

6 avril 2023

# Rapport de Projet

Pilotage d'une matrice de micro-miroirs



Rémi Truong, Jérôme Touquet, Samar Rabe, Guillaume Chapelant

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Cahier des charges . . . . .	2
1.2	Guide d'utilisation . . . . .	2
1.2.1	Chargement des images sur le DMD . . . . .	2
1.2.2	Choix du dossier d'enregistrement des images de fluorescences . . . . .	2
1.2.3	Lancement de l'acquisition . . . . .	3
1.2.4	Temps d'exposition . . . . .	3
1.2.5	Quitter et nouvelle acquisition . . . . .	3
1.3	Schéma Fonctionnel . . . . .	3
<b>2</b>	<b>Description du prototype</b>	<b>3</b>
2.1	Objectifs . . . . .	4
2.2	Réalisation / Choix technologique . . . . .	4
2.3	Tests de validation et résultats . . . . .	4
<b>3</b>	<b>Bilan</b>	<b>4</b>
3.1	Avancement final . . . . .	4
3.1.1	Déclenchement du DMD avec une source externe . . . . .	5
3.1.2	Déclenchement de la caméra avec une source externe . . . . .	5
3.2	Retour d'expérience . . . . .	6

# 1 Introduction

Au cours de ce projet PROTIS nous avons créé une interface dans le but d'automatiser l'acquisition de données avec le DMD pour l'expérience de PIMS Biophotonique MIS. Pour cela nous avons travaillé avec Python pour communiquer avec le DMD et la caméra.

## 1.1 Cahier des charges

Le cahier des charges initial nous donnait des contraintes de fonctionnement du système :

1. Communication avec le DMD (chargement d'images et envoie d'ordres)
2. Communication avec la caméra (Prise d'images de fluorescence)
3. Présence d'une interface graphique

De plus, au milieu de notre projet, on nous a demandé un autre objectif optionnel qui était de pouvoir contrôler le déclenchement de chaque image sur le DMD et la prise des photos avec la Nucléo car il existait déjà un programme sur la Nucléo qui ajustait l'exposition de l'éclairage pour avoir une horloge commune. Les contraintes attendues sont dûes à la compréhension de la documentation pour communiquer avec le DMD, mais également de pouvoir garantir la synchronisation entre tout les périphériques.

## 1.2 Guide d'utilisation

Nous allons maintenant présenter l'utilisation de notre interface.

### 1.2.1 Chargement des images sur le DMD

Au lancement du programme, l'interface graphique s'ouvre (1)

Les deux voyants rouges : DMD déconnecté et Caméra déconnecté deviennent vert une fois la caméra et le DMD connectés. Dans un fonctionnement normal, ces voyants passent au vert automatiquement. Si ce n'est pas le cas, il faut vérifier dans un premier temps les connections au PC, puis dans un second temps vérifier les Vendor et Product ID.

Une fois que ces témoins sont verts, ils devient possible de charger des images dans le DMD.

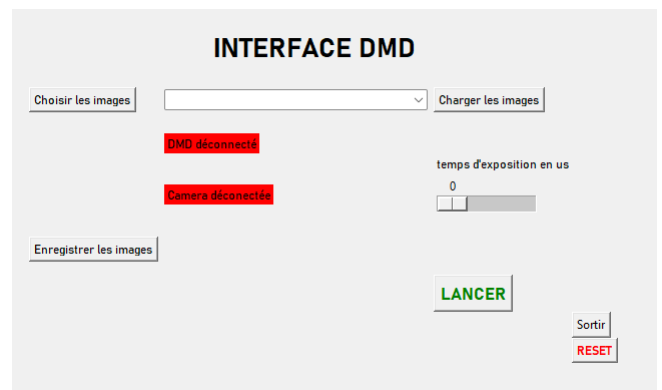


FIGURE 1

Le chargement de ces images dans le DMD s'effectue grâce à la première ligne de l'interface. Le premier bouton Choisir les images, permet d'ouvrir l'explorateur de fichier pour sélectionner des images. Un fois les images sélectionnées, leurs adresses sont visibles en cliquant sur la flèche à droite de la barre. Puis, pour les charger dans le DMD, il suffit de cliquer sur le bouton Charger les images.

### 1.2.2 Choix du dossier d'enregistrement des images de fluorescences

Afin de choisir le dossier d'enregistrement des images, le bouton Enregistrer les images, ouvre l'explorateur de fichier. Un fois le dossier sélectionné, son adresse apparaît à droite du bouton Enregistrer les images. Le nom des images sera alors donné en prenant en compte l'image sélectionnée sur le DMD (ex :nom\_fichier\_fluorescence).

### 1.2.3 Lancement de l'acquisition

Une fois que les images ont été chargées dans le DMD, et que le dossier d'enregistrement des images de fluorescence a bien été renseigné, il ne reste plus qu'à cliquer sur le bouton LANCER, pour démarrer le protocole d'acquisition des images.

### 1.2.4 Temps d'exposition

Une échelle sur l'interface permet de régler le temps d'exposition sur l'intervalle possible de la caméra ce qui permettra de changer le temps d'exposition mais aussi la fréquence des images sur le DMD.

### 1.2.5 Quitter et nouvelle acquisition

Une fois l'acquisition terminée, il suffit, pour relancer une acquisition, d'utiliser le Bouton RESET.

Enfin, pour quitter le programme correctement, il faut utiliser le bouton Sortir, prévu à cet effet.

## 1.3 Schéma Fonctionnel

Le diagramme de la Figure 2, expose le principe de fonctionnement que nous avons adopté pour notre interfaçage. Sans rentrer dans les détails, il rend compte de la démarche que nous avons adoptée pour organiser notre interface.

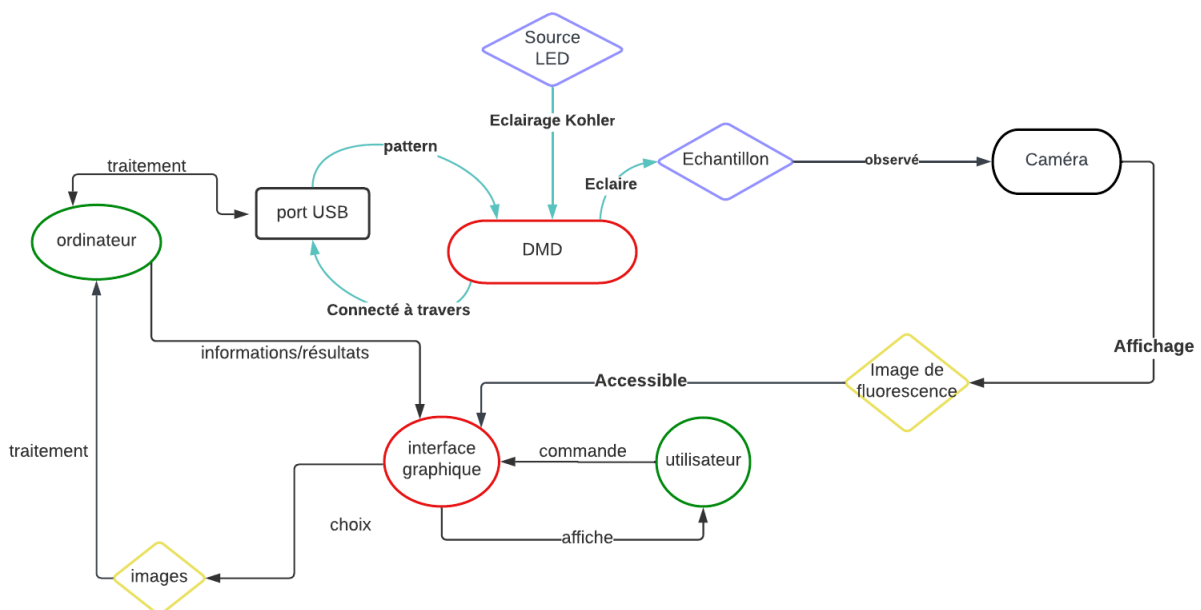


FIGURE 2 – Diagramme fonctionnel

## 2 Description du prototype

Le coeur de notre projet était de comprendre comment communiquer avec le DMD via Python ainsi que créer une interface qui permet de piloter tous les éléments de l'expérience et donc simplifie sa manipulation pour l'utilisateur.

## 2.1 Objectifs

L'objectif principale de notre travail était de réussir à contrôler de DMD à l'aide d'un langage de programmation (Python) et donc définir une classe permettant de contrôler la puce DLP6500 du DMD de Texas Instruments. Ce travail revient à créer les différentes fonctions nécessaires pour la manipulation du DMD permettant de réaliser les tâches demandées.

Nous avons construit notre programme sur les bases données par : Paolo Pozzi sur <https://github.com/csi-dcsc/Pycrafter6500>.

Le second objectif était de pouvoir commander la caméra à partir du même langage de programmation et donc effectuer l'acquisition des photos résultant de l'expérience.

Afin de pouvoir communiquer avec la caméra, nous avons principalement utilisé le Git-Hub de BASLER qui donnait d'excellents exemples d'utilisation. ([https://github.com/basler/pypylon/blob/master/samples/save\\_image.py](https://github.com/basler/pypylon/blob/master/samples/save_image.py)).

Le dernier objectif était de pouvoir tout connecter à une seule interface graphique qui pourra faciliter la manipulation pour l'utilisateur. Cette interface devra donc regrouper toutes les commandes réalisables sur l'expérience ainsi que la sauvegarde des données.

## 2.2 Réalisation / Choix technologique

Sur le plan pratique, la réalisation de nos objectifs a nécessité d'effectuer un choix judicieux concernant le type de d'envoi des commandes entre DMD et l'ordinateur. Deux options étaient possibles : utilisation d'un circuit I2C ou une liaison USB pour manipuler la matrice de micro miroirs. On a choisi de travailler avec la configuration où le DMD est connecté par la voie USB à l'ordinateur. Ce choix est plus simple à mettre en oeuvre et nous permet de d'effectuer les tâches nécessaires pour le bon fonctionnement de la manipulation. En suite, le deuxième choix consistait à choisir le temps d'acquisition de la caméra pour qu'il soit synchronisé avec le temps d'affichage de chaque motif sur le DMD. Même si nous n'avons pas pu le réaliser, notre choix s'est porté sur l'envoi de Trigger sur les différents composants pour qu'ils aient une horloge commune via la Nucléo.

## 2.3 Tests de validation et résultats

L'étape finale du projet consistait à relier les codes qui génèrent l'interface aux codes de commande du DMD et valider le fonctionnement final du projet. Pour tester le bon fonctionnement, on a générer les images de mires de différentes fréquences à l'aide du DMD. Le set-up a bien fonctionné et on a pu voir le résultat espéré. Le choix des mires provient du fait que la manipulation consiste à éclairer les cellules biologiques de manière à pouvoir au final, estimer leur tailles. En effet, c'est à partir du résultat de l'observation des cellules fluorescentes à travers l'éclairage modulé par une mire d'une certaine fréquence spatiale qu'on peut estimer ça taille.

# 3 Bilan

## 3.1 Avancement final

A la fin du projet, une première version de l'interface a été fourni. Elle permet actuellement de vérifier le cahier des charges en communiquant avec la caméra et le DMD à travers une interface graphique. Cependant, nous pouvons poursuivre l'automatisation de l'ensemble en intégrant le contrôle de la calle piézo-électrique et le contrôle de l'éclairage.

De plus, le déclenchement des images sur le DMD et la prise de photos avec la Nucléo n'a pas été réalisée mais a été documentée. D'autres améliorations pourront également être apportée en raison de l'utilisation souhaitée des autres.

### 3.1.1 Déclenchement du DMD avec une source externe

Nous pouvons utiliser le connecteur d'entrée J23 sur le DMD pour contrôler le déclenchement des images. Pour chaque entrée, il y a trois pins utilisées récapitulées dans 3.

**Table 4-1. Trigger Input Connector Pins**

Description	Pin	Supply Range
Trigger In 1 Supply	1	External or internal 1.8 V and 3.3 V.
Trigger In 1	2	
Ground	3	Ground
Trigger In 2 Supply	4	External or internal 1.8 V and 3.3 V.
Trigger In 2	5	
Ground	6	Ground

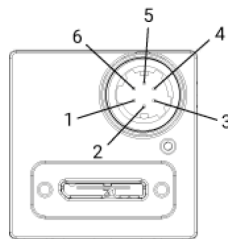
FIGURE 3

Il y a deux entrées possibles  $TRIG_{IN-1}$  qui ordonne de passer à l'image suivante et  $TRIG_{IN-2}$  qui mets en pause et redémarre le pattern. Ces deux commandes peuvent être configurées en utilisant le guide programmeur du DMD.

De plus, pour pouvoir utiliser le DMD dans cette configuration, il faudra le renseigner dans le code lorsqu'on lance le motif sur le DMD. Un des problèmes pour la réalisation technique était que les ports utilisés actuellement sont difficiles d'accès à cause du boîtier du DMD. On aurait du démonter le boîtier et faire nos tentatives ce qui était impossible dans le laps de temps restant.

### 3.1.2 Déclenchement de la caméra avec une source externe

Pour pouvoir contrôler la caméra, nous pouvons utiliser un des pins 1, 2 ou 4(4).



Pin	Line	Function
1	Line 3	General purpose I/O (GPIO) line
2	Line 1	Opto-coupled I/O input line
3	Line 4	General purpose I/O (GPIO) line
4	Line 2	Opto-coupled I/O output line
5	-	Ground for opto-coupled I/O lines
6	-	Ground for General Purpose I/O (GPIO) lines

FIGURE 4

Ensuite, il faut configurer sur python l'utilisation du pin que l'on a choisi.

1. camera.TriggerSelector.SetValue("Frame Start")
2. camera.TriggerSource.SetValue("Line1")
3. camera.TriggerMode.SetValue("On")

Ici, avec la fonction *Frame Start*, la photo va être prise à un moment avec un temps d'exposition prédéfini. La deuxième ligne permet de désigner le pin alloué à notre fonction et la dernière définit le statut *ON* ou *OFF*.

## **3.2 Retour d'expérience**

Durant ce projet, nous avons pu apprendre de nouvelles compétences notamment dans la compréhension des liaisons USB mais également sur l'interfacage graphique. De plus, il y avait une bonne ambiance de travail tout au long du projet ce qui nous a permis d'avancer et de progresser même si nous sommes déçus de ne pas avoir pu ajouter toutes les implémentations que l'on aurait souhaité.

## Références

- [1] BASLER. *Git-Hub*. URL : [https://github.com/basler/pypylon/blob/master/samples/save\\_image.py](https://github.com/basler/pypylon/blob/master/samples/save_image.py).
- [2] *Laboratoire d'enseignement Expérimental*. URL : <http://lense.institutoptique.fr/>.
- [3] MARQUET-COUSTHAM-COHEN. *PIMS 02 – Biophotonique MIS*. URL : <http://lense.institutoptique.fr/realpims1920mai/>.
- [4] Paolo POZZI. URL : <https://github.com/csi-dcsc/Pycrafter6500>.
- [5] TI. *Kit de développement*. URL : <https://www.ti.com/tool/DLP-ALC-LIGHTCRAFTER-SDK#downloads>.