

## Description du projet pour la JDP

Durant la période de Octobre à Mai, nous avons choisi de travailler sur un projet proposé par Thales s'intitulant : Développement d'un radar millimétrique pour la détection et le suivi de cibles. Pour rappel, Thales Groupe est une société française spécialisée dans cinq domaines principaux, l'aéronautique, le spatial, les transports terrestres, la sécurité et la défense. L'entreprise est présente dans 68 pays dans le monde et compte 80 000 employés. Le projet sur lequel nous avons travaillé a été proposé par Brus Antoine, notre référent chez Thales, dont l'un de ses domaines d'expertise est le radar.

L'objectif était alors de développer des traitements radars comme la détection et le suivi de cible, en utilisant du matériel déjà existant, mais néanmoins commercialisé récemment. En effet, pour ce projet nous avons utilisé la carte boost awr1843 sortie en août 2019. Ce genre de carte est développé chez Texas Instrument depuis des années et la nôtre en était l'évolution la plus récente.

Le challenge était alors d'une part d'appréhender la théorie derrière le radar à un niveau plus mathématique, et d'autre part, de se documenter un maximum pour comprendre le fonctionnement de la carte et de l'API et ainsi y implémenter tout le traitement.

Parlons maintenant un peu de l'organisation de l'équipe ! En effet, nous étions sept personnes à travailler sur ce projet : Sébastien ROUX, Clément CANTON, Anojan SRITHARAN, Sami BEYAH, Ashvith ALAGARAJAH, Prince SATKUNARAJAH, Viknesh-Raj PALANI.

Au cours de l'année, nous nous sommes tous documentés sur la théorie du radar et sur la carte. Vers Décembre, nous avons commencé à tester la carte que nous avons reçue fin novembre. Puis nous nous sommes séparés en deux sous-équipes.

La première était chargée de trouver une méthode afin de compiler notre futur code et le faire fonctionner avec l'existant. En effet, TI fournit bien un code démo, que nous comptons modifier pour intégrer nos traitements. D'un point de vue extérieur, cette tâche peut paraître anecdotique, cependant l'architecture du code de TI est complexe et modulaire. Cela implique d'avoir beaucoup de dépendance dans les fichiers et il n'est donc pas si aisé de retrouver tous les morceaux.

La seconde équipe devait quand à elle programmer les traitements sur Matlab. En effet, nous n'avons pas tout de suite choisi de coder en C avec l'API de TI. Cela permettait de s'affranchir dans un premier temps, de l'appréhension des fonctions implémentées par TI et de coder rapidement un traitement. Cela a été rendu possible grâce au code démo fourni par le constructeur. En effet, il nous a permis de mettre en route la carte et de tester ses fonctionnalités et même d'enregistrer des mesures de l'environnement. Par la suite, nous appliquions le code Matlab sur ces mesures pour produire des courbes et déduire la position d'un objet.

Finalement, nous avons réussi à programmer une partie des traitements et à trouver une méthode ainsi que les fichiers pour compiler du code sur la carte. Il ne nous restait donc plus qu'à programmer avec l'API de TI pour implémenter ce que nous avons fait sur Matlab. Le plus gros obstacle durant le projet a été, il nous semble, l'accès et la compréhension de la documentation.

