

# ATISE

Soutenance Intermédiaire

Adrien ARTAUD  
Myriam LOMBARD  
Killian PAREILLEUX  
Alexandre SALMON

# Sommaire

- Présentation d'ATISE.....	3
- Architecture d'ATISE.....	4
- Equipe et Organisation.....	5
- Technologies et Outils.....	6
- Etapes du Projet.....	7
- Backlog et Avancement.....	8
- Difficultés rencontrées.....	10

# ATISE

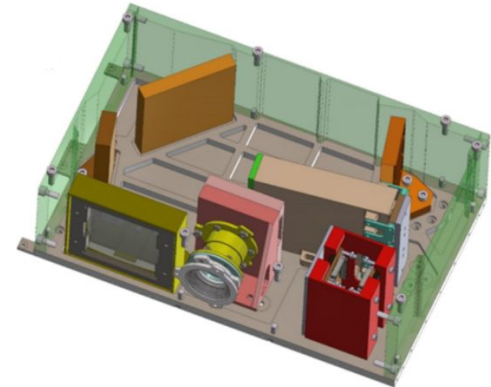
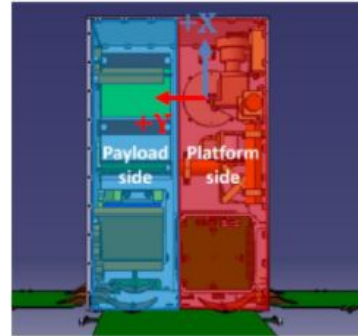
## Aurora Thermosphere Ionosphere Spectrometer Experiment

Le satellite:

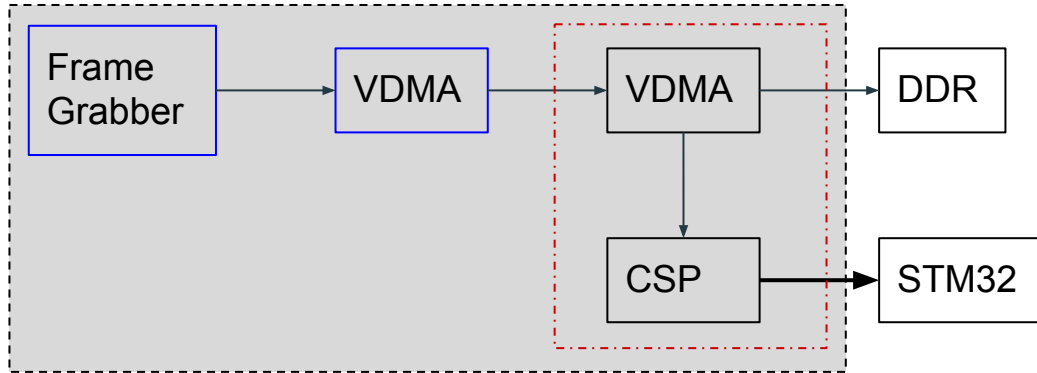
- Format 12U (20cm x 20cm x 30cm)
- 1 appareil photo OnyX
- 3 Spectrographes HDPyx

La mission de ATISE:

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



# Architecture d'ATISE



Logique FPGA

Processing System: ARM Cortex A9

Notre mission:

- Permettre la communication via UART entre la partie capteur et la partie communication du satellite
- Porter le code pour FreeRTOS

# 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.)

# 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*

- **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*



# Jalons - Grandes étapes du projet

1. Adapter le code en FreeRTOS (**Sprint 1**)
2. Échanger des données avec la carte via l'UART (**Sprint 2**)
  - 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**)
4. Optimiser le débit (Si il y a le temps)
5. Intégrer le traitement d'images (Si il y a le temps)

# Backlog / Avancement

Ce qui est fait :

- Documentations
  - CSP
  - UART pour FreeRTOS
- Recherche d'un simulateur FreeRTOS
- Analyse et Synthèse code C fourni
- "Hello World" FreeRTOS
- Prise en main du logiciel Vivado
- Début de l'adaptation du code FreeRTOS



# Backlog / Avancement

Ce qui reste à faire :

- Tester Output UART de la carte
- Envoi d'un tableau de bits
- Rajout de l'utilisation du protocole CSP
- Mesure du débit

# Difficultés rencontrées

- Carte MARSXZ3 unique
  - Partage de la carte compliqué avec peu de présentiel
  - -> Recherche d'un simulateur FreeRTOS ?
  
- Prise en main de l'environnement
  - Technologie inconnue
  - Architecture de la carte
  - etc.

Merci pour votre attention