

Mur d'images

Guide d'utilisation et de maintenance

EL BAHRI Ilyes CAMON Baptiste
BOUCHICHIT Walid DUBARLE Louis

29 mai 2023

Résumé

Ce document sert de guide d'utilisation et de maintenance pour notre projet *Mur d'images*. Il y est décrit les technologies utilisées, le fonctionnement du système ainsi que les connexions entre les différents modules constituant.

Table des matières

1	Description générale du projet	3
1.1	Jetson XAVIER	3
1.2	Matrice d'écrans	3
1.3	Écrans secondaires	3
1.4	Microcontrôleur Arduino	3
1.5	Routeur et switch	3
1.6	Prises commandées	3
2	Scripts	3
2.1	Jetson XAVIER	3
2.1.1	Site web	3
2.1.2	Récupération des données (météo)	4
2.1.3	Récupération des données (transports)	4
2.1.4	Conversion des vidéos	4
2.1.5	Conversion des images	4
2.1.6	Envoi des fichiers de configuration	4
2.1.7	Envoi des sources	4
2.1.8	Envoi des données	4
2.1.9	Gestion de l'alimentation	5
2.1.10	Routine	5
2.2	Arduino	5
2.3	Jetson NANO	5
2.3.1	Godot	5
3	LOGS et adresses IP	6
3.1	Jetson XAVIER	6
3.2	Jetson NANO	6
3.2.1	Jetson slave 1	6
3.2.2	Jetson slave 2	6
3.3	RaspberryPi 2	6
3.4	Routeur	6

1 Description générale du projet

1.1 Jetson XAVIER

Dans ce guide nous utiliserons l'expression *maître* ou *PC maître* pour faire référence à la Jetson XAVIER. Cette Jetson XAVIER sert à administrer tout le système et à envoyer les ordres aux différentes composantes de notre projet. Elle sert aussi à récupérer les informations des transports en commun et de la météo. Enfin, elle est utilisée comme *host* pour un site web permettant de déposer des vidéos et images affichables sur les différents écrans. Le site permet aussi de créer des *timelines* d'affichage des ressources sur les différents écrans.

1.2 Matrice d'écrans

Chaque écran est contrôlé par une RaspberryPi 2 à l'aide du logiciel PiWall dont la documentation est disponible sur le site piwall.co.uk

Chacune des RaspberryPi sont numérotées et associées à un écran de 1 à 9, de gauche à droite et de haut en bas. Les RaspberryPi 2 sont mises en attente jusqu'à recevoir une vidéo du PC maître.

Cette matrice sert à afficher des vidéos déposées par les utilisateurs, des images et le flux vidéo en direct de la serre géodésique.

1.3 Écrans secondaires

Contrôlés par des Jetson NANO, ils permettent d'afficher les informations liées à la météo et aux transports en commun envoyées par le PC maître. Ils servent aussi à afficher les vidéos et images déposées sur le site web.

1.4 Microcontrôleur Arduino

Cette carte Arduino permet de gérer l'alimentation du système grâce à un module de transmission 433 MHz et une *Real Time Clock* abrégée en *RTC*

1.5 Routeur et switch

Permet de mettre en réseau tous nos modules.

1.6 Prises commandées

Permettent d'éteindre et d'allumer le système de manière autonome.

2 Scripts

2.1 Jetson XAVIER

La plupart des scripts que nous avons développé sont exécutés sur la Jetson XAVIER et sont décrits ci-dessous.

2.1.1 Site web

Comme décrit plus haut le site web permet de déposer des vidéos et images à afficher sur les différents écrans du projet. Il sert aussi à créer une *timeline* d'affichage pour les écrans secondaires. Cette *timeline* générera un fichier de configuration et sera de la forme :

E 0/1 V *filename* I *filename* t B t J t M t

- E pour choisir l'écran concerné par cette timeline.
- V pour jouer une vidéo avec son nom de fichier.
- I pour afficher une image pendant t minutes avec son nom de fichier.
- B pour afficher les horaires de bus pendant t minutes.
- J pour afficher les métros au départ de Jussieu pendant t minutes.

— M pour afficher les informations relatives à la météo pendant t minutes.

Une seconde *timeline* peut être créée pour gérer l’affichage de la matrice d’écrans avec un fichier de configuration de la forme :

$$M\ S\ t\ V\ filename\ I\ filename\ t$$

avec les même spécifications que précédemment et S pour afficher le flux vidéo de la serre pendant t minutes.

2.1.2 Récupération des données (météo)

Pour récupérer les données météo nous avons utilisé l’API météo-france avec la librairie *meteo-france-api*. Le code est disponible sur la Jetson XAVIER (*appli/src/meteo.py*).

2.1.3 Récupération des données (transports)

Une API de test a été utilisée lors du développement de notre projet. Il serait préférable d’utiliser l’API officielle d’Île-de-France Mobilités : [lien](#).

Le code utilisé actuellement est disponible sur la Jetson XAVIER (*appli/src/ratp.py*).

2.1.4 Conversion des vidéos

Écris en python et avec *ffmpeg* ce script unifie le format des vidéos (.mp4) déposées pour qu’elles aient une résolution de 1920x1080 pixels avec un *bitrate* de 1024 kbits/s et 25 fps.

Il se trouve dans le dossier *appli/src2/piwall/*, il convertit les fichiers vidéos se trouvant dans le dossier *appli/src2/piwall/video/raw/* vers le dossier *appli/src2/piwall/video/clean/*. C’est à partir de ce dossier que les vidéos vont être jouées sur la matrice d’écrans.

Un deuxième script est aussi dans le dossier *appli/src2/nano* qui permet de convertir les vidéos à jouer sur les Jetson NANO. Il fonctionne sur le même principe mais baisse la qualité des vidéos pour qu’elles soient fluides.

2.1.5 Conversion des images

Écris en python et avec *ffmpeg* ce script convertit les images (.png) à afficher sur la matrice d’écrans en vidéos (.mp4), cela nous permet de faciliter l’affichage avec PiWall.

2.1.6 Envoi des fichiers de configuration

Un script python utilisant la commande *scp* permet de copier le fichier de configuration précédemment décrit dans les dossiers *sources* des Jetson NANO.

2.1.7 Envoi des sources

Comme pour l’envoi des fichiers de configuration les fichiers sources seront copiés dans les dossiers *sources* des Jetson NANO grâce à la commande *scp*.

2.1.8 Envoi des données

Pour les données liées à la météo et aux transports en commun, un fichier config sera créé dans les scripts de requêtes API respectifs (*ratp.py* et *meteo.py*) :

— "data_metro.txt" dans lequel il y aura la trame correspondant aux prochains passages des metros à la station de Jussieu. Cette trame sera construite de cette façon :

$$R\ T1\ T2\ T3\$$

où chacun des temps T correspondra au temps d’attente des metros dans un ordre défini au préalable qui est celui de l’affichage réalisé de gauche à droite et de haut en bas. Cet affichage pour les metros (mais aussi pour les bus) est situé dans le dossier *video/src/metro.jpeg* (*video/src/bus.jpeg* pour les bus).

- "data_bus.txt" dans lequel il y aura la trame correspondant aux prochains passages des bus autour de Jussieu. Cette trame et le fonctionnement suivra exactement le même raisonnement que pour les métros.
- "data_meteo.txt" dans lequel il y aura la trame correspondant aux températures minimale et maximale ainsi qu'une description du temps du jour et des deux prochains jours. Cette trame sera construite de cette façon :

M-T_min-T_max-Description1-T_min-T_max-Description2-T_min-T_max-Description3-

où T_min correspondra à la température minimale et T_max la température maximale suivi d'une description du type "ensoleillé".

Ces fichiers config seront ensuite copiés dans le dossier source */video/src* des Jetson NANO grâce à la commande *scp*.

2.1.9 Gestion de l'alimentation

Un script python permet d'éteindre les Jetson NANO et les RaspberryPi à une heure précise via une connexion *ssh* en lançant la commande *sudo shutdown -h now*.

2.1.10 Routine

Un script de routine doit être mis en place pour lancer les différents scripts aux bons moments. Le schéma à suivre est le suivant :

1. Allumage du système complet à 8h.
2. Conversion des vidéos.
3. Conversion des images.
4. Copie des fichiers vers les Jetson NANO.
5. Récupération des informations météo et envoi vers les Jetson NANO.
6. Lancement d'une boucle permettant de récupérer les informations concernant les transports et de les envoyer vers les Jetson NANO.
7. Lancement d'une boucle permettant d'afficher les vidéos sur la matrice d'écrans
8. Extinction du système à 20h

2.2 Arduino

Le seul script présent sur la carte arduino est celui qui permet d'allumer et d'éteindre les prises commandées à des heures précises (8h15 pour l'allumage et 18H15 pour l'extinction). Ce script n'a pas encore été implémenté.

2.3 Jetson NANO

2.3.1 Godot

Le script Godot utilisé pour afficher des trames d'images et de vidéos est *Node2D.gd* situé dans le dossier du projet *appli/godot/video/*. Le script est rattaché au noeud Node2D et contient des fonctionnalités pour afficher différents types de contenu, tels que les horaires de métro, les horaires de bus et les prévisions météorologiques. Ce noeud principal contient les différents éléments dans la scène 2D tels qu'un *VideoPlayer*, un *Sprite* et des *Labels* pour le bon affichage des vidéos, des images et des données. Dans le dossier */src* se trouveront toutes les sources du projets copiées grâce au *scp*, soit les fichiers config et les vidéos/images.

Les points importants du code (sinon veuillez simplement vous référer directement au fichier *Node2D.gd*) sont :

- Des variables de contrôle pour afficher une image ou une vidéo et selon s'il s'agit de l'écran 0 ou 1.
- Des tokens qui contiendront les différentes données extraites des fichiers config.
- Une fonction principale appelée à chaque frame du moteur graphique. Elle gère la logique principale du script, y compris la gestion des transitions entre les différents types de contenu à afficher selon l'écran.

- Les fonctions *get_metro_time()*, *get_bus_time()*, *get_meteo()* et *get_trame()* qui vont ouvrir leur fichier config correspondant, lire les données contenues et les stocke dans leur token respectif.
- Les fonctions d’affichagees *display_metro()*, *display_bus()*, *display_meteo()* et *display_trame()* qui vont faire apparaître et disparaître les images avec un fondu ainsi que les vidéos à partir de leur token respectif contenant les différentes données nécessaires.

Le script se lance automatiquement à l’allumage des Jetson NANO.

3 LOGS et adresses IP

3.1 Jetson XAVIER

- Login : nvidia
- Password : nvidia
- Adresse IP : 192.168.1.12

3.2 Jetson NANO

3.2.1 Jetson slave 1

- Login : slave1
- Password : slave123
- Adresse IP : 192.168.1.201

3.2.2 Jetson slave 2

- Login : slave2
- Password : slave123
- Adresse IP : 192.168.1.202

3.3 RaspberryPi 2

- Login : pi
- Password : slave123
- Adresse IP : 192.168.1.10X (X = numéro de la RaspberryPi de 1 à 9)

3.4 Routeur

- Login : admin
- Password : Router123456789
- Question 1 : AX1800
- Question 2 : Mahares
- Adresse IP : 192.168.1.1