

# ATISE

Soutenance Finale

Adrien ARTAUD  
Myriam LOMBARD  
Killian PAREILLEUX  
Alexandre SALMON

# Sommaire

- Rappel du sujet et cahier des charges - 3
- Technologies employées - 5
- Architecture techniques - 6
- Réalisations techniques - 7-9
- Axes d'amélioration - 10
- Gestion de projet - 11-14
- Outils - 15
- Métriques logicielles - 16-17
- Conclusion - 18
- Démonstration - 19

# Rappel du sujet : ATISE

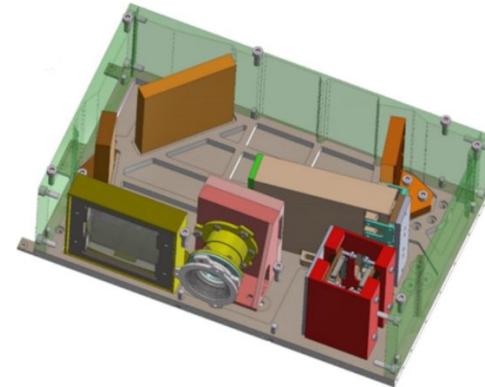
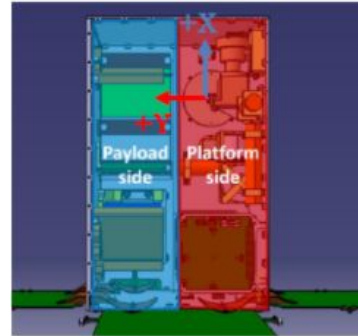
## Aurora Thermosphere Ionosphere Spectrometer Experiment

Le satellite:

- Format 12U (20cm x 20cm x 30cm)
- 1 appareil photo OnyX
- 3 Spectrographes HDPyx

La mission de ATISE:

- Observer les aurores boréales depuis l'espace
- Mieux comprendre la magnétosphère et l'activité solaire



# Rappel du sujet : ATISE

## Cahier des charges

**Objectif : Permettre la communication via UART entre la partie capteur et la partie communication du satellite.**

Contraintes :

- le code doit être sous FreeRTOS (voir Technologies et outils utilisés)
- le code doit utiliser le protocole CSP
- programmations et tests avec la carte MARS ZX3

Résultats attendus :

- Envoi de fichiers (images) fonctionnel par UART, conformément à nos contraintes.

# Technologies et outils utilisés

- **FreeRTOS**

*Système d'exploitation en temps réel pour micro-contrôleurs*



- **UART**

*Composant utilisé pour faire la liaison entre l'ordinateur et le port qui nous intéresse. Nous l'avons testé avec des Raspberry Pi.*



- **Git**

*Outil de gestion de versions*



- **VIVADO**

*IDE avec des outils de niveau système électronique*



- **Cubesat Space Protocol**

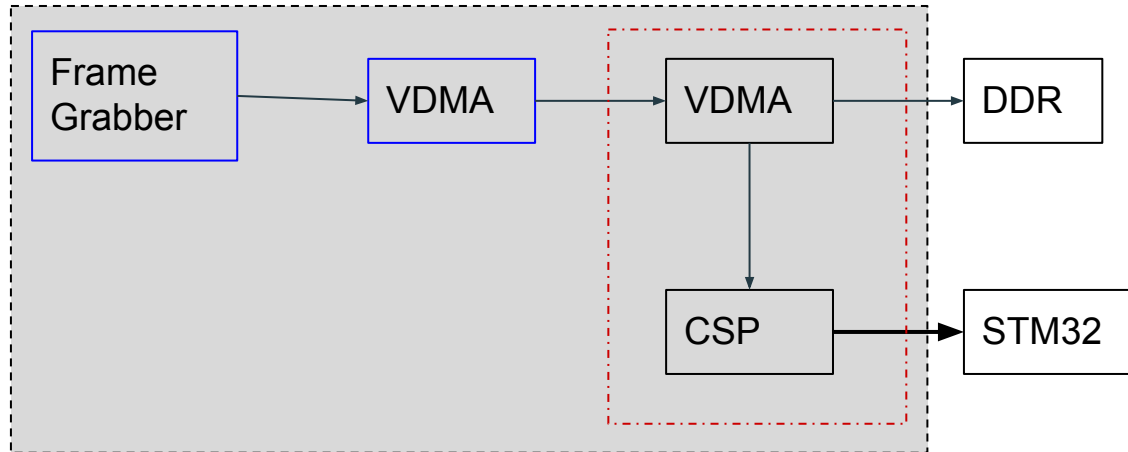
*Protocole permettant de simplifier la communication entre systèmes embarqués au sein d'un petit réseau*

- **Eclipse**

*Environnement de développement*



# Architecture d'ATISE



Logique FPGA

Processing System: ARM Cortex A9

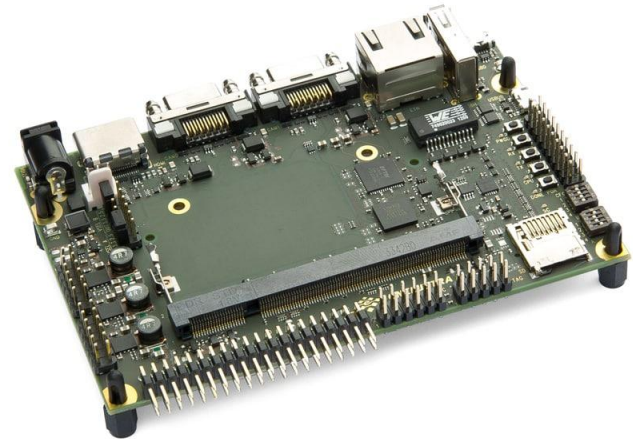
# Réalisations techniques (1/3)

## État initial du projet :

- Code C existant
- Carte électronique Mars ZX3

## Deux tâches principales :

- Gérer la communication avec la carte MARS ZX3
- Comprendre comment faire fonctionner l'UART et le CSP ensemble, puis le tester en conditions réelles



Carte électronique MARS ZX3

# Réalisations techniques (2/3)

## Gérer la communication avec la carte MARS ZX3

### Étapes suivies :

- Adapter le code C en FreeRTOS
- Questionnement et Spécifications
- Échange de chaînes de caractère avec la carte électronique via l'UART
- Choix de Conception
- Transfert de fichiers plus volumineux comme des images au format PGM



# Réalisations techniques (3/3)

Comprendre comment faire fonctionner l'UART et le CSP ensemble, puis le tester en conditions réelles

- Réalisation d'une liaison UART entre 2 RaspBerries PI
  - Test des librairies requises pour communiquer en liaison UART
- Test de code utilisant le CSP comme mode de communication
  - De nombreux paquets requis pour utiliser libcsp
- Difficultés rencontrées...

# Axes d'amélioration

- Parties à terminer
  - Corriger l'instabilité de l'uart (1 octet absent sur 50)
  - Rajouter le protocole CSP au code source
  - Vérifier le débit de transfert
- Parties à ajouter
  - Faire le lien avec le traitement de l'image
  - Ajouter une UART customisée sur les broches de la carte pour enlever la limitation de fréquence.

# Gestion de projet

## Equipe & Organisation

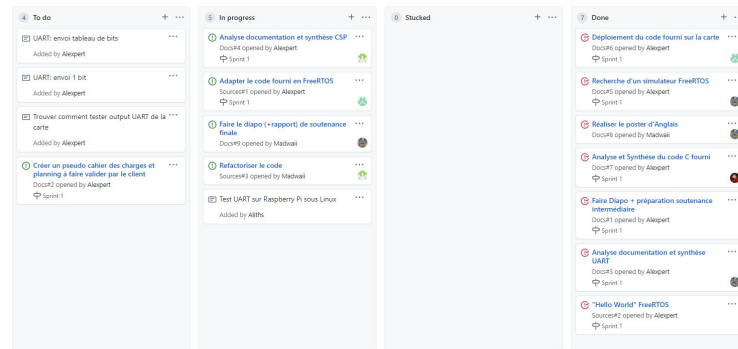
Scrum Master : Killian Pareilleux

Product Owner : Alexandre Salmon

Développeurs : Adrien Artaud et Myriam Lombard

Méthode Agile :

- Quasi Daily Meetings
- Découpage du projet en sprints
- Organisation avec outils de git (issues, Kanban, plusieurs repository, etc.)
- Rencontres à Polytech deux fois par semaine



# Gestion de projet

## Planning prévisionnel

1. Adapter le code en FreeRTOS (**Sprint 1 - du 08/02 au 21/02**)
2. Échanger des données avec la carte via l'UART (**Sprint 2 - du 22/02 au 07/03**)
  - a. Transfert d'un seul bit
  - b. Transfert d'un tableau de bits (structure de données)
3. Utiliser le CSP avec l'UART (**Sprint 3 - du 08/03 au 19/03**)
4. Optimiser le débit (Si il y a le temps)
5. Prendre en compte le traitement d'images (Si il y a le temps)

# Gestion de projet

## Planning effectif

**Du 08/02 au 21/02 :**

- Sprint 1

**Du 22/02 au 07/03 :**

- Sprint 2
- Poster d'Anglais
- Prise en main des Raspberries

**Du 08/03 au 19/03 :**

- Sprint 3
- Bugs

**Légende :**

**Vert :** Comme prévu

**Jaune :** Travail non prévu

**Rouge :** En retard

# Gestion de projet

## Gestion des risques

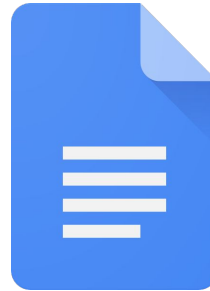
Difficultés rencontrées :

- Carte MARS ZX3 unique
  - Partage de la carte compliqué avec peu de présentiel
  - -> Recherche d'un simulateur FreeRTOS infructueuse. Mise en place de pair programming. Test de la liaison UART entre 2 Raspberry Pi, sans la Mars ZX3.
- Prise en main de l'environnement
  - Technologie inconnue
  - Architecture de la carte
  - -> Recherche de documentations sur les différentes technologies et partages de connaissances lors de nos réunions en présentiel.
- Problème de contexte
  - Problèmes d'envois de données (driver)
  - Configuration des machines

# Outils

## Outils de collaboration :

- Discord : Meetings, discussions, partage de ressources
- Google Drive/Docs : Création de documents collaboratifs
- Github : Gestionnaire de versions et de projets d'équipe



# Métriques logicielles

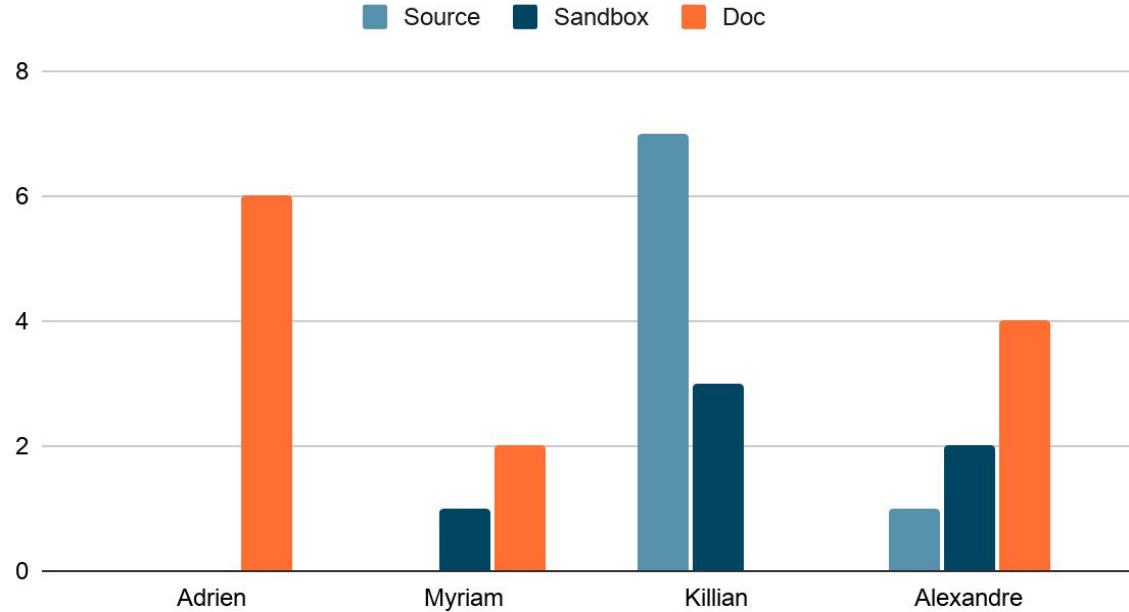
## Languages





# Métriques logicielles

Commits par Repo



# Conclusion

- Projet formateur mais limité en efficacité à cause du covid
  - Bugs qui ont fait perdre beaucoup de temps
  - Bonne gestion du projet, avec le plus de meetings possibles entre nous, et des réunions avec notre responsable en cas de besoin.
  - Nous sommes contents d'avoir appris de nouvelles choses et d'avoir pu participer à un vrai projet de grande envergure.
- > Beaucoup de temps dépensé pour apprendre les technologies.
- Regret : peu de place à la créativité dans ce projet : les séances de MPI ne nous ont que très peu servi.

# Démonstration

Merci pour votre attention