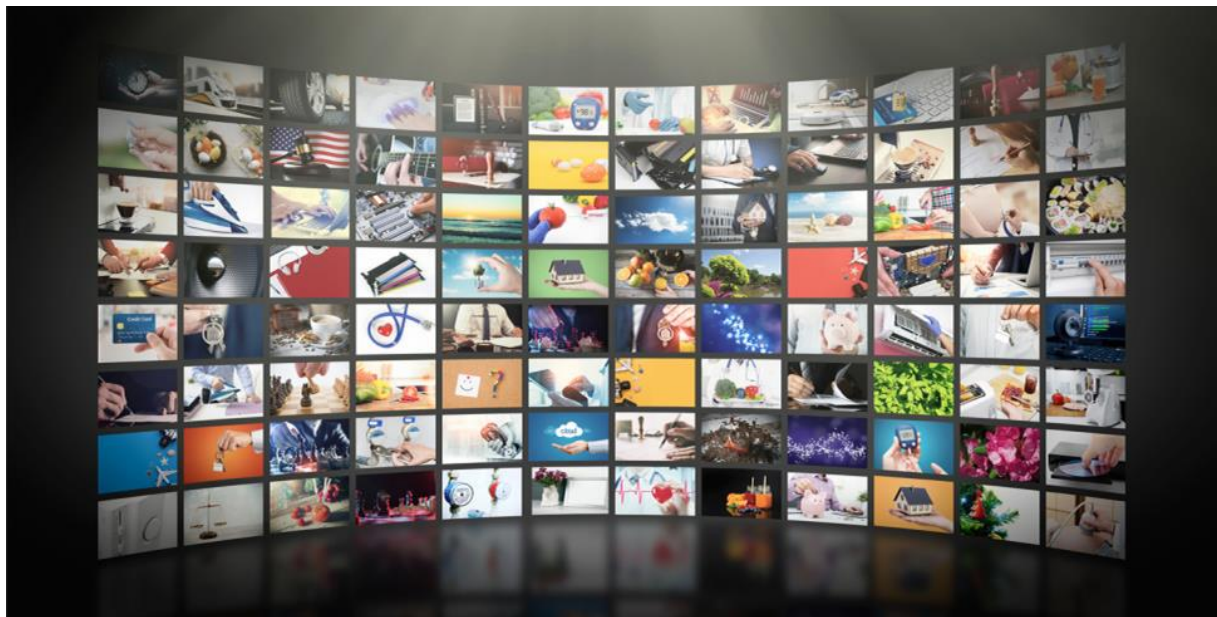


Projet Station TV

RAPPORT PROJET SMART SYSTEM



Anthony LONDAIS
Clément HAUDEBOURG
Promotion 2019-2022
DII4A

Projet Smart system – Station TV
Encadrant – M. Mathieu DELALANDRE

Anthony LONDAIS
Clément HAUDEBOURG
POLYTECH TOURS – 2020/2021

Sommaire

1	Remerciements	4
2	Présentation et contexte	4
2.1	Contexte du projet.....	4
2.2	Problématique	4
2.3	Reprise de projet	5
3	Cahier des charges	5
3.1	Définition des besoins	5
3.2	Contraintes liées au projet	5
4	Organisation	6
5	Réalisations.....	8
5.1	Assemblage de l'armoire.....	8
5.2	Programmation des phidgets.....	12
5.2.1	Choix du langage.....	12
5.2.2	Configuration du phidget.....	12
5.2.3	Protocole NEC.....	12
5.2.4	Code à envoyer.....	13
5.2.5	Utilisation du programme.....	14
5.3	Smart Zapping.....	14
5.3.1	Travail à réaliser.....	14
5.3.2	Utilisation du programme.....	15
5.4	Smart mozaïcisme.....	15

5.4.1	Récupération d'un fichier XML.....	16
5.4.2	Tri par catégorie des programmes	16
5.4.3	Utilisation du programme	18
5.4.4	Génération de fichier CSV	18
5.4.5	Lecture de fichier CSV	18
6	Tests.....	19
6.1	Pilotage individuel.....	19
6.2	Chenillard.....	19
6.3	Smart zapping	19
6.4	Smart mozaicing	20
6.4.1	Validation des fichiers CSV	20
6.4.2	Lecture des fichiers CSV	20
7	Budget	21
8	Bilan.....	22
9	Conclusion.....	23

1 Remerciements

Nous voulons remercier Mathieu DELALANDRE pour nous avoir encadré durant cette année de projet et de nous avoir permis d'en apprendre plus sur le projet station TV. Nous tenons aussi à remercier Polytech Tours pour nous avoir fourni les différentes commandes pour accomplir notre projet. Enfin nous voulons remercier Pierre PALLIER pour les échanges concernant le projet smart zapping et Van-Hao LE pour avoir stoppé la capture le temps que nous puissions notre projet sur la station TV.

2 Présentation et contexte

Dans cette première sous-partie nous parlerons tout d'abord dans quel contexte s'inscrit ce projet, dans un deuxième temps à quelle problématique il tente de répondre et dans une troisième partie des éléments déjà existants pour ce projet.

2.1 Contexte du projet

Dans le cadre de notre deuxième année d'études à Polytech Tours en Informatique Industrielle par apprentissage, nous avons été assigné au projet Station TV encadré par Mathieu Delalandre. Ce projet fait partie d'un groupe de plusieurs projets dont l'idée principale est de développer un zapping télévisuel.

Ce rapport retrace notre parcours sur ce projet : les solutions proposées, les problèmes rencontrés ainsi que la solution finale.

2.2 Problématique

La consommation intensive de contenus multimédias, notamment au travers de la télévision, ainsi que les nouvelles technologies ont fait émerger de nouvelles habitudes. Les utilisateurs sont devenus « dual-screener2 » : ils utilisent leurs terminaux mobiles (Smartphone, tablette,...) en complément des programmes TV proposés par les chaînes de télévisions ou de la vidéo à la demande pour chercher des informations, consulter des réseaux sociaux en relation avec la télévision et les programmes, etc. Mathieu DELALANDRE et Jordan NICOT, ingénieur informatique, ont eu un projet de start-up pour répondre aux problématiques liées à ce contexte. Le but principal de cette start-up est de créer une plateforme qui proposera un guide TV/VoD nouvelle génération. Ce guide sera ergonomique, adapté aux besoins des utilisateurs. Dans cette optique, des modules permettant d'apporter des fonctionnalités à l'utilisateur ont été pensés. Dans le cadre de ce projet, nous considérerons la capture des génériques TV grâce à une planification puis la recherche en temps réel de ces génériques TV sur les chaînes.

2.3 Reprise de projet

Ce projet a débuté l'année dernière mais en raison de la crise sanitaire, il n'a pas pu aboutir. Nous sommes donc partis de ce qui a été fait l'année dernière, c'est-à-dire : le choix des tuners et des émetteurs infrarouges.



Figure 1 : Equipements existant

3 Cahier des charges

3.1 Définition des besoins

Le premier objectif de ce projet est de piloter des tuners avec des émetteurs infrarouges permettant l'acquisition des images télévisuelles.

Le deuxième objectif de ce projet est un travail collaboratif avec un étudiant PRD, notre objectif sera de lire des fichiers fournis par ce dernier.

La troisième et dernière partie est la création d'un programme smart mozaicisme, ce dernier devra permettre de changer l'affichage des chaînes afin d'afficher des chaînes en rapport avec une catégorie.

3.2 Contraintes liées au projet

Pour ce projet plusieurs contraintes :

L'accès à la station TV : en effet, plusieurs groupes travaillent sur cette station, nous avons donc des créneaux réservés pour y accéder. De plus avec la pandémie du COVID 19 l'accès à Polytech et donc à la station a été réduit durant ce projet.

Deuxième contrainte identifiée : le pilotage individuel des tuners par leurs phidgets. En effet l'ensemble des objectifs du projet dépend de ce point. Si un phidget pilote plusieurs tuners cela va mettre en péril le reste du projet.

Troisième contrainte, les commandes. Avec la pandémie, il est à prévoir que les commandes mettent plus de temps à arriver c'est un autre aspect à prendre en compte.

Quatrième contrainte, il nous faut avoir une bonne communication avec Pierre (étudiant PRD) afin de réaliser correctement le programme qu'il souhaite pour piloter les phidgets.

4 Organisation

Pour notre organisation, nous avons séparé les tâches en deux grandes parties :

- L'assemble du meuble : Toutes les tâches autour de la réalisation du meuble ont été réalisées par Anthony.
- Programmation : Toutes les tâches autour de la programmation ont été réalisées par Clément

Cette répartition nous a permis de travailler en parallèle et de faire avancer le plus efficacement notre projet.

NOM DU PROJET	INTERVENANTS	DATE DE DÉBUT	DATE DE FIN	PROGRESSION GÉNÉRALE
Projet station TV	Anthony LONDAIS Clément HAUDEBOURG	09/09/2020	12/05/2021	100%

TÂCHES	RESPONSABLE(S)	DÉBUT	FIN	JOURS	STATUT
Lancement projet smart system	Anthony - Clément	9-sept.	9-sept.	1	Terminé
Réunion : explication du projet	Anthony - Clément	18-sept.	18-sept.	1	Terminé
Lecture des rapports précédents	Anthony - Clément	18-sept.	9-oct.	22	Terminé
Reflexion sur l'installation des équipements	Anthony	9-oct.	9-oct.	1	Terminé
Programmation des phidgets	Clément	10-oct.	9-nov.	31	Terminé
Recherche d'équipements + commande	Anthony	10-oct.	9-nov.	31	Terminé
Installation du rack x24 par satellite37	Anthony - Clément	21-nov.	21-nov.	1	Terminé
Schéma d'installation des équipements dans le rack	Anthony	9-nov.	2-déc.	24	Terminé
Soutenance mi-parcours	Anthony - Clément	2-déc.	2-déc.	1	Terminé
Installation des équipements dans le rack	Anthony	20-déc.	15-mars	86	Terminé
Recherche d'équipements manquant + commande	Anthony	20-déc.	15-mars	86	Terminé
Codage d'un chenillard sur les chaines TV	Clément	20-déc.	15-mars	86	Terminé
Codage d'un programme lisant des fichiers CSV	Clément	15-mars	7-avr.	24	Terminé
Codage du programme smart mozaïcisme	Clément	15-mars	7-avr.	24	Terminé
Premier tests	Anthony - Clément	7-avr.	7-avr.	1	Terminé
Installation du rack x8 dans la station TV	Anthony - Clément	7-avr.	7-avr.	1	Terminé
Reflexion d'une nouvelle solution évitant le multi-pilotage	Anthony	8-avr.	15-avr.	8	Terminé
Réalisation de la nouvelle solution	Anthony	15-avr.	5-mai	8	Terminé
Tests finaux	Anthony - Clément	5-mai	5-mai	1	Terminé
Rédaction du rapport + préparation soutenance	Anthony - Clément	5-mai	11-mai	7	Terminé
Soutenance finale	Anthony - Clément	12-mai	12-mai	1	Terminé
LANCEMENT		10-oct.	10-oct.	1	

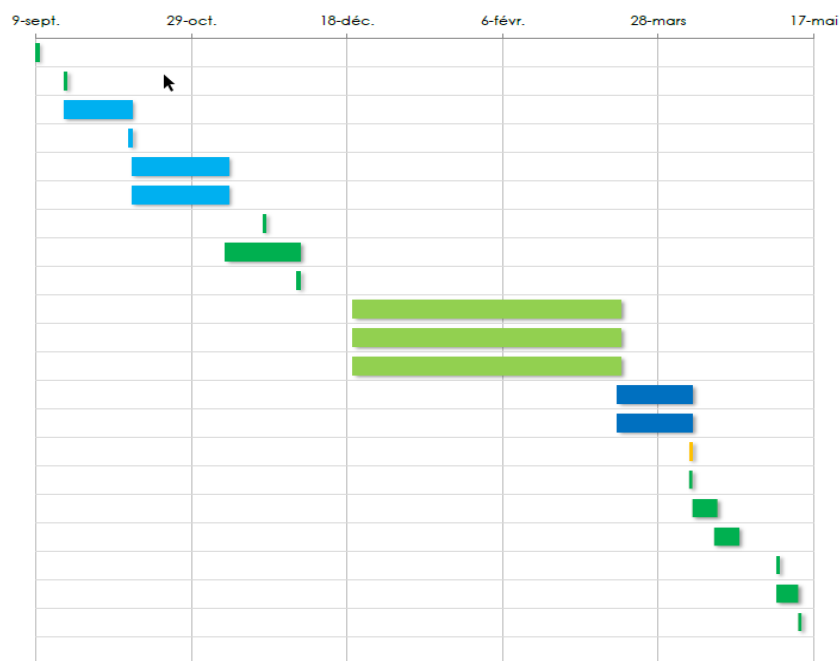


Figure 2 : Diagramme de Gantt

5 Réalisations

5.1 Assemblage de l'armoire

Ayant repris un projet de l'année dernière, nous avons pu gagner du temps en reprenant les équipements déjà choisis et validés et nous concentrer sur le choix des autres équipements..

Notre mission pour le montage de l'armoire était alors de :

- 1- Réfléchir aux matériels nécessaires pour permettre le pilotage des tuners et trouver un moyen d'installer tous ces équipements dans l'armoire,
- 2- En accord avec le tuteur de projet, Trouver un moyen simple de séparer les phidgets pour empêcher que les phidgets pilotent plusieurs tuners Avermedia que celui qui lui était attribué

Après réflexions et discussions avec l'encadrant , les matériels que nous avons commandés sont:

- Les Tuners Astrell manquants
- Les phidgets manquants
- 2 hub usb 4 ports avec alimentation pour piloter plus facilement les phidgets.
- 4 ventilateurs pour éviter la surchauffe des équipements qui peuvent rester allumé longtemps dans une armoire
- 4 Transformateurs pour pouvoir brancher les ventilateurs aux multiprises
- 4 multiprises 4 ports
- 1 multiprise 4 ports avec interrupteur
- 2 rallonge usb de 3 mètres
- 8 cables mini USB
- 8 cable HDMI de 3mètres
- 8 cables coaxiales male/femelles
- 1 câble satellite fiche F 4m
- Adaptateur Fiche F male coaxiale*4

Ci-dessous un schéma de l'installation des équipements dans l'armoire:

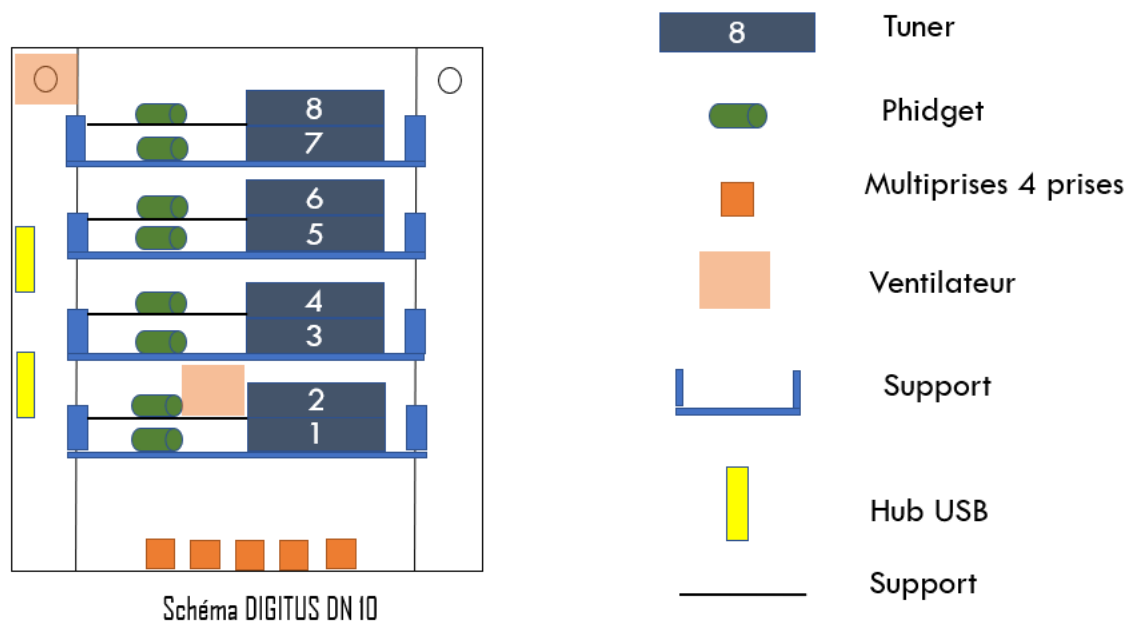


Figure 3 : Schéma Armoire Mi-Parcours

Ensuite nous avons reproduit ce schéma avec les équipements et l'armoire. Nous avons rencontré quelques difficultés, notamment sur l'installation des câbles antenne et la longueur des câbles tels que les câbles antennes car les magasins comme Leroy Merlin ne vendaient que des câbles d'1m50 et les câbles d'alimentation des tuners qui sont aussi long.

En ce qui concerne la méthode pour séparer les phidgets et tuners, nous avons d'abord eu l'idée de mettre une planche de carton entre chaque couple de tuners et d'attacher à cette planche, 2 rouleaux de carton pour y mettre les phidgets.



Figure 4 : Armoire Mi-Parcours

Après le montage de l'armoire, nous avons contacté notre tuteur afin d'organiser un rendez-vous pour tester le chenillard, la lecture de fichier ".csv" et le smart zapping.

Lors des tests avec la station TV, nous avons pu constater que notre solution pour séparer les phidgets ne fonctionnait pas. En effet, certains phidgets pilotaient plusieurs tuners en même temps. Nous avons supposé que les faisceaux lumineux de l'émetteur infrarouge se propageaient aussi par l'arrière et par effet de réflexion, envoyaient leur commande sur les autres Tuners.

Nous avons donc cherché un autre moyen pour éviter ce multi pilotage. Nous avons décidé de fabriquer nous-mêmes des boîtes pour chaque Phidget ainsi que de mettre une plaque entre chaque tuners.

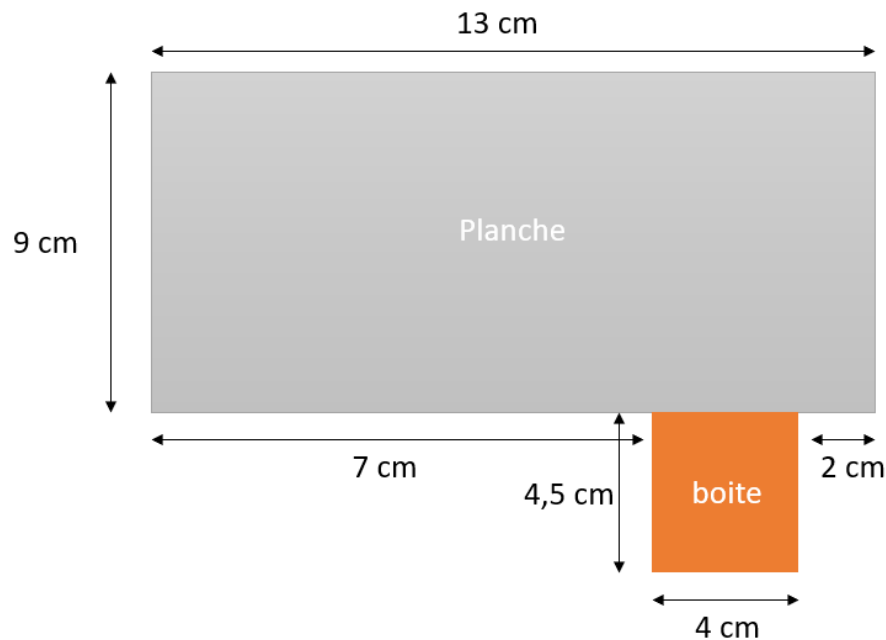


Figure 5 : Mesure Solution Boîte

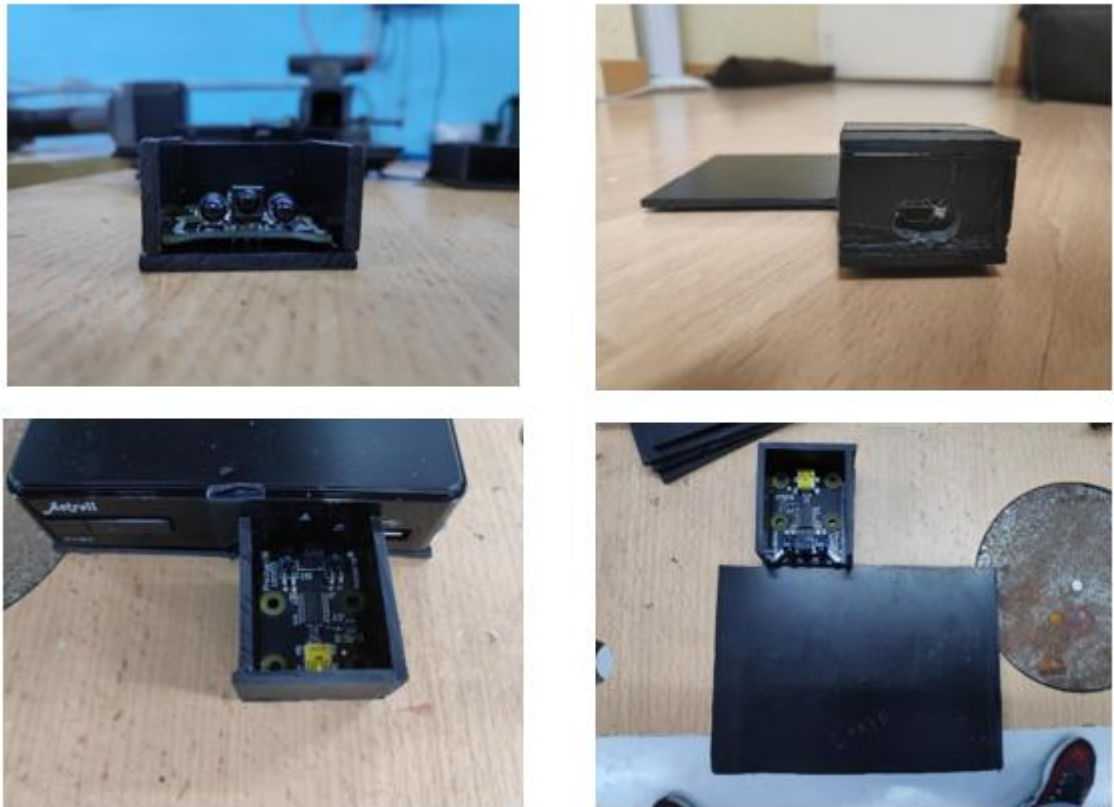


Figure 6 : Solution boîte

Voici l'armoire finale :



Figure 7 : Armoire finale

5.2 Programmation des phidgets

5.2.1 Choix du langage

Tout d'abord il nous a fallu choisir le langage dans lequel nous allons les programmer. Sur le site du fabricant on retrouve cette liste de langage proposée et fourni une API pour chacun de ces langages.

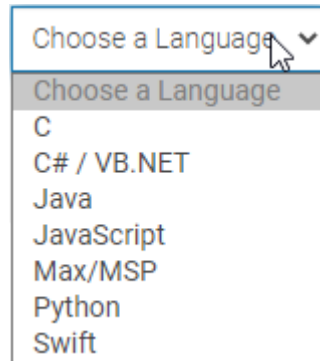


Figure 8 : Language Programming Phidgets

Notre choix a été le C car c'est le langage que nous maîtrisons le plus.

5.2.2 Configuration du phidget

Installation des librairies : [https://www.phidgets.com/docs/OS - Windows](https://www.phidgets.com/docs/OS_-_Windows)

Il faut attribuer à chaque phidget un numéro de série (indiqué sous le phidget).

Configurer le phidget pour qu'il utilise le protocole NEC

5.2.3 Protocole NEC

Le protocole NEC a été développé par l'entreprise japonaise NEC, dans ce protocole chaque fabricant attribue un code unique (code système) à toutes les commandes et un code pour chaque commande.

Caractéristiques du protocole :

- Adresse de 8 bits (mode étendu disponible, doublant la taille de l'adresse)
- Commande de 8 bits
- L'adresse et la commande sont transmises deux fois pour plus de fiabilité.
- Modulation de la distance d'impulsion.
- Fréquence porteuse de 38 kHz.
- Temps de bit de 1,125 ms ou 2,25 ms.

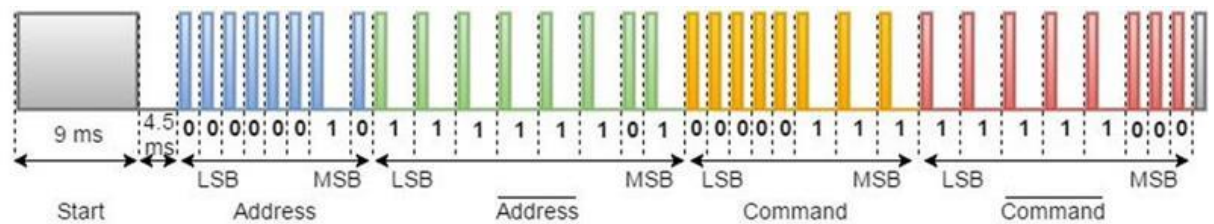


Figure 9 : Trame Protocole NEC

Fonctionnement de l'envoi d'un code :

- Emission d'une impulsion de 9ms
- Pause de 4,5 ms
- Code système LSB
- Code système bar LSB
- Commande LSB
- Commande bar LSB

5.2.4 Code à envoyer

Le code à envoyer va changer en fonction de chaque chaînes.

Exemple touche power :

Code système : 0x08F6

En suivant les indications du protocole NEC :

Code système MSB	0000 1000
Code système LSB	0001 0000
Code système MSB bar	1111 0110
Code système LSB bar	0110 1111

Code système à envoyer : Code système LSB + code système LSB bar :

0001 0000 0110 1111 → 0x106F

Commande power : 0x43

Commande power MSB	0100 0011
Commande power LSB	1100 0010
Commande power MSB bar	1011 1100
Commande power LSB bar	0011 1101

Commande à envoyer : commande power LSB + commande power LSB bar :

1100 0010 0011 1101 → 0xC23D

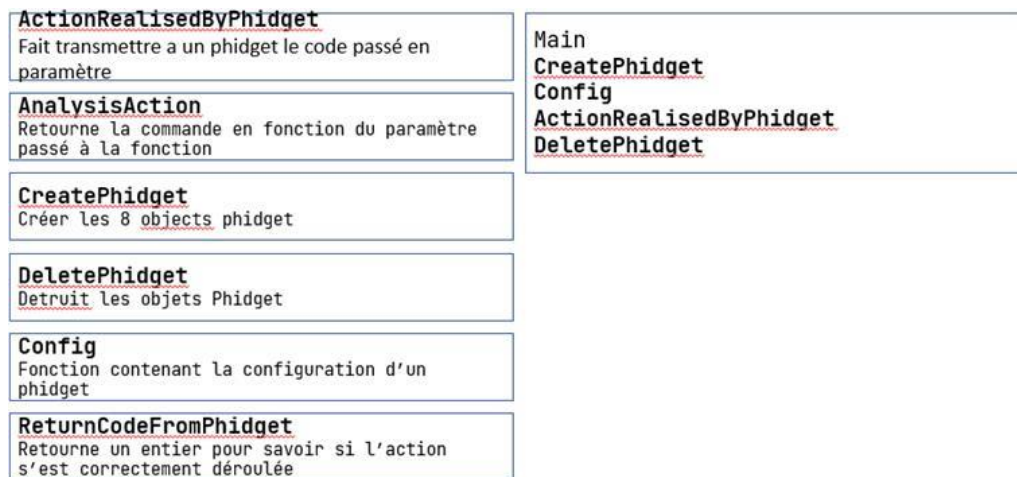


Figure 10 : Algorithme Programme Phidget

5.2.5 Utilisation du programme

L'installation des librairies va installer une application graphique, avant d'utiliser le programme il faut fermer cette application sinon elle va bloquer le programme.

Lancement de l'exécutable : `./nom_executable phidget_a_controler`
numéro_chaine_a_afficher

phidget_a_controler : Nombre compris entre 0 et 7

numéro_chaine_a_afficher : nombre compris entre 000 et 999

5.3 Smart Zapping

5.3.1 Travail à réaliser

Cette partie du projet est en collaboration avec un étudiant de DI 5A effectuant un PRD.

Le travail que nous devons réaliser sur cette partie est la lecture de fichier CSV afin de piloter les phidgets en fonction des données présentes dans ces fichiers CSV.

Dans ces fichiers CSV on retrouve 3 informations :

- Le nom de la chaine
- La date de début d'enregistrement
- La date de fin d'enregistrement

La date de début et de fin d'enregistrement est stocké sous la forme d'un timestamp.

Le timestamp représente la quantité de secondes passées depuis le premier janvier 1970 de l'horloge UNIX.

Le programme fonctionne de la façon suivante :

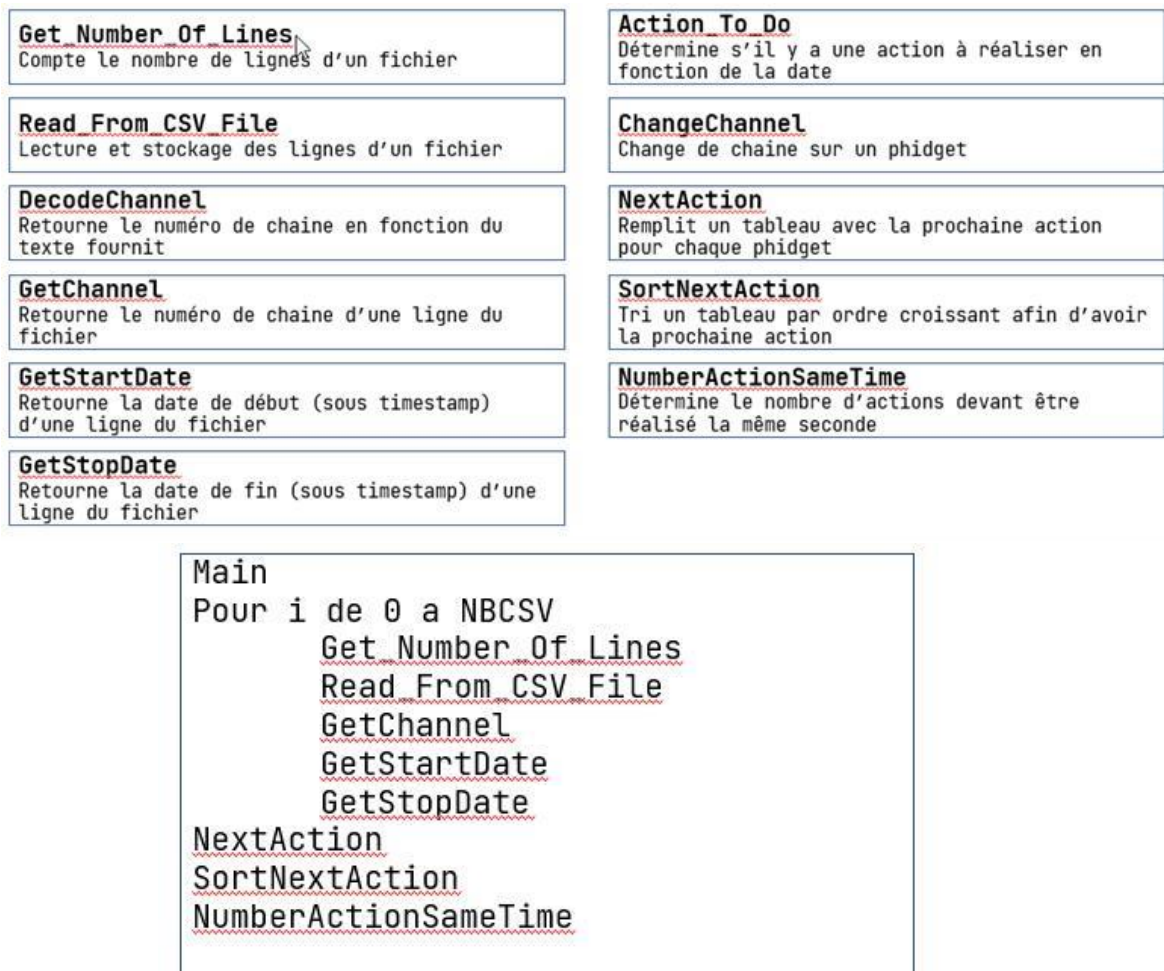


Figure 11 : Algorithme lecture CSV

5.3.2 Utilisation du programme

Lancement de l'exécutable : `./nom_executable`

5.4 Smart mozaïcisme

La partie smart de notre projet a pour objectif un affichage de chaînes en fonction d'une catégorie.

Elle va donc se séparer en plusieurs parties :

- Récupération d'un fichier xml
- Tri par catégorie des programmes
- Génération de fichier CSV
- Lecture de fichier CSV

5.4.1 Récupération d'un fichier XML

La première étape est de récupérer la liste de tous les programmes.

On retrouve ces informations dans des fichiers au format XML.

Qu'est-ce qu'un fichier XML ?

XML est un métalangage de structuration et de représentation de données. Sa syntaxe est reconnaissable par son usage des chevrons <> encadrant les noms des balises.

Nous avons choisi d'utiliser <https://xmltv.ch> afin de récupérer la liste des programmes.

```
<?xml version="1.0" encoding="UTF-8"?>
<tv source-info-url="https://api.telerama.fr" source-data-url="https://api.telerama.fr" generator-info-
name="XMLTV" generator-info-url="http://mythtv-fr.org/">
  <channel id="C4.api.telerama.fr">
    <display-name>France 2</display-name>
    <icon src="https://television.telerama.fr/sites/tr_master/files/sheet_media/tv/500x500/4.png" />
  </channel>
  <channel id="C34.api.telerama.fr">
    <display-name>Canal+</display-name>
    <icon src="https://television.telerama.fr/sites/tr_master/files/sheet_media/tv/500x500/34.png" />
  </channel>
```

Figure 12 : Debut fichier XML

Une fonction du programme sera donc de télécharger ce fichier XML.

5.4.2 Tri par catégorie des programmes

La deuxième étape de cette partie est de trier les programmes en fonction de leur catégorie.

Pour exploiter ce fichier XML, il va nous falloir utiliser un analyseur syntaxique XML (ou parseur) qui va alors nous permettre de récupérer dans la structure XML les **balises**, leurs **contenus** ainsi que leurs **attributs**.

Structure du fichier XML : Le fichier se divise en deux parties, au début on retrouve la liste des chaînes et leurs **ID** associés, dans une seconde partie on retrouve la liste des programmes avec différentes informations.

Les balises qui nous intéressent sont les suivantes :

Balise programme : récupération de la date de début, de la date de fin et de la chaîne.

Balise Category : récupération de la catégorie du programme.


```

<programme start="20210326001000 +0100" stop="20210326004000 +0100" channel="C4.api.telerama.fr">
  <title>Nous, les Européens</title>
  <desc lang="fr">Chaque numéro est thématiqué autour d'un pays européen avec un reportage de quinze
minutes sur une initiative inspirante, un invité en plateau pour savoir si le modèle présenté est applicable à la
France et une chronique sur les actualités.</desc>
  <category lang="fr">magazine de société</category>
  <length units="minutes">30</length>
  <icon
src="https://television.telerama.fr/sites/tr_master/files/sheet_media/media/5d00c73ed700d6dd37855fe0888
4750007a11081.jpg" />
  <rating system="CSA">
    <value>Tout public</value>
  </rating>
</programme>
<programme start="20210326004000 +0100" stop="20210326012500 +0100" channel="C4.api.telerama.fr">
  <title>13h15, le dimanche...</title>
  <sub-title>Le village des enfants</sub-title>
  <desc lang="fr">Immersion au cœur d'un village d'une dizaine de maisons, à Amboise, en Indre-et-Loire dans
lesquelles vivent des enfants placés par un juge. Ici, contrairement à ce qu'il se passe dans les foyers ou les
familles d'accueil, les fratries ne sont pas séparées.</desc>
  <category lang="fr">magazine d'information</category>
  <length units="minutes">45</length>
  <icon
src="https://television.telerama.fr/sites/tr_master/files/sheet_media/media/4177700539bd3eda2d87680d41c
78b3223d73c51.jpg" />
  <previously-shown />
  <rating system="CSA">
    <value>Tout public</value>
  </rating>
</programme>

```

Figure 13 : Programme XML



Figure 14 : Algorithme Smart mozaicing

Actuellement le programme peut générer des fichiers CSV en fonction des catégories suivantes :

- sport
- musique
- politique
- divertissement
- documentaire
- film
- série
- jeu
- dessin animé
- journal
- magazine
- météo
- talk-show

5.4.3 Utilisation du programme

Lorsque l'on souhaite changer le type de catégorie des programmes à afficher, il faut appeler le programme avec comme argument la catégorie, ce dernier va alors régénérer des fichiers en fonction de la nouvelle catégorie choisies.

5.4.4 Génération de fichier CSV

L'avant dernière étape de cette partie est la génération de fichier CSV. Nous avons décidé de générer des fichier CSV car nous avons réalisé auparavant un programme permettant de lire ces fichiers CSV.

Tout d'abord le programme va créer/écraser 8 fichiers CSV.

Ensuite lorsqu'un nouveau programme associé à la catégorie choisi a été trouvé il va l'insérer dans le premier fichier, le prochaine programme sera insérer dans le deuxième fichier et ainsi de suite jusqu'au dernier programme.

Cela permet d'avoir une répartition équivalente sur tous les phidgets et un phidget ne sera pas plus utilisé qu'un autre

5.4.5 Lecture de fichier CSV

Le programme réalisé cette fonction à déjà été développé, cependant on notera une modification par rapport à la première version créer :

Les fichiers CSV générer indiquent directement le numéro de chaines à afficher ainsi le programme de lecture CSV n'a pas besoin d'effectuer ce tri.

6 Tests

6.1 Pilotage individuel

Afin de vérifier le bon fonctionnement des boîtes, nous avons fait quelques tests. Nous avons envoyé la commande ON/OFF sur le phidget en vérifiant bien que le phidget pilote le tuner en face de lui sans pour autant changer l'état d'un autre tuner

Avant

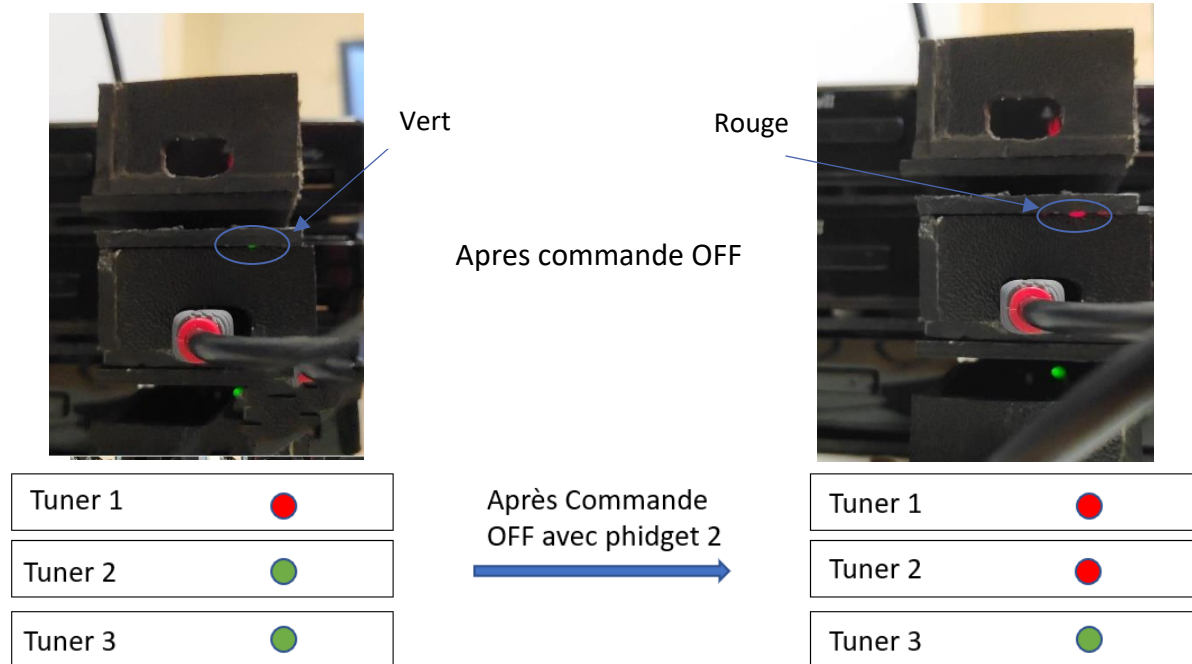


Figure 15 : Validation Solution avec Test ON/OFF

Nous observons bien (plutôt difficilement sur l'image) qu'il y a bien que le tuner au milieu qui change d'état (passe du vert au rouge)

6.2 Chenillard

Le test du chenillard consistait à tester sur la station le bon pilotage individuel des tuners ainsi que l'accès à toutes les chaînes disponibles.

Ce test a été validé.

6.3 Smart zapping

Le test du smart zapping était l'illustration du projet réalisé pour Pierre. Cependant les fichiers générés par Pierre interviennent à des moments précis et nous n'avons pas utilisé sur l'un de ces moments précis. Comme notre objectif était de montrer la bonne lecture des fichiers CSV, nous avons modifié les dates d'exécution de ses fichiers afin de valider ce fonctionnement.

Après modification nous avons validé le bon fonctionnement : Changement de la chaîne à l'heure demandé (simulant le début d'enregistrement) et l'envoi de la chaîne à la fin simulant la fin d'enregistrement).

6.4 Smart mozaicing

Les tests pour cette partie se sont séparés en deux parties :

- La validation de la génération correcte des fichiers CSV
- La lecture et envoi des chaînes correspondante sur la station TV

6.4.1 Validation des fichiers CSV

Pour tester cette partie, il a fallu manuellement vérifier qu'après avoir généré les fichiers pour un type de catégorie, les programmes présents correspondaient.

Pour cela nous avons testé pour chaque catégorie les 3 premiers programmes de chaque fichiers qu'ils étaient en accord avec le programme TV.

Ces tests ont été concluants.

6.4.2 Lecture des fichiers CSV

Ce test a été le même que pour la partie de Pierre, même s'il avait de fortes probabilités de fonctionnement nous avons quand même validé le bon fonctionnement de cette partie.

7 Budget

Equipements	Prix unitaire (€)	Quantité	Prix (TTC)(€)
Phidget 1055	47,42	8	379,34
Tuner Astrell	24,90	8	199,2
Cable HDMI	12,95	8	103,58
Multiprises 4 ports	9,95	4	39,79
Multiprise interrupteur	8,26	1	8,26
Transformateurs	9,95	4	39,79
Adaptateurs Fiche F	5,88	4	23,52
Répartiteur	74,99	1	74,99
Hub USB 4 ports	10,98	4	43,9
Rallonge USB	9,95	2	19,9
Cable mini USB	4,79	8	38,3
Cable satellite Fiche F	7,08	1	7,08
Câble coaxial	4,20	8	33,6
Armoire Digitus DN10	74,62	1	74,62
Ventilateurs	8,91	4	35,62
TOTAL			1121,49

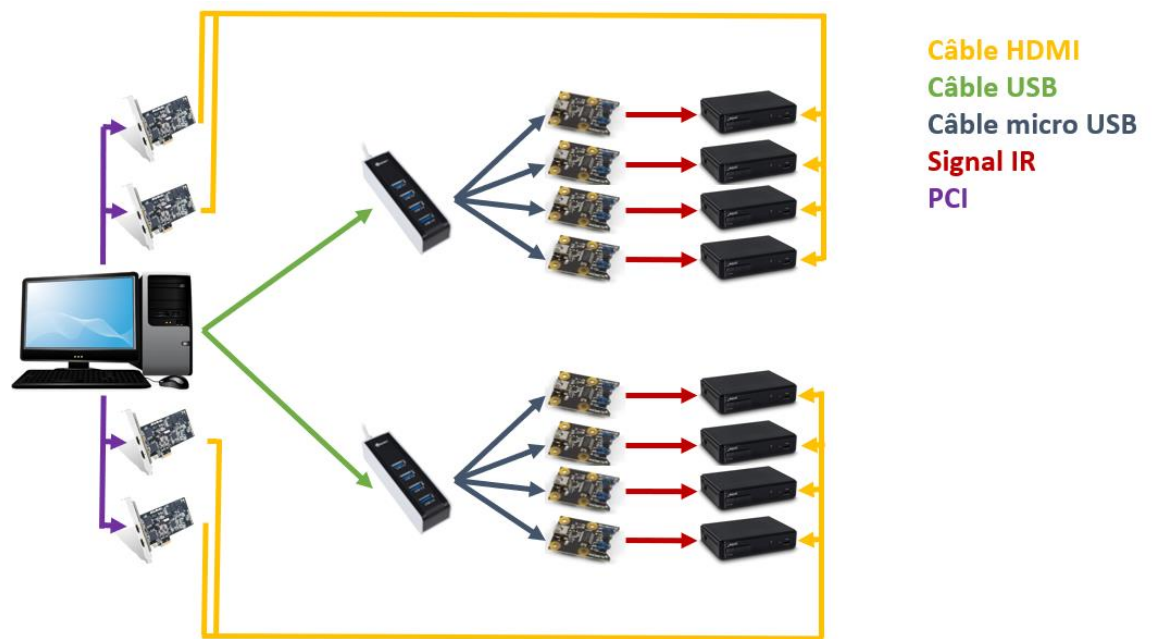
Figure 16 : Tableau Budget

8 Bilan

Sur l'avancement du projet, nous avons rempli le cahier des charges

Pour l'armoire, il est possible de l'améliorer en raccourcissant les câbles afin de faire plus de place dans l'armoire.

Le programme smart triant par catégorie et générant les fichiers CSV peut-être amélioré. Nous avons choisi de réaliser un tri manuel des programmes et le modifier en utilisant des algorithmes NPL serait plus efficace



9 Conclusion

Pour conclure, le projet proposé par Mathieu DELALANDRE fut fort intéressant, d'une part puisque nous avons eu l'opportunité de travailler avec des composants que nous n'avions jamais utilisé auparavant comme les tuners, mais aussi puisqu'il nous a permis de comprendre le fonctionnement de la réception d'images, celles-ci étant nécessaire pour le zapping télévisuel intelligent.