

# Fin du sprint 3 => release du premier prototype python EVEEX

Auteur : Guillaume LEINEN

## Introduction

Ce compte rendu à pour but de résumer ce qui a été fait au terme du 3ème sprint, en particulier il détaille le fonctionnement du premier prototype fonctionnel en python.

Vous pouvez retrouver le code du projet à l'adresse suivante : [github](https://github.com)

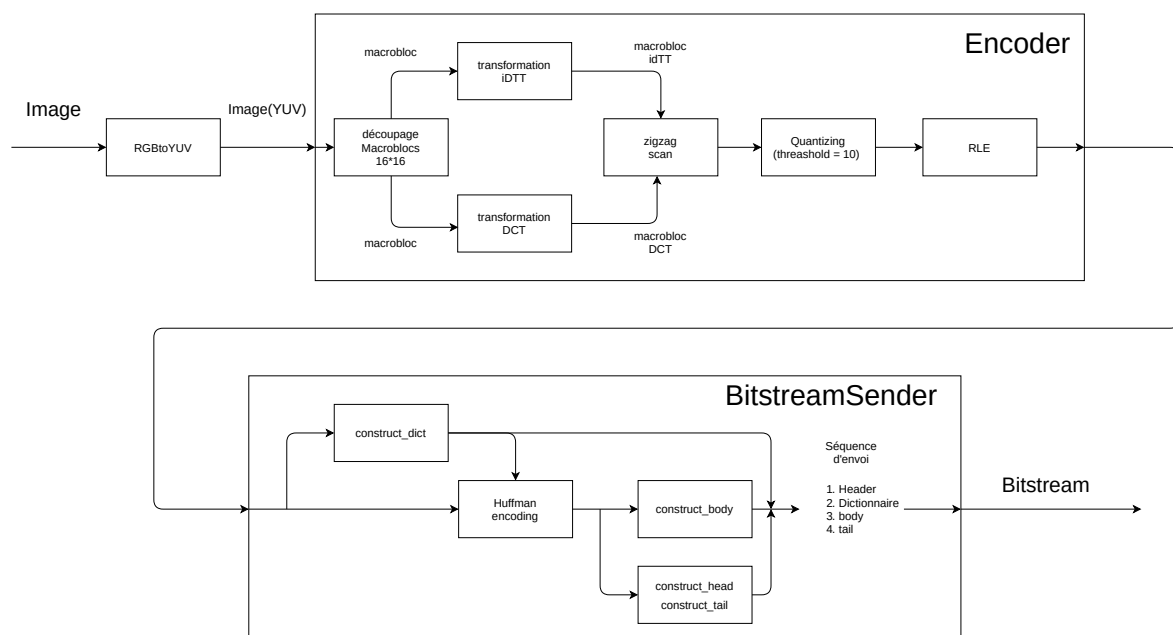
## Présentation de l'algorithme final

Ce 3ème sprint servait principalement à terminer la première version de l'encodeur/décodeur EVEEX en python.

**Les fonctionnalités de cette algorithme sont les suivantes :**

- Utilise une image au format jpg comme source de départ, de taille standard 16/9 (480p, 720p, etc..)
- Permet de séparer cette image en macroblocs 16x16.
- Comprime l'image par un algorithme proche du MJPEG, macroblocs par macroblocs
- envoie un bitstream encodé par un arbre binaire par la méthode Huffman à un client
- réceptionne le bitstream chez le client
- le décode grâce au dictionnaire envoyé dans le bitstream
- affiche l'image ainsi envoyé.

Le fonctionnement de l'algorithme d'encodage est détaillé dans ce diagramme (le décodeur reprend toutes les étapes dans le sens inverse)



Fonctionnement schématique de l'encodeur EVEEX en python

# Utilisation de l'algorithme

---

Pour en simplifier l'utilisation, il est possible d'utiliser le programme en ligne de commande

---

*extrait du fichier README*

## Installation

Before you begin, make sure you have a minimal python 3.8 installation on your system.

The first step is to clone the project and go into the project's folder.

```
git clone https://github.com/EVEEX-Project/EVEEX-Code.git
cd EVEEX-Code/python_prototype
```

The next step is to make sure you have all requirements already installed. You will need the `pip` tool that is often already installed with python (if not please [check the documentation](#) for the extra steps).

```
python -m pip install -r requirements.txt
```

This line will install every requirements in the text file in order to be able to launch the project.

## Usage

By simply typing `python eveex.py` you will be greeted with an helper message letting you know which command is available.

## Encoding an image

One can generate an image from a JSON description file for example. Let's save this image as `test.png`.

In order to encode the image and save it as a bitstream file called `bitstream.txt` one can use the following command.

```
python eveex.py -e test.png bitstream.txt
```

---

## Performance de l'algorithme

Nous sommes encore en train d'implémenter d'autres métriques, néanmoins nous sommes en mesure de dire que

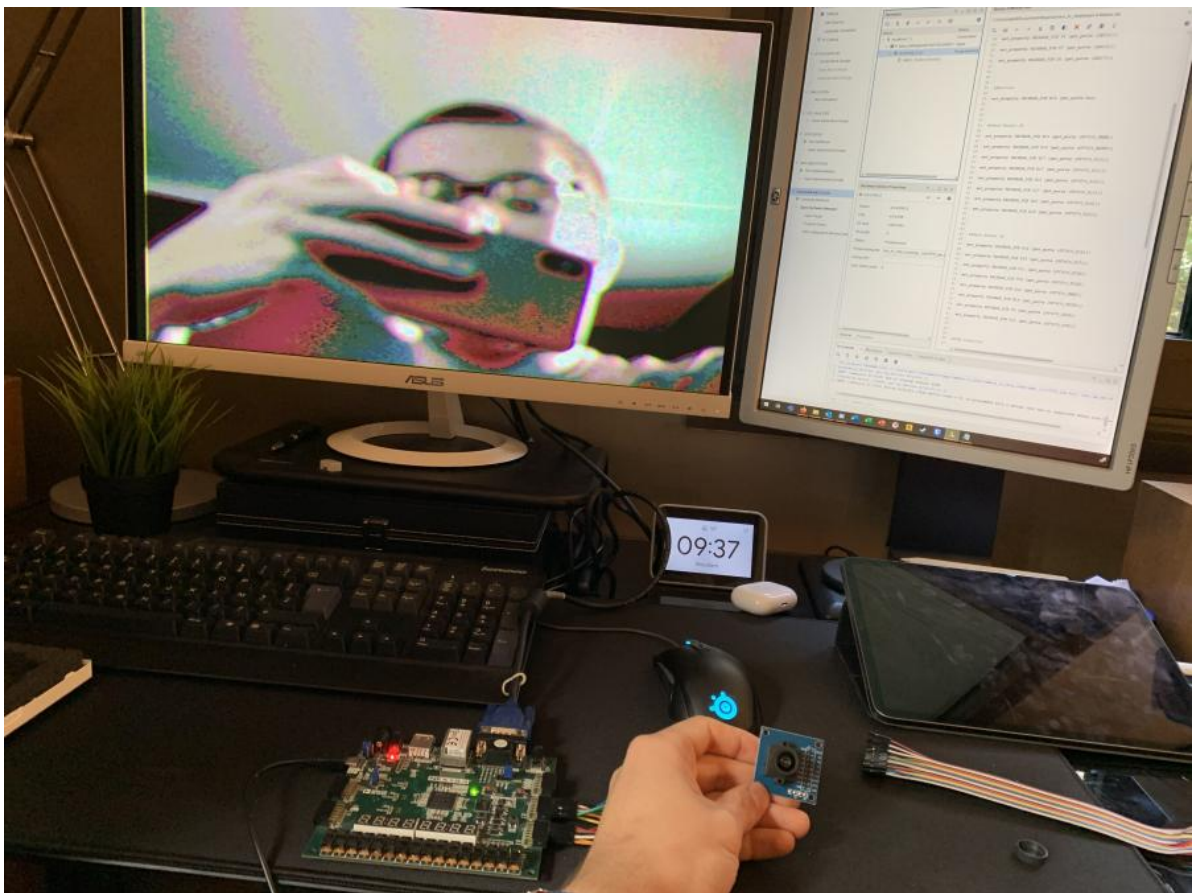
**Le taux de compression varie entre 5:1 et 10:1** pour des images comme celle ci :



Au niveau de la qualité, l'image transmise est **indiscernable** de l'image originale.

## Au niveau du matériel

Grâce à un code adapté de celui-ci ([lien](#)), nous avons été en mesure de récupérer une image d'une camera ov7670 et de la transmettre en VGA sur un écran.



Comme vous pouvez le constater, les couleurs sont délavées et imprécises, j'ai donc essayer d'échanger des pins de données entre eux pour réparer les couleurs.

Évidemment, j'ai dans l'opération inversé par mégarde des pins d'alimentation ce qui a eu pour effet de tuer la caméra (avec la fumée et le pschitt réglementaire, je pense qu'une résistance à servi de fusible, malheureusement je ne parvient pas à identifier de dégâts visuels)

La carte va bien, et je me suis permis de recommander 3 autres cameras (1 euro l'unité). je ferais attention à l'avenir.

Mise à part ça, il existe du code VHDL permettant d'effectuer un scan zigzag et une RLE => [lien](#) , ainsi qu'un codage Huffman => [lien](#)

## Suite des opérations

---

Nous commençons à présent la traduction de ce prototype python en c, avec comme première étape la fabrication des types non-natifs et la définition de la structure de fichiers.

En parallèle, nous avançons sur d'autres fonctionnalités en python, notamment le découpage dynamique en macroblochs (taille variant en fonction du contenu), ainsi que la définition "statistique" du dictionnaire, ou encore la compression du dictionnaire en lui-même.

Concernant les manip FPGA, nous avons comme objectif d'avoir un premier jet VHDL avant la fin d'année, l'objectif est volontairement optimiste afin d'avancer le plus possible.