produit. Enfin les composants ont été commandé aujourd’hui

5-ème séance, 08/04/19 : Aujoud’hui cinquième séance du projet, c’est la présentation intermédiaire, nous avons fait un powerpoint avec la présentation du projet, les objectifs, le planning et la répartition du travail.

6-ème séance, 15/04/19 : Aujourd’hui sixième séance du projet, la plus grosse phase commence. Nous avons découpé toutes les parois de la nacelle, en perçant les panneaux latéraux avec des formes hexagonales pour une légèreté et une ouverture accrue, tout en conservant une résistance significative. Nous réfléchissons donc maintenant à comment assembler la boîte. Colle, vis ou clou à voir. La plupart des composants sont arrivés et ils sont bien. Le plus important est là, le weather shield. L’anémomètre est là également, et il est particulièrement léger c’est un très bon point. Antoine a produit les attaches pour relier les câbles au ballon, en deux parties et une sera amovible, il faudra bien qu’on pense à coller la seconde partie au plafond de la nacelle.

7-ème séance, 19/04/19 : Aujourd’hui septième séance du projet, nous avons assemblé la boîte. Pour relier les parois entre elles, Antoine a eu la bonne idée d’utiliser des scratchs autocollants. Très pratique pour ouvrir la boîte si besoin est, peu encombrant, léger et ça tient très bien. On a récupéré la cellule solaire qui est plutôt grande, elle fait la taille d’un panneau latéral, et on a également la batterie qui, inversement, est vraiment très petite. Je suis un peu en retard je commence toute la partie serveur d’appli, DB et Graphe dans les jours à venir.

8-ème séance, 26/04/19 : Aujourd’hui huitième séance du projet, j’ai enfin fait le tutoriel sur Node-red et Influxdb, l’utilisation semble relativement simple pour ces nœuds-là, l’utilisation est très appréciable et les possibilités assez complètes, mais je ne sais pas du tout comment marche le nœud twitter et j’ai bien peur ce que ce soit relativement plus complexe que les nœuds ttn.

9-ème séance, 29/04/19 : Aujourd’hui neuvième et dernière séance de projet, l’utilisation du nœud twitter est vraiment beaucoup plus complexe que ce à quoi je m’attendais et c’est problématique que je découvre ça à aussi peu de temps du rendu j’ai très mal géré cela.

Pour le nœud twitter, il faut des clés d’API et des clés de Token pour associer le compte Twitter au nœud, mais ces informations ne sont uniquement disponibles par un compte twitter développeur dont il faut faire la demande à twitter. La démarche est complexe mais une fois faite pas de problème de connexion. Sur ce point le problème est réglé, mais twetter se révèle très très dur. Un simple lien entre le bloc ttn et celui de twitter ne fonctionne pas, une erreur de transmission de données, plus précisément sur le type des données. Il faut utiliser une fonction, à écrire donc en JS, afin de convertir le type des données. J’ai réussi à twetter mais le filtrage que j’effectue n’est pas le bon.

Dernière séance, 03/05/19 : Aujourd’hui dernière séance de projet, c’est le jour du rendu. La première transmission OTAA de la board de la nacelle a été effectuée hier soir. Antoine a réglé le programme pour qu’il envoie une chaîne de 32 caractères. Mais les données sont contenues dans un buffer donc leur extraction va être différente de ce que j’ai réussi à faire hier, je m’explique. Hier j’ai enfin réussi à filtrer et formater de la bonne manière les informations reçues de ma board. Mais le message capté comporté différents champs qui eux-mêmes étaient composés de sous champs donc je stockais dans mes variables les extractions de valeurs que je faisais par suite de noms, « a : msg.payload.luminosity ». Or ici, aujourd’hui je reçois mes 32 caractères dans une liste et ils sont en code ascii. Donc il faut les extraire par index, comme en python finalement, et il faut également convertir la valeur sortie en string, donc par la fonction String.fromCharCode(). Et là, la magie opère. En plaçant dans les variables les bons caractères et en n’oubliant pas de les formater à l’aide d’un bloc de Template, les données sont postées sur Twitter sans problème.