2018

Lycée Alphonse Benoit

ANNICELLI Nathan - BILLY Olivier - HUET Valentin



REVUE DE PROJET

Mise en place d'un système d'affichage d'informations pour Lycées

Sommaire

1. Présentation et situation du projet 3

1.1. Présentation du projet 3

1.2. Cahier des charges – Expression du besoin 3

2. Spécifications 4

3. Diagrammes UML/SYSML 5

3.1. Diagramme des cas d'utilisation 5

3.2. Diagramme de déploiement 6

3.3. Diagramme des exigences générales 6

3.4. Diagramme de séquence système 7

3.5. Contraintes de réalisation 8

3.6. Ressources mises à disposition des étudiants (logiciels/matériels/documents) 8

4. Répartition des fonctions par étudiant 9

5. Compte rendu de réunion 11

6. Partie Physique 12

Le projet : 12

HDMI, Colorimétrie et ratio : 12

L'évolution de l'HDMI : 13

Quelques caractéristiques : 13

7. ANNICELLI Nathan : Technicien IR1 14

7.1. Introduction 14

7.2. Prendre connaissance de l'existant 15

7.3. Serveur local 17

7.4. Création du thème « tableaux » 20

7.5. Ajout de couleurs et d'une nouvelle zone 'Partout' 25

7.6. Problèmes rencontrés 28

8. BILLY Olivier : Technicien IR2 30

8.1. Introduction 30

8.2. Taches à effectuer 30

8.3. Programme Principale AIES2018 30

8.4 Mise en évidence de ma partie 31

8.5 Visuel de l'application AIES2018 31

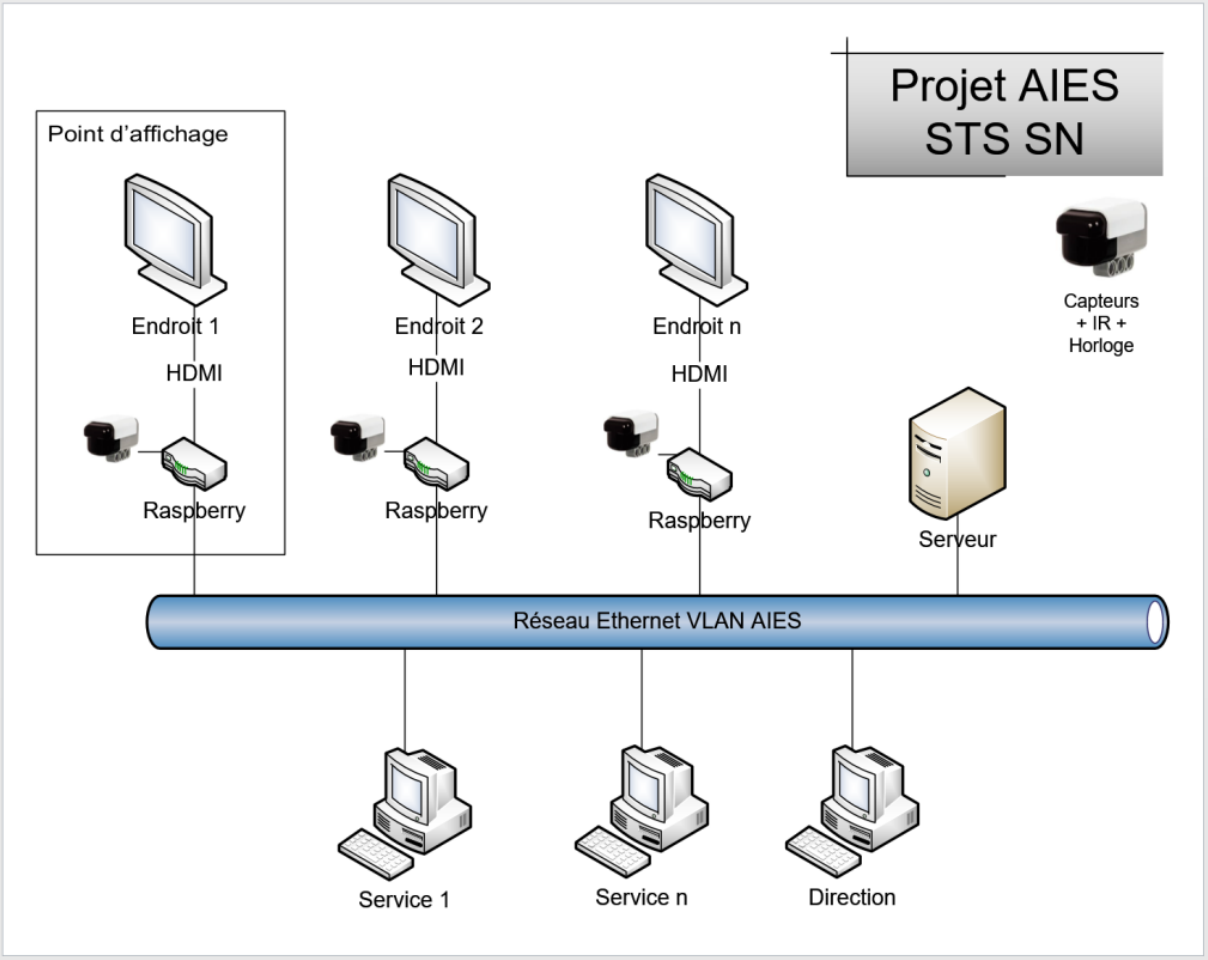
8.6 Programme Principal AIES2018 32

8.7 Diagramme de classe AIES2018 34
8.8 Formulaire AppAIES2018 35
8.9 Diagramme de séquence Mise en Cache 36
8.10 Diagramme de séquence Système Automatique 37
8.11 Bandeau Flash 38
8.12 Problèmes et solutions retenues 39
8.13 Evolution au cour du projet 40
8.14 Conclusion 40
9. HUET Valentin : Technicien EC 41
9.1 Introduction : 41
9.2 Travail effectué : 41
9.3 Réalisation d'un Gantt : 42
9.5 Commande de la carte et son pochoir 52
9.6 Câblage d'une carte AIES2017 55
9.7 Défaut sur l'horloge concernant l'heure 57
9.10 Partie Physique 61

1. Présentation et situation du projet

1.1. Présentation du projet

Le projet consiste en la création d'un système d'affichages dynamiques d'informations contrôlables par l'outil informatique.
Ce système est composé de plusieurs points d'affichages répartis dans les endroits passants.
Chaque service a la possibilité de publier des informations.
Des informations flashs peuvent être possiblement affichées dans la partie basse des écrans d'affichage.



1.2. Cahier des charges – Expression du besoin

La direction du lycée BENOIT est soucieuse de délivrer une information plus dynamique, moderne, attractive et plus pertinente auprès des personnels et des élèves. De ce fait, le lycée désire se doter d'un système d'affichages dynamiques d'informations, installés dans les endroits passants.
Le projet se voudra évolutif, il sera possible dans l'avenir d'ajouter des points d'affichage (en fonction des besoins, du budget et de la couverture réseau de l'établissement).

Le système a été déployé durant l'année 2016-2017, 5 points d'affichages sont fonctionnels.

2. Spécifications

Un point d'affichage (PA) est composé :

- D'un écran de télévision.
- D'une carte RASPBERRY PI 2 B+ reliée par la prise HDMI.
- Un capteur de présence afin de permettre l'extinction automatique des écrans lorsque personne ne passe (temporisé 1h).
- Un capteur de température.
- Carte de réception d'une horloge (synchronisée ou pas).
- Un émetteur infrarouge pour commander les TV non compatibles avec la norme CEC (nouveauté 2017).

Chaque PA est en liaison réseau permanente avec un serveur LINUX (virtualisé) de bases de données afin de récupérer les informations à afficher. Un serveur WEB permet aux différents services de configurer les informations à afficher.
Chaque PA est identifié de manière unique par l'adresse MAC de l'interface réseau de la carte RASPBERRY.

- **Informations et gabarits**

Chaque service dispose de plusieurs gabarits de page d'informations en fonction de leurs besoins.
Lors de la publication d'une information, l'utilisateur d'un service choisi le gabarit correspondant au type d'information à publier, et ne fait que remplir le texte. Il peut également intervenir sur les paramètres de décoration de la page (logo, couleurs, etc…).
Attention au choix de la police de caractères et de sa taille pour disposer d'un contenu lisible de loin.

- **Zones**

Une zone peut comprendre un ou plusieurs PA. L'affichage est identique pour une zone donnée. Chaque zone peut afficher des informations différentes.

- **Services et profils**

Chaque service de l'établissement autorisé à publier des informations dispose d'un profil.
Le profil administrateur permet de contrôler tous les paramètres de fonctionnement du système.
L'administrateur gère les utilisateurs pouvant se connecter, les services, attribue les zones d'affichage pour chaque service.
Chaque service dispose d'un logiciel client installé (ou interface WEB) permettant de placer du contenu à afficher. Les services n'auront pas besoin de compétences techniques autres que l'utilisation de l'application pour publier des informations (modèles).

- **Message flash**

Un message flash est une information spéciale qui s'affiche de manière continue et pendant une durée déterminée dans la partie basse de l'écran (1/6ème de la hauteur). Seul l'administrateur peut publier des messages flashs.

- **Sécurité**

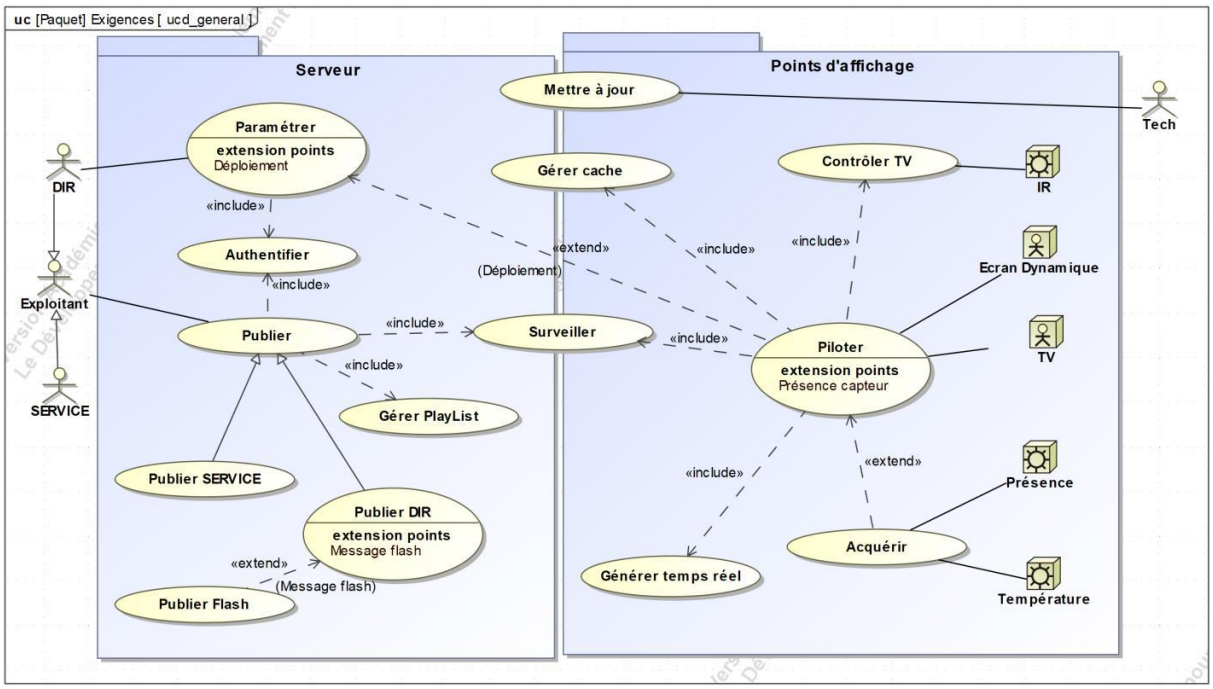
L'administrateur peut à tout moment interrompre l'affichage sur toute les zones en cas d'anomalie. Il peut en outre accéder à toutes les pages d'affichage créées par tous les services.

Pour l'année 2017-2018 :

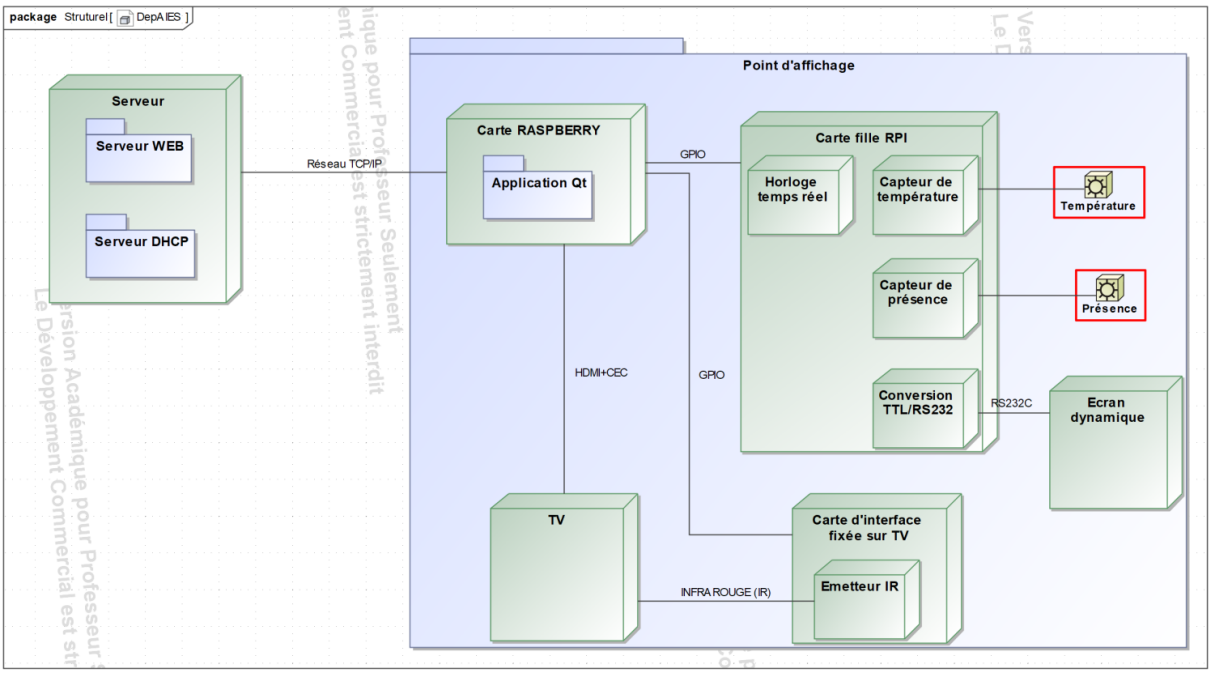
- Installation de 2 PA aux ateliers dans le bâtiment C. Les écrans ne sont plus des TV mais des écrans dynamiques commandés par liaison série RS232C. Un protocole est disponible.
- Déploiement de la version 2017 du logiciel des PA et du site WEB d'administration.
- Création de plusieurs masques de saisie d'information (image, vidéo).
- Surveillance de la capacité de la carte SD sur les PA.
- Conception et fabrication de la nouvelle carte électronique du PA pour gérer les écrans dynamiques.

3. Diagrammes UML/SYSML

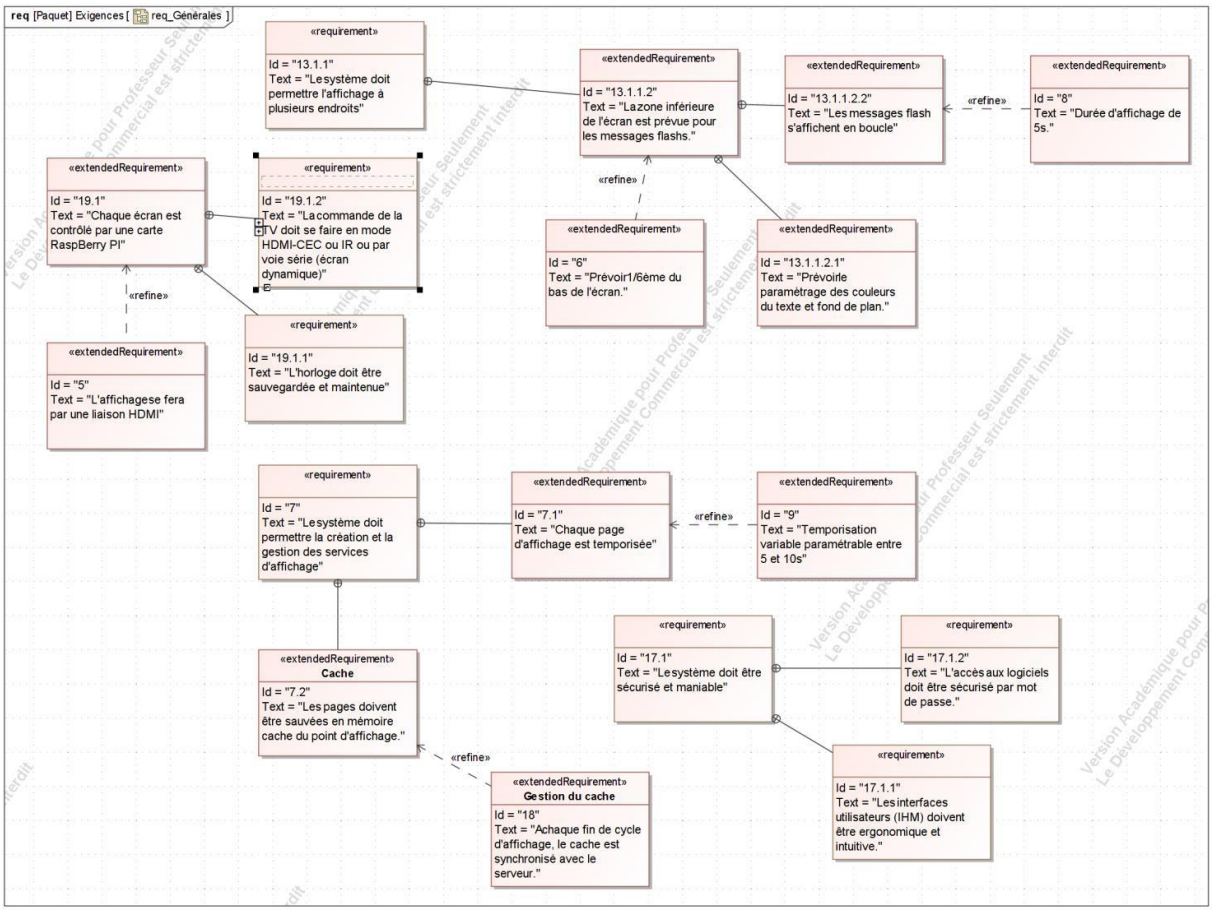
3.1. Diagramme des cas d'utilisation



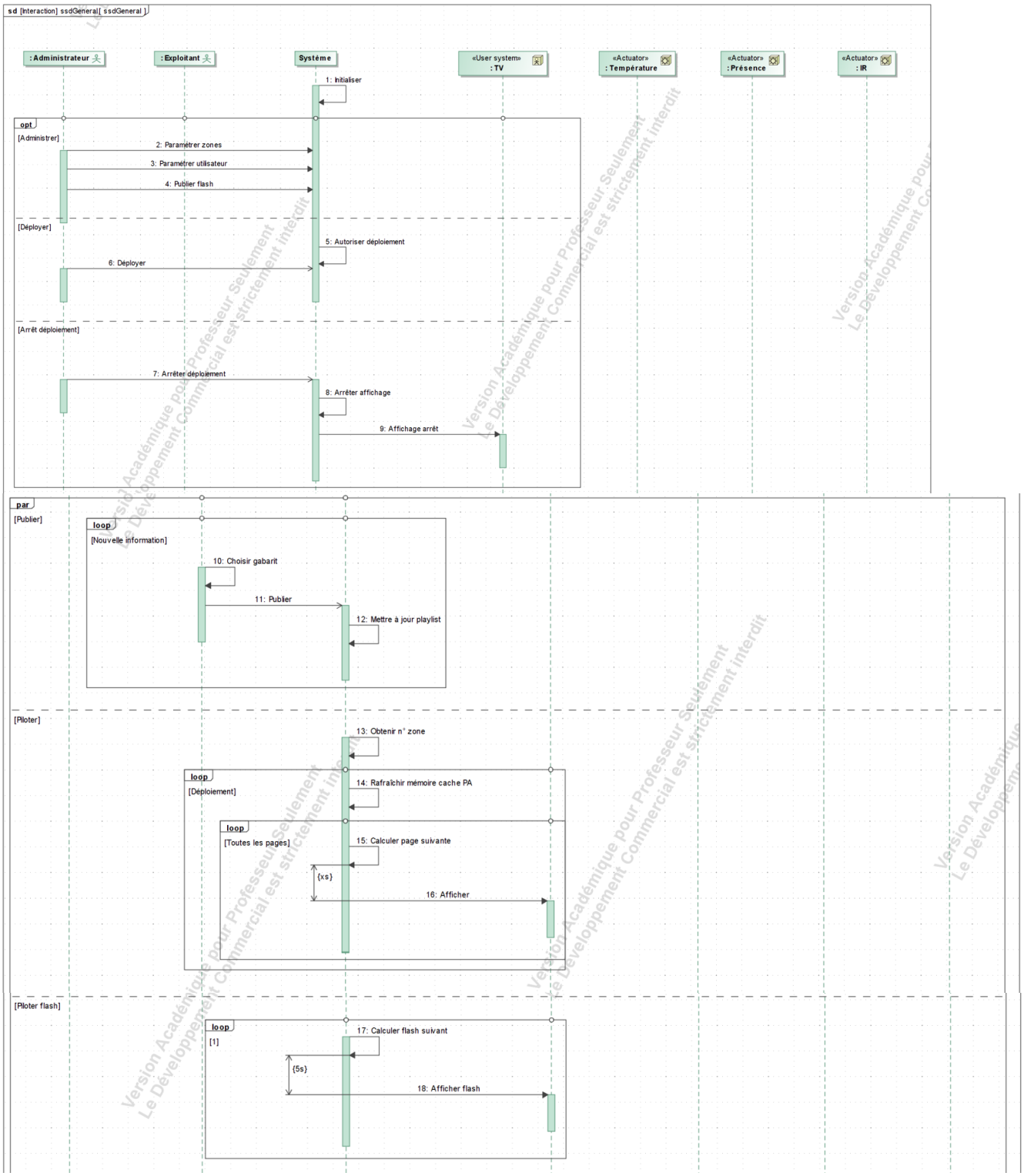
3.2. Diagramme de déploiement

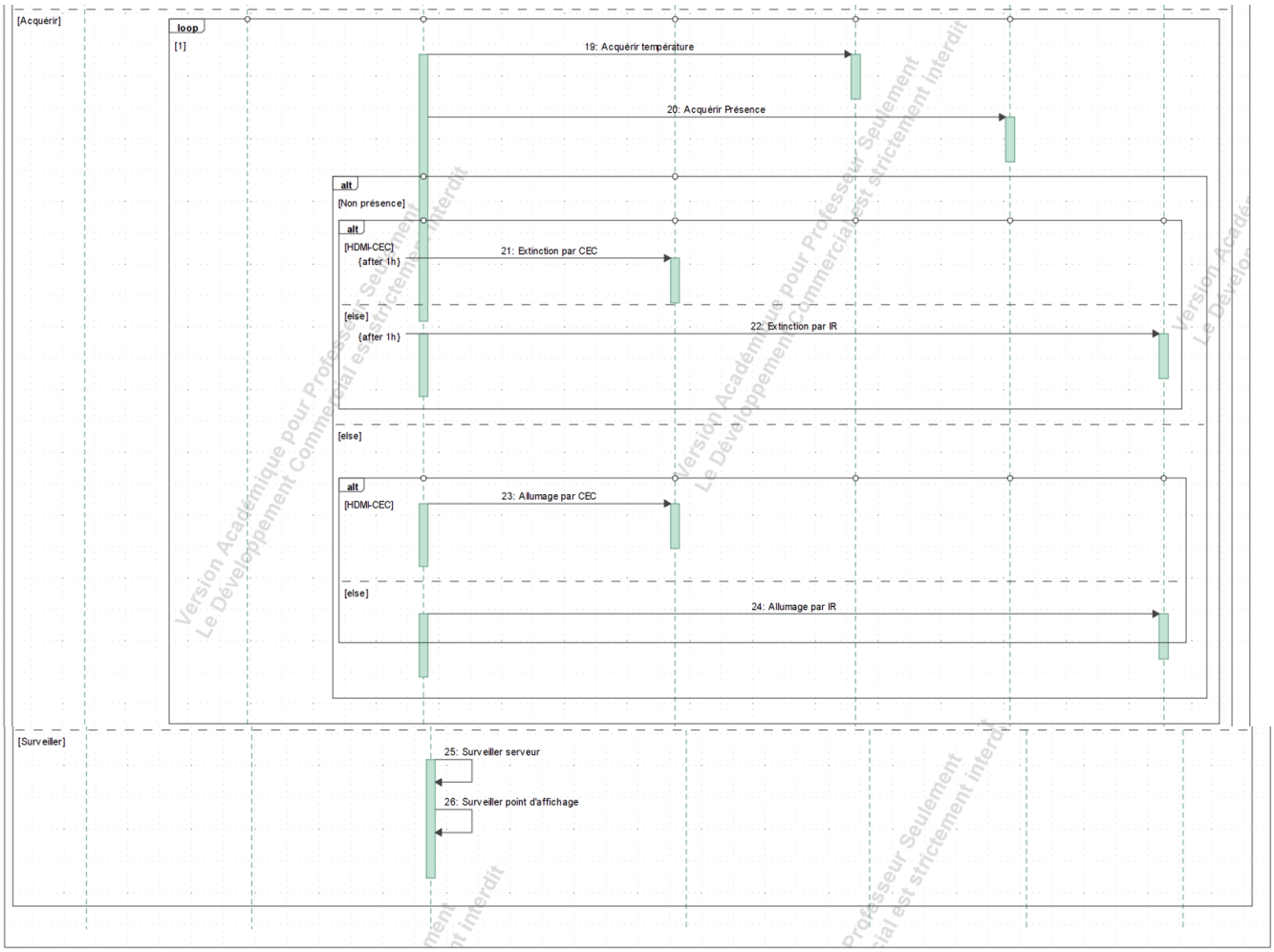


3.3. Diagramme des exigences générales



3.4. Diagramme de séquence système





3.5. Contraintes de réalisation

Contraintes financières (budget alloué) :
Budget de 5000€ déjà payé. Année 2017-2018, achat réalisé de 2 écrans dynamiques.
Contraintes de développement (matériel et/ou logiciel imposé / technologies utilisées) :
La spécification, conception et codage seront modélisés.
Contraintes qualité (conformité, délais, …) :
Maintenable, maniable (ergonomie). Livraison pour le 25 mai 2018 dernier délai.
Contraintes de fiabilité, sécurité :
Les accès logiciels seront sécurisés.

3.6. Ressources mises à disposition des étudiants (logiciels/matériels/documents)

Matériel Raspberry PI2 B+ (7 cartes) et composants électroniques, capteurs Les ordinateurs et logiciels de la section.
Les TV et écrans de PC (HDMI).

4. Répartition des fonctions par étudiant

Étudiant 1 (Serveur) IR	*Liste des fonctions assurées par l'étudiant* Cas d'utilisation : Paramétrer, Authentifier, Publier, Gérer cache. **Tester/comprendre/corriger** la partie logicielle existante. **Amélioration** : Déploiement de la nouvelle version logicielle des PA et fiabilisation du code. Pages d'administration du système. Mise en mémoire cache (SD) des informations (côté serveur). CSS de présentation à unifier et conforme au W3C. Intégration du format vidéo (WEB)	Installation : - Serveur Linux virtualisé - *Services WEB sur serveur* Mise en oeuvre : Configuration : - Serveurs Linux, WEB, MySQL Réalisation : - BDD - *Logiciel PHP client WEB* Documentation : - Logiciel client - Documentation
Étudiant 2 (PA) IR	*Liste des fonctions assurées par l'étudiant* Cas d'utilisation : Piloter, Surveiller, Publier, Gérer cache **Tester/comprendre/corriger** la partie logicielle existante. **Amélioration** : Déploiement de la nouvelle version logicielle des PA et fiabilisation du code. Surveillance du remplissage de la carte SD des PA. Mise en mémoire cache (SD) des informations (côté PA). Intégration d'informations vidéo (PA)	Installation : - *Services clients MySQL sur RPI* - *Qt et cross développement* Mise en oeuvre : - Script SHELL (PA) Configuration : - *Système RPI pour accès BDD* Réalisation : - *Logiciel C/C++ RPI (client MySQL)* Documentation : - *Configuration totale de la RPI*

Étudiant 3 (PA) EC	*Liste des fonctions assurées par l'étudiant* Terminer le déploiement de la partie matérielle de la solution 2016-2017. Comprendre le schéma structurel complet de la carte d'extension associée à la carte Raspberry Pi version 2017. Concevoir/Réaliser/Tester une nouvelle carte d'extension intégrant une communication série permettant de communiquer avec la dernière génération d'écrans utilisés sur le système. Développer une petite application en mode console permettant de vérifier, à minima, le bon fonctionnement de la carte, et plus spécifiquement la transmission série avec l'écran. Puis faire évoluer	Installation : - Des écrans Mise en oeuvre : *- En utilisant la documentation de la version précédente du projet : capteur de température, horloge temps réel, détecteur de présence, commande par CEC et par infrarouge.* Configuration : - Télévisions - Configuration des IP du réseau Réalisation : *- Prototypage rapide pour tester la communication série avec les nouveaux écrans.* *- A partir des fichiers de fabrication de la version précédente, conception d'une carte intégrant également l'interface de communication série.* *- Interconnexion / Fixation de la RPI sur l'écran* Documentation : *- Bibliothèque C/C++ d'accès série à l'écran.* *- Documents de fabrication de la carte. Ces documents devront pouvoir permettre une fabrication industrielle du circuit imprimé.*

5. Compte rendu de réunion

Date	11 Janvier 2018
Secrétaire	BILLY Olivier
Temps de la réunion	45 minutes
Participants	M.Antoine, ANNICELLI Nathan, BILLY Olivier, HUET Valentin
Sujet	Discussion générale sur le projet

IR1

Retrouver les fonctions actuelles sur la nouvelle version. Mise en local du serveur. Format vidéo pas en priorité. Voir les données sur le serveur. Création nouvelle SD (16 Go).

IR2

Fiabilisation code C++. Surveillance du cache de stockage. Vérification des versions. Nouveau format vidéo pas en priorité.

EC

Installation de l'infrarouge. Recherche pour la connectique. Programme permettant de tester. Prototypage.

6. Partie Physique

Le projet :

Le projet AIES a pour but d'afficher des informations sur différents téléviseurs, ils ont des dimensions différentes suivant le modèle et l'emplacement.

HDMI, Colorimétrie et ratio :

La connectique entre l'écran et la raspberry ce fait via HDMI. C'est une interface audio et vidéo mais dans notre cas seul le flux vidéo est utilisé.

La résolution est Full HD 1920*1080p en format 16/9, soit 1920 pixels de largeur et 1080 pixels de hauteur.

Le p correspond a un affichage progressif, les normes vidéos de diffusion d'images progressives sont dénotées par l'utilisation de la lettre **p** (exemple : 480p, 720p) ; à l'inverse, on note **i** (exemple : 1080i) pour les systèmes entrelacés.

Le 1080p (progressif) veut dire que l'image est envoyé en un seul paquet à la vitesse de 24 images/s.
Le 1080i (entrelacé) veut dire que l'image est envoyé en 2 paquets (lignes paires puis impaires) mais à 50 images/s.



Au final le téléviseur affiche bien une image full HD (1920 x 1080 pixels).
Techniquement, le p est plus difficile à gérer car il faut faire passer plus d'infos d'un coup, alors que le i en fait passer 2 fois moins mais 2 fois plus vite.
Historiquement, la diffusion TV a toujours été en i.

Ainsi, la résolution d'une image 1080i (1080 lignes entrelacées) serait équivalente à celle d'une image 720p (720 lignes progressives) malgré un nombre plus important de lignes.

Le ratio 16/9 correspond aux proportions de l'image, c'est-à-dire que la largeur de l'écran vaut seize neuvième de sa hauteur.

L'évolution de l'HDMI :

HDMI version	1.0	1.1	1.2	1.3	1.4	2.0
Date de publication	2002	2004	2005	2006	2009	2013
Fréquence maximale (MHz)	165			340		600
Débit total maximum (Gbit/s)	4,95			10,2		18
Profondeur de couleur maximale (bit/px) *	24			48		
Résolution maximum sur un seul lien à 24-bit/px	1920*1200p60			2560*1600p75	4096*160p30	4096*160p60

* nombre de bits utilisés pour représenter la couleur d'un pixel dans une image.

Quelques caractéristiques :

L'écran est composé de : 1920*1080 = 2 073 600 pixels

Avec 3 bits, un pour chaque couleur primaire RGB : 2 073 600 * 3 = 6 220 800 octets

Nous avons une fréquence de 60 Hz, cela correspond à 60 images par seconde

6 220 800 * 60 = 373 248 000 octets/s soit 373Mo/s

*un **octet** est un multiplet de 8 bits codant une information.

Le standard HDMI a un flux vidéo qui peut atteindre au maximum 680 mo/s, donc suffisant dans notre cas.

7. ANNICELLI Nathan : Technicien IR1

7.1. Introduction

Mon but dans ce projet est de gérer un des aspects pour le client, à savoir l'interface web qui permet de créer, modifier et supprimer les slides sur les points d'affichages du système. Les *slides* sont des diapositives créées grâce à l'interface web, ces *slides* sont diffusés sur les cartes Raspberry, qui les affichent sur les écrans.

Le projet sur lequel nous travaillons est une suite du même projet sur lequel plusieurs étudiants ont déjà travaillés les années précédentes.
De ce fait, le système inclus déjà un système de mise à jour concernant le site web ainsi que la carte Raspberry, le site web, quant à lui, inclus un système de gestion de compte à plusieurs niveaux.
On peut notamment visualiser les points d'affichages depuis le site web.

Le site web devra à terme permettre de gérer plusieurs types de slides, à savoir du texte (déjà en place), des images, des tableaux et enfin des vidéos si le temps le permet.
Le design du site web a déjà été mis à jour par l'ancien étudiant s'occupant du site web, il correspond à un design professionnel et est plus lisible que les versions précédentes.

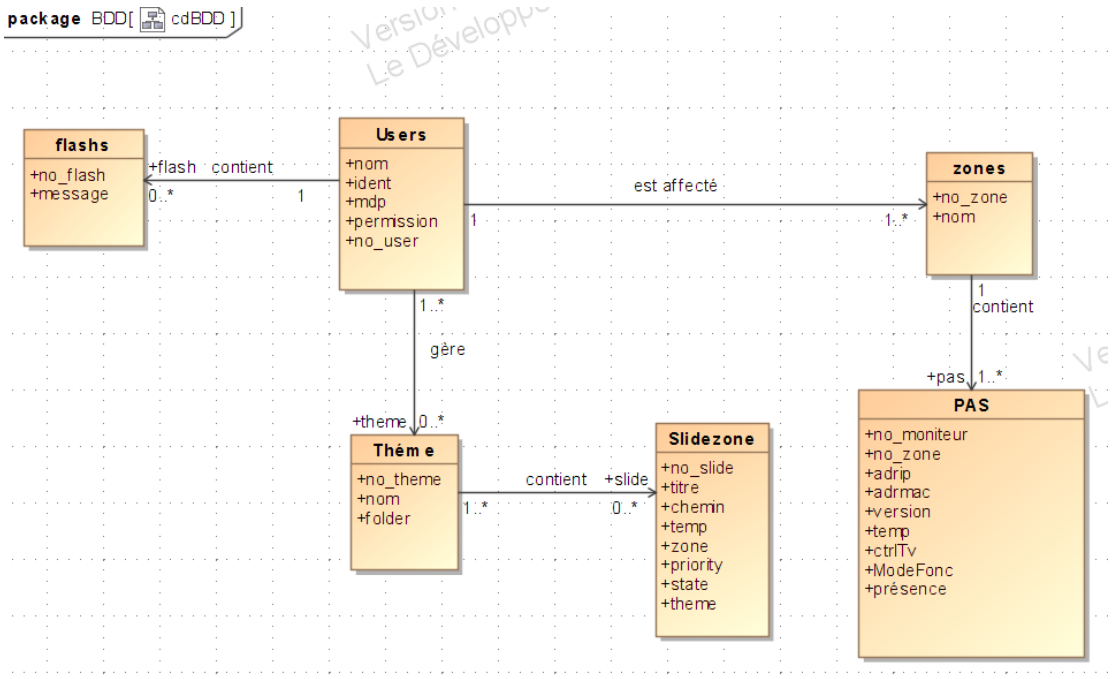
Les tâches que je dois effectuer sont séparées en trois itérations, le voici :

	Taches à effectuer	Justifications des itérations
Itération 1	Prendre connaissance de la version 2016	Pour pouvoir comprendre le code actuelle et de la nouvelle version et la BDD
	Prendre connaissance de v.2017	
	Fiabilisation de la v.2017	Pour rendre opérationnelle cette version et pouvoir la déployer
Itération 2	Déploiement de la v.2017	Pour déployer la nouvelle version et ainsi avoir une version plus professionnelle
	Intégration du format 'image' , 'texte et image' et 'tableaux'	Afin d'avoir les mêmes formats que le version 2016
	Mise en mémoire cache des infos	Afin d'optimiser l'espace de la RPI et le trafic réseau
Itération 3	CSS à unifier et conformiser	Finition du projet pour donner un aspect plus professionnel
	Intégration du format 'vidéo'	Ne peut se faire qu'après la mise en cache pour optimiser le trafic réseau

Pour le commencement du projet, mon but principal était de prendre connaissance de l'existant, donc à savoir, le projet en général (comprendre comment fonctionne le système, ce qu'on attends de nouveau pour cette version), et surtout ma partie donc prendre connaissance des bases de données existantes (pour les 2 versions), ainsi que des sites web (des 2 versions aussi).

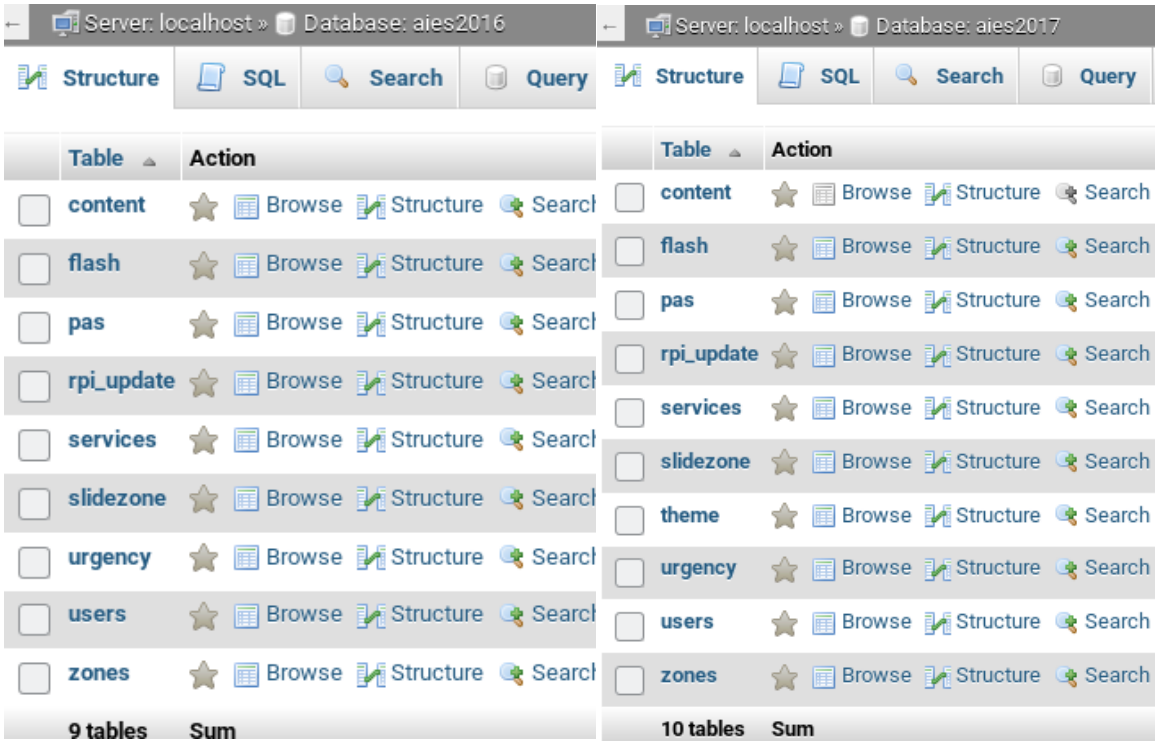
7.2. Prendre connaissance de l'existant

Pour comprendre la base de données un diagramme de classe est à disposition :



Celui-ci nous explique les relations entre les différentes classes.

Les bases de données sont accessibles par le gestionnaire accessible par navigateur *phpMyAdmin*, une capture d'écran permet de comprendre la différence entre les versions 2016 et 2017 :



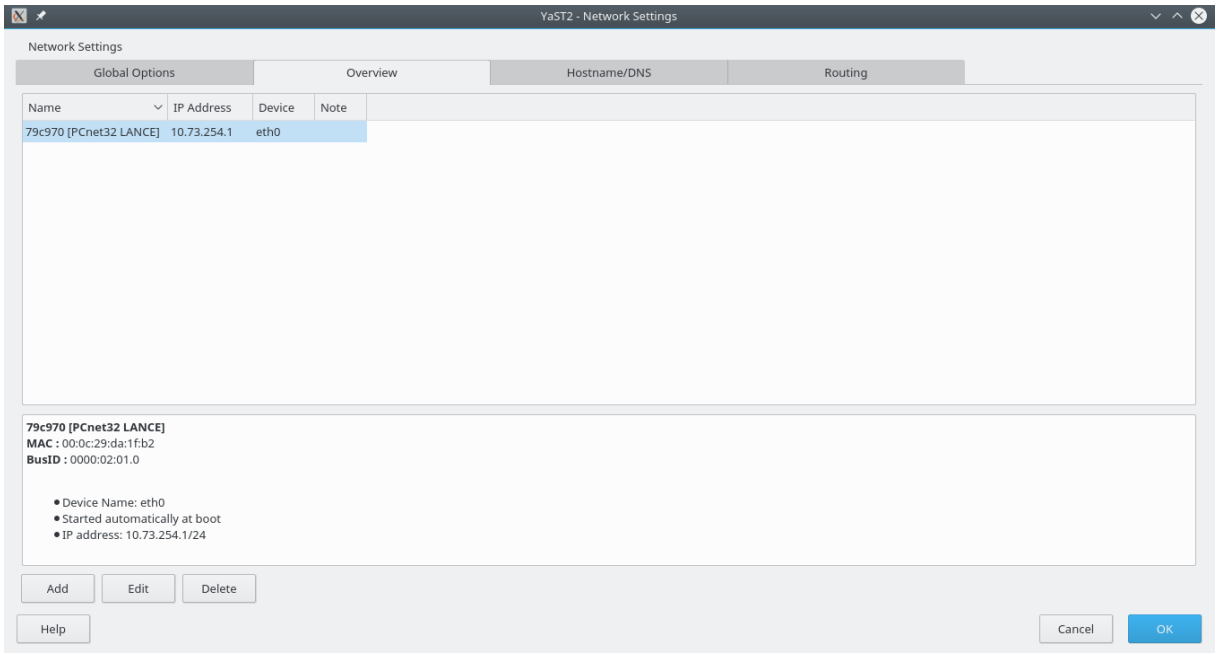
La différence principale entre les 2 versions est que la version 2017 contient une table *« thème »,* cette table change la manière dont une slide est créé.
Dans la version 2017 qui contient cette table, l'utilisateur va choisir le *« thème »* de la slide créé, ce thème va comprendre un fichier de création d'une slide, un fichier de modification de la slide, un fichier qui définit le style graphique du thème, ce thème définit le type de slide créé.



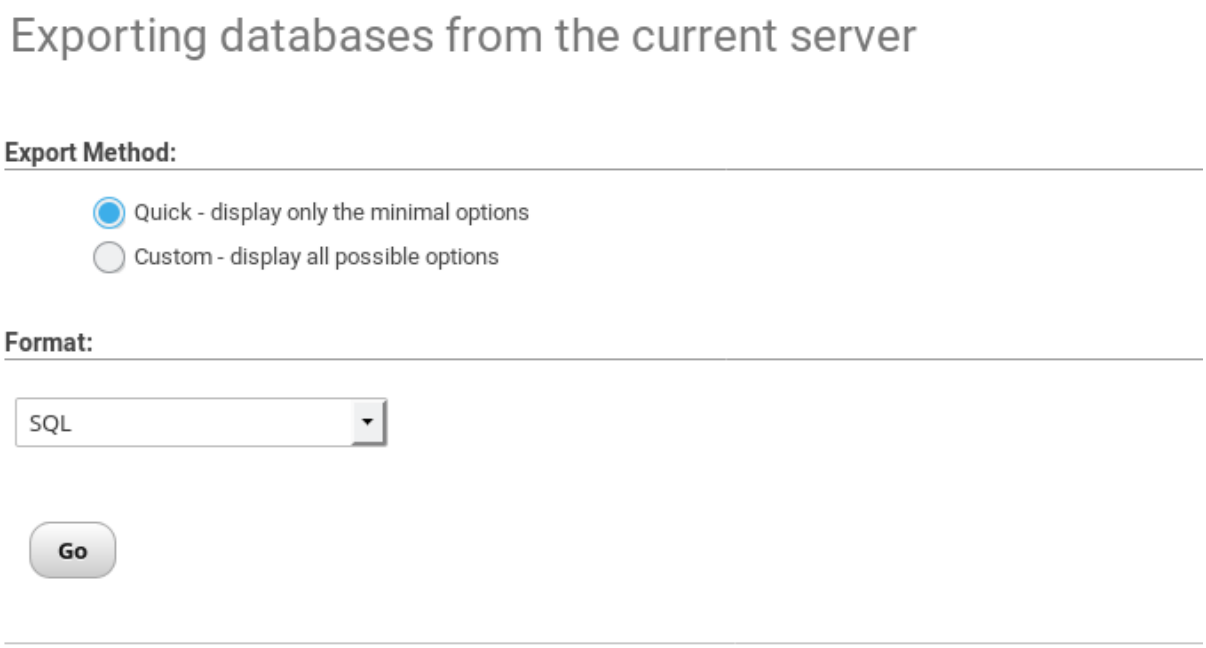
7.3. Serveur local

Afin de fiabiliser la version 2017, j'ai décidé de mettre en place un serveur local sur une machine virtuelle qui correspond trait pour trait au serveur actuel en fonctionnement.
Ce serveur local a pour but de servir de serveur test afin de tester toutes les modifications que je pourrais apporter dans le futur sans corrompre le fonctionnement du serveur actuel (puisque celui-ci est utilisé pour l'affichage du lycée).

Pour ma machine virtuelle, j'ai utilisé le logiciel *Vmware*, j'ai donc utilisé la même distribution de linux que sur le serveur, à savoir *OpenSUSE 13.2*.
Concernant la configuration j'ai installé les paquets nécessaires au fonctionnement d'un serveur local, donc *LAMP (Linux Apache MySQL phpMyAdmin)*. J'ai aussi configuré le serveur avec la même adresse que celui en fonctionnement pour ne pas avoir de problème d'accès au cas où un fichier renvoie à l'adresse du serveur.



Après cela il fallait importer les bases de données ainsi que les fichiers du serveur.
Donc pour importer/exporter les bases de données je me suis servi de l'outil du gestionnaire *phpMyAdmin* :

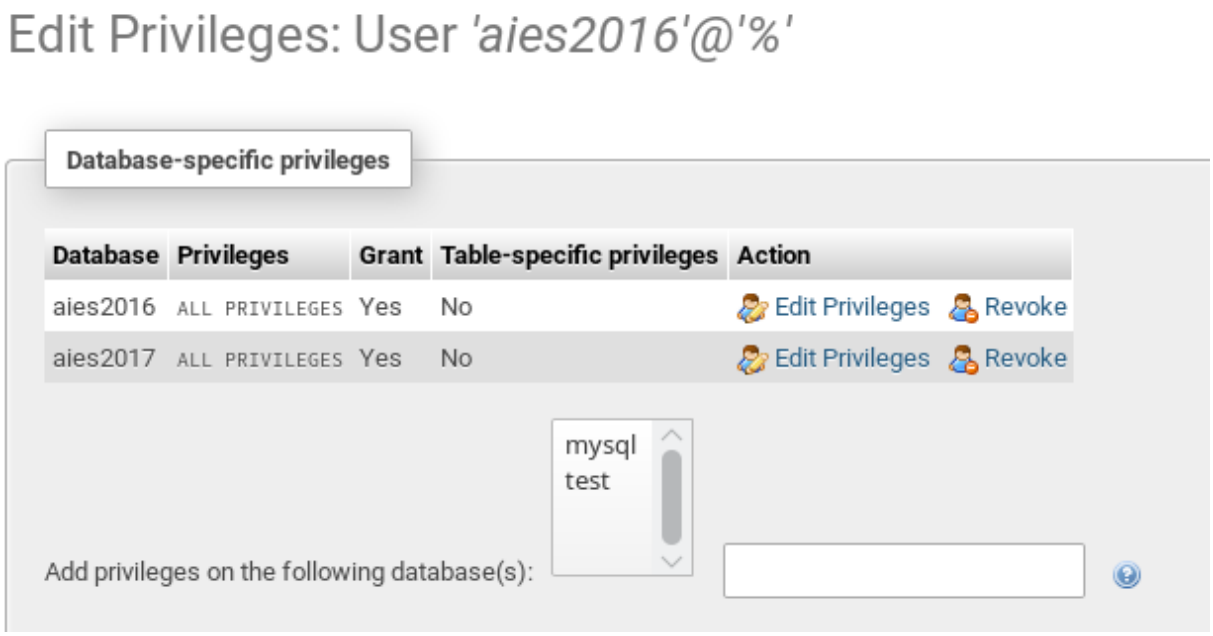


Durant l'importation j'ai rencontré un petit problème, résolu grâce à une solution trouvé sur internet. Le problème était que lors de l'importation, la base de données ne se créait pas, pour résoudre cela, il fallait passer en mode d'exportation custom et coché *« Add create database /use statement ».*
Pour rendre la base de données opérationnel, il a fallu recréer les comptes et réaffectez les privilèges aux utilisateurs sur les bases de données pour rendre les sites web accessible.

Users overview

	User name	Host	Password	Global privileges	Grant	Action
☐	Any	%	--	USAGE	No	Edit Privileges Export
☐	Any	linux-9f0e	No	USAGE	No	Edit Privileges Export
☐	Any	localhost	No	USAGE	No	Edit Privileges Export
☐	aies2016	%	Yes	USAGE	No	Edit Privileges Export
☐	aies2016	localhost	Yes	USAGE	No	Edit Privileges Export
☐	root	127.0.0.1	No	ALL PRIVILEGES	Yes	Edit Privileges Export
☐	root	::1	No	ALL PRIVILEGES	Yes	Edit Privileges Export
☐	root	linux-9f0e	No	ALL PRIVILEGES	Yes	Edit Privileges Export
☐	root	localhost	Yes	ALL PRIVILEGES	Yes	Edit Privileges Export

Check All *With selected:* Export



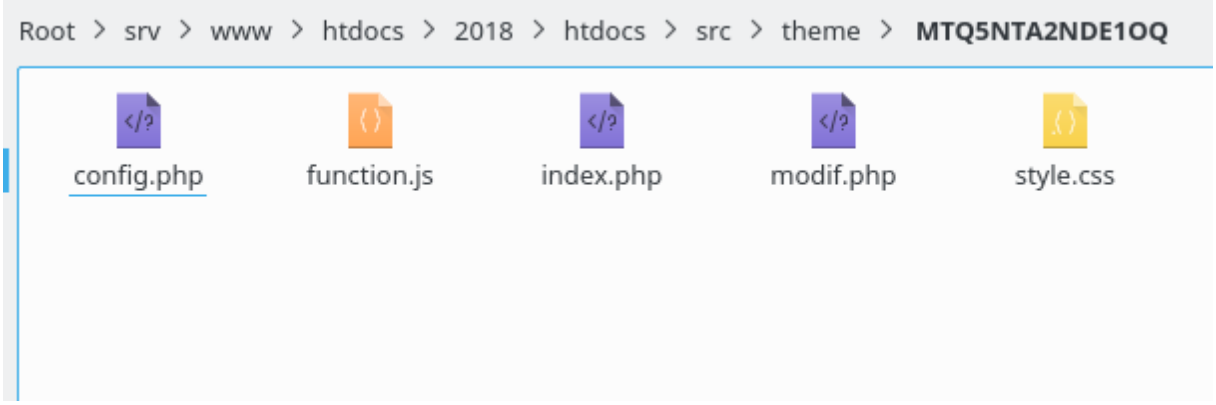
Un autre problème rencontré fut pendant la copie des fichiers du serveur, après avoir effectué la copie la création de slides ne fonctionnait plus il suffisait simplement de changer les droits des répertoires.

7.4. Création du thème « tableaux »

Pour l'avancement de la nouvelle version, ma principale tâche à effectuer était de créer le thème « tableaux » pour pouvoir faire un tableau d'absence des professeurs et donc permettre un déploiement de la nouvelle version.

Pour créer ce thème, je me suis inspiré de celui existant, à savoir le thème texte, pour rester dans la même optique de développement ainsi que dans la même charte graphique.
Un thème se constitue de quatre fichiers principaux qui sont compris dans un seul dossier correspondant au thème choisi. Les quatre fichiers sont :

- Le fichier *config.php* qui sera la page affiché lorsque l'on souhaite créer une slide.
- Le fichier *modif.php* qui sera la page de modification d'une slide.
- Le fichier *style.css* qui définit le style graphique des pages du thème.
- Le fichier *function.js* qui contient les fonctions javascript utilisées dans les autres pages du thème.



Le premier fichier à créer a été le fichier *config.php*, je ne suis pas parti de zéro, j'ai repris le code du fichier *config.php* du thème « texte » afin de me faciliter la modification du code.

Le but de la modification était que, dès l'ouverture de la page un tableau s'affiche (au lieu du texte du thème « texte ») et que l'on puisse écrire dans les cases du tableau.

Pour réaliser cette modification, j'ai d'abord modifié la zone de texte dans le * *.

```
<div id="content" class="blue">
    <h1 id="content_h1" onclick="content_select(this, 'text')">Cliquer ici pour ecrire !</h1>
    <p id="content_p" onclick="content_select(this, 'text')">Cliquer ici pour ecrire !</p>
</div>
```

J'ai donc remplacé le *<p>* (zone de texte) par un tableau en langage html.
J'avais d'abord réalisé le tableau ligne par ligne en html mais suite aux conseils de mon professeur, j'ai créé une fonction php permettant de générer le tableau.

```
<div id="content" class="blue">
    <h1 id="content_h1" onclick="content_select(this, 'text')">Cliquer ici pour ecrire !</h1>

        <?php
        $nbRow = 12 ;
        $nbCol = 4;
        echo "<table>";
        echo "<tr><th>Enseignants</th><th>Absent du</th><th>au</th><th>info</th></tr>";
        for($r=0 ; $r<$nbRow ; $r++) {
            echo "<tr>";
            for ($c=0 ; $c<$nbCol ; $c++) {
                echo "<td id='content_td' onclick='content_select(this, \"text\")'></td>";
            } // for $c
            echo "</tr>";
        } // for $r
        echo "</table>";
        ?>


</div>
```

Le nombre de lignes et de colonnes est défini à l'avance puisque le tableau sert de tableau d'absence et pour permettre un remplissage plus simple, le tableau occupe déjà entièrement la page.
La fonction php crée donc des cases de tableau qui sont des cases de texte dans lesquelles un clic de souris active la fonction javascript « content_select ».

Voilà à quoi ressemble la page avec le tableau :



La zone pour écrire le texte dans chaque case se situe en dessous du tableau.

La seconde modification du fichier *config.php* consistait à changer le nom du fichier css utilisé par la page et changer le numéro de thème.

```
$top_content =
"<!DOCTYPE html>
<html>
<head>
<title>Lycée Alphonse Benoit</title>
<meta http-equiv=\"Content-Type\" content=\"text/html; charset=UTF-8\">
<link rel=\"stylesheet\" type=\"text/css\" href=\"../../src/css/global.css\"/>
<link rel=\"stylesheet\" type=\"text/css\" href=\"../../src/theme/MTQ5NTA2NDE1OQ/style.css\"/>
<link rel=\"shortcut icon\" type=\"image/x-icon\" href=\"../../src/img/logo_aies.png\"/>
</head>
```

Il m'a suffit de mettre le nom du dossier correspondant au bon thème, donc le dossier qui contient le fichier css pour le thème « tableau ».

```
$req = $db->query("INSERT INTO slidezone (title, path, time, zone,
    priority, state, date_create, theme, creator) VALUES ('" . $_POST['
    title'] . "', '/rpi/slide/" . $newfile . "', " . $_POST['time'] . ",
    " . $_POST['zone'] . ", '" . $_POST['priority'] . "','arch', '" . $
    date_create . "',4, '" . $_SESSION['common'] . "')");
```

Ici, j'ai uniquement changé le numéro correspondant au thème (donc ici j'ai mis le numéro 4 pour le thème « tableau »). Cette modification est importante puisqu'elle concerne le stockage dans la base de données et renseigne le thème de la slide.

Le second fichier à modifier fut le fichier *style.css*, pour garder le même style graphique sur tous les thèmes j'ai donc repris le fichier du thème « texte » et ajouter ce qu'il manquait pour le thème « tableau ».
Les premières choses à ajouter furent les paramètres pour le tableau que j'ai créé auparavant.

```
#content table {
    width: 99.9%;
    height: 99.9%;
    border-collapse: collapse;
    border : 1px solid black;
    font-size: 150%;
}
#content th {
    width: 25%;
    max-width: 600px;
    border : 1px solid black;
    height: 50px;
    padding: 15px;
    text-align: left;
    background-color: #9E9E9E;


}
#content td {

    border : 1px solid black;
    vertical-align: center;
    padding: 10px;
    column-width: 25%;
    height: 26px;
}
```

Pour les paramètres, j'ai donc décomposé tous les paramètres du tableau, ceux pour le tableau général qui se trouve dans le * * (#content table), ceux pour les cases contenant les intitulés de colonnes (#content th) et ceux pour les cases du tableau (#content td).
Les paramètres du tableau (#content table) concerne sa taille (largeur/hauteur) renseigné en pourcentage pour s'adapter à la page, ses bordures (taille de la bordure, séparation ou non des bordures) ici la taille de la bordure est en pixels et enfin la taille de la police utilisé dans le tableau.

Dans les paramètres des intitulés de colonnes (#content th), sont définis la largeur de la case (pour avoir 4 cases de la même taille), la hauteur de la case, l'alignement du texte dans la case, l'espace entre la bordure et le texte (padding) et enfin la couleur de la case qui est défini sous une forme hexadécimale correspondant aux valeurs des trois couleurs (rouge, vert, bleu) donc ici #9E9E9E qui est du gris.

Enfin pour les paramètres des cases du tableau, concernent les bordures, l'espace entre la bordure et le texte, la largeur de la colonne et la hauteur de la case.

Tous ces paramètres contribuent à embellir le tableau et sont indissociables du fichier *config.php* et *modif.php* sinon le rendu de la page en sera très affecté et peut même rendre la page illisible.

Concernant le fichier *function.js*, je n'ai quasiment rien modifié dedans puisque les fonctions utilisées dans le fichier *config.php* et *modif.php* restent les mêmes pour le thème « texte » et le thème « tableau ». J'ai uniquement modifié le « content_p » en le transformant en « content_td » dans la fonction *create().*

```
var res = document.getElementById("content_h1").innerHTML.length + document.getElementById("content_td").innerHTML.length;
```

Le dernier fichier à modifier fut le fichier *modif.php* qui sert à modifier une slide créée auparavant.
Concernant le code, une seule chose était à modifier pour le rendre utilisable pour le thème « tableau », il s'agissait de modifier le fichier *style.css* utilisé.

```
<meta http-equiv=\"Content-Type\" content=\"text/html; charset=UTF-8\">
<link rel=\"stylesheet\" type=\"text/css\" href=\"../../src/css/global.css\"/>
<link rel=\"stylesheet\" type=\"text/css\" href=\"../../src/theme/MTQ5NTA2NDE1OQ/style.css\"/>
<link rel=\"shortcut icon\" type=\"image/x-icon\" href=\"../../src/img/logo_aies.png\"/>
```

7.5. Ajout de couleurs et d'une nouvelle zone 'Partout'

Pour permettre le déploiement de la nouvelle version et avoir des fonctionnalités équivalentes aux anciennes versions, il a fallu que je rajoute dans les options de création d'une slide, un choix de couleurs suffisant pour avoir une variété visuelle conséquente à proposer.

Pour ajouter ces nouvelles couleurs, il a fallu premièrement que je rajoute de nouvelles classes dans le fichier *style.css* (dans les deux fichiers *style.css* donc celui du thème « texte » et celui du thème « tableau »).

```
.blue
{
    background-color: rgb(43,57,105);
    color: white;
}

.blue h1
{
    background-image: linear-gradient(to right, rgb(3,9,29), rgb(9,32,65));
}


.lightskyblue
{
    background-color: lightskyblue;
    color: black;
}

.lightskyblue h1
{
    background-image: linear-gradient(to right, rgb(70,120,190), rgb(100,190,210));
}
```

Les classes sont codées de la manière suivante :

- La couleur de fond est défini par la ligne *background-color*, la couleur peut se définir de trois manières différentes, soit en donnant les valeurs RGB de la couleur

  ```
  background-color: rgb(43,57,105);
  ```

  Soit en donnant le nom de la couleur qui est reconnu par le langage HTML

  ```
  background-color: lightskyblue;
  ```

  Soit en donnant la valeur sous forme hexadécimale

  ```
  background-color: #DCDCDC;
  ```

- La couleur de texte est définie par la ligne *color* `color: white;`
- La couleur du fond du titre de la slide est donnée dans une sous-classe h1 dans laquelle est défini un dégradé de la couleur avec la ligne *linear-gradient*

```
.lightskyblue h1
{
    background-image: linear-gradient(to right, rgb(70,120,190), rgb(100,190,210));
}
```

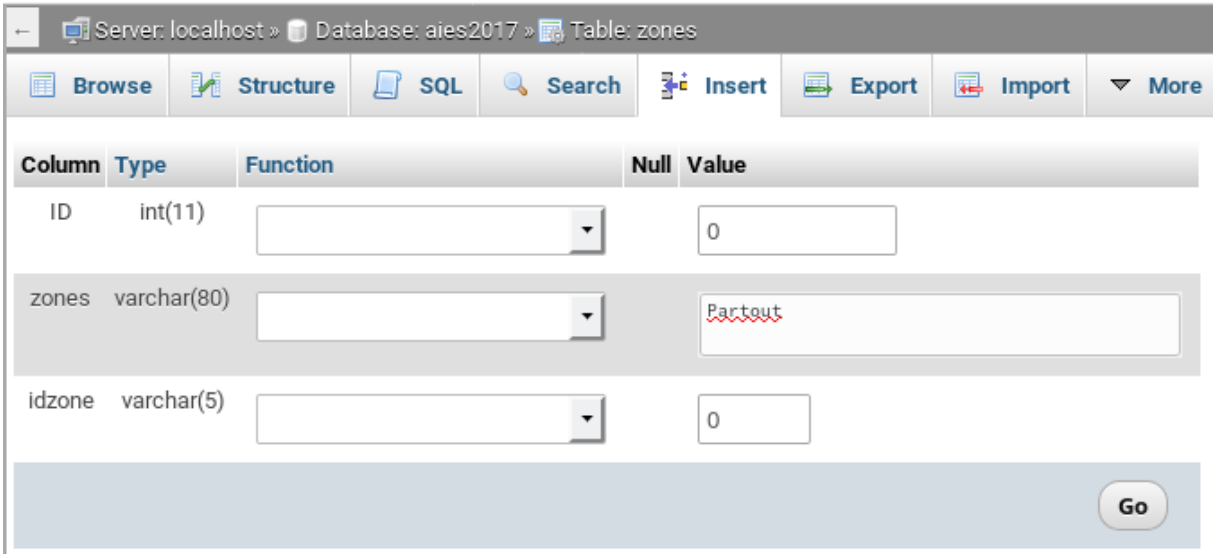
Une fois les classes créées, il fallait les rajouter dans les fichiers *config.php* et *modif.php* pour pouvoir les sélectionner et s'en servir.
Les couleurs sont codées dans le * *, elles sont dans un tableau dans lequel un clic de souris permet de sélectionner la bonne classe (donc la bonne couleur) et changer de couleur sur la page en cours.

```
<div id="select_color">
    <table>
        <tr><td style="border: 2px red solid;" class="blue" onclick="teinte_change(this, 'blue');">
        </td><td class="lightskyblue" onclick="teinte_change(this, 'lightskyblue');"></td><td class
        ="gainsboro" onclick="teinte_change(this, 'gainsboro');"></td><td class="orange" onclick="
            teinte_change(this, 'orange');"></td><td class="indianred" onclick="teinte_change(this,
             'indianred');"></tr>
            <tr><td class="darkslategrey" onclick="teinte_change(this, 'darkslategrey');"></td><td
            class="darkolivegreen" onclick="teinte_change(this, 'darkolivegreen');"></td><td class=
            "lightsteelblue" onclick="teinte_change(this, 'lightsteelblue');"></td><td class="pink"
             onclick="teinte_change(this, 'pink');"></td><td class="lemonchiffon" onclick="
                teinte_change(this, 'lemonchiffon');"></tr>
            </table>
        </div>
```

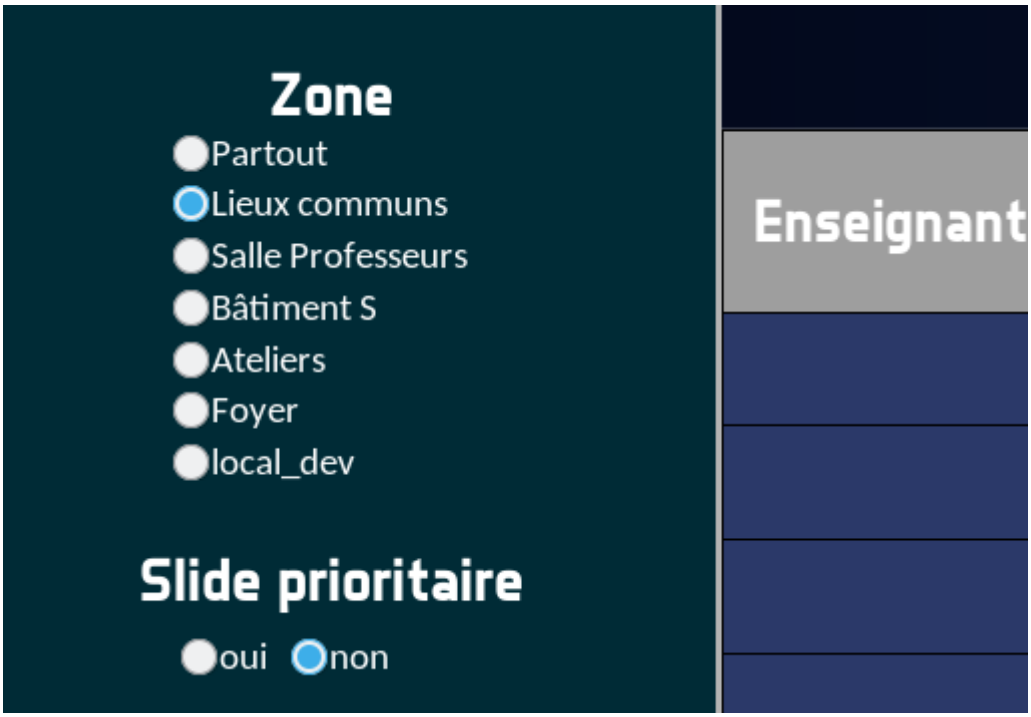
Chaque case de ce tableau correspond à une classe à sélectionner. Pour rajouter les couleurs il m'a suffit de mettre les bonnes classes dans *class=nom de la classe* et dans la fonction *teinte_change.*

```
<td class="lightskyblue" onclick="teinte_change(this, 'lightskyblue');">
```

Pour pouvoir avoir les mêmes fonctionnalités que les anciennes versions de AIES, il fallait que je rajoute une zone *Partout*, c'est-à-dire une zone qui définit l'affichage des slides dans toutes les zones.
Donc pour rajouter cette zone, il a fallu que je rajoute la zone dans la base de données la nouvelle zone.



Pour cette nouvelle zone j'ai mis le *ID* et le *idzone* à *0* pour que cette zone soit considéré comme la première et qu'elle s'affiche en premier lors du choix des zones dans le *config.php*.



J'ai aussi modifié une valeur dans les fichiers *config.php* (donc ceux des deux thèmes fonctionnant).

```
while($data_zone = $req_zone->fetch())
{
    if($data_zone['idzone'] ==0) echo "<input type=\"radio\" name=\"zone\"
        value=\"" . $data_zone['idzone'] . "\" checked><span> " . $
        data_zone['zones'] . "</span></input><br>";
    else echo "<input type=\"radio\" name=\"zone\" value=\"" . $data_zone['
        idzone'] . "\"><span> " . $data_zone['zones'] . "</span></input><
        br>";
}
```

Le changement est une modification d'un *1* en *0* qui correspond à la zone où le curseur de choix de zones sera au chargement initial de la page donc en mettant *0*, comme l'id de la zone *Partout* est *0*, le curseur sera sur la zone *Partout*.

7.6. Problèmes rencontrés

Durant le projet plusieurs problèmes se sont présentés, lors de la création du tableau, un problème de retour à la ligne que ne s'effectuait pas lorsqu'un mot était trop grand, pour le résoudre j'ai du rajouter dans le code du fichier *style.css* une ligne permettant de limiter la largeur de la colonne.

```
#content td {

    border : 1px solid black;
    vertical-align: center;
    padding: 10px;
    column-width: 25%;
    height: 26px;
}
```

J'ai aussi rencontré un problème concernant l'oublie de modification des fichiers *.css* correspondant au bon thème et par conséquent, je n'obtenais pas le résultat que je souhaitais. Un problème similaire était présent sur la création de slide puisque je n'avais pas modifié le numéro du thème dans le fichier *config.php*

```
$req = $db->query("INSERT INTO slidezone (title, path, time, zone,
    priority, state, date_create, theme, creator) VALUES ('" . $_POST['
    title'] . "', '/rpi/slide/" . $newfile . "', " . $_POST['time'] . ",
    " . $_POST['zone'] . ", '" . $_POST['priority'] . "','arch', '" . $
    date_create . "',4, '" . $_SESSION['common'] . "')");
```

Le principal qui s'est posé à moi, fut le temps, puisque sur l'itération n°2, il était prévu que je réalise deux thèmes, donc le thème « tableau » et le thème « image », cependant par manque de temps, je n'ai pu que réaliser le thème « tableau » et certaines choses ce sont ajoutées au projet, tel que la demande de rendre le système déployable le plus rapidement possible.
La tenue d'un journal de bord sous la forme d'un compte twitter, m'a permis de renseignais l'avancement du projet après chaque intervention et ainsi y renseigner les problèmes rencontrés et les solutions trouvées.



8. BILLY Olivier : Technicien IR2

8.1. Introduction

Ma partie concerne la partie logicielle sur la Raspberry Pi, il s'agit surtout d'une amélioration du projet existant. Ce système évolutif a de nombreuses fonctionnalités qui seront modifiés, et d'autres rajouter. Ces fonctions ont un but informatif.

Le code C++ existant est modifié, les classes ont-elles aussi étaient revues pour simplifié la compréhension.

8.2. Taches à effectuer

J'ai divisé ma partie en 3 itérations, cela me permet de gérer le temps en fonction de ce que je dois accomplir.

Itérations	Raisons
Prendre connaissance version actuelle (2016) et de la nouvelle version en cours de développement (2017).	IT1 Comprendre avant d'améliorer. Le capteur de présence et infrarouge ne fonctionne pas dans la version 2017.
Analyse du travail à effectuer pour la version 2018	IT1 Analyse et Conception pour retrouver le fonctionnement du capteur de présence et de l'émetteur infrarouge.
Créer la carte SD 2018 Affichage des états de présence et de l'émetteur infra rouge.	IT1
Mise en mémoire cache des infos coté PA	IT2 Solution
Surveillance du remplissage carte SD des PA	IT2 Solution
Intégration du format vidéo	IT3 Ne peut se faire qu'après la mise en cache pour optimiser le trafic réseau.

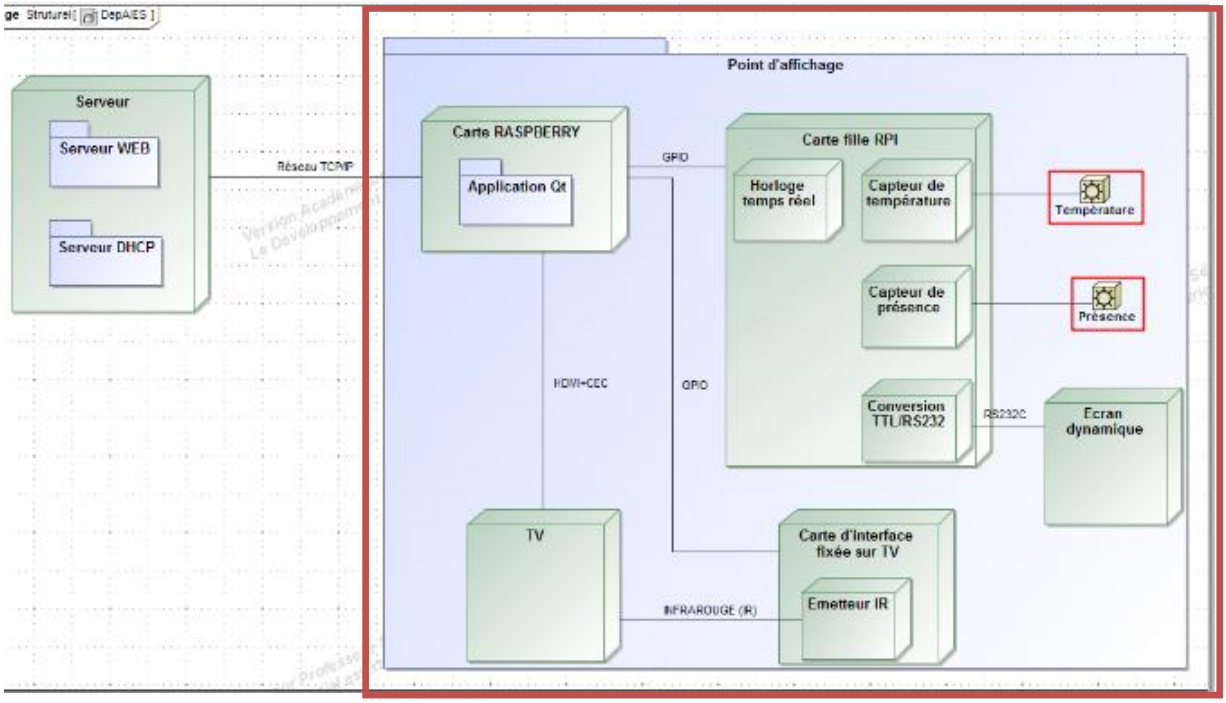
8.3. Programme Principale AIES2018

Le programme est en 2 parties :

-Une première partie, la plus grande, il s'agit d'un navigateur web affichant les différentes diapos stockées sur le serveur. (QWebView)

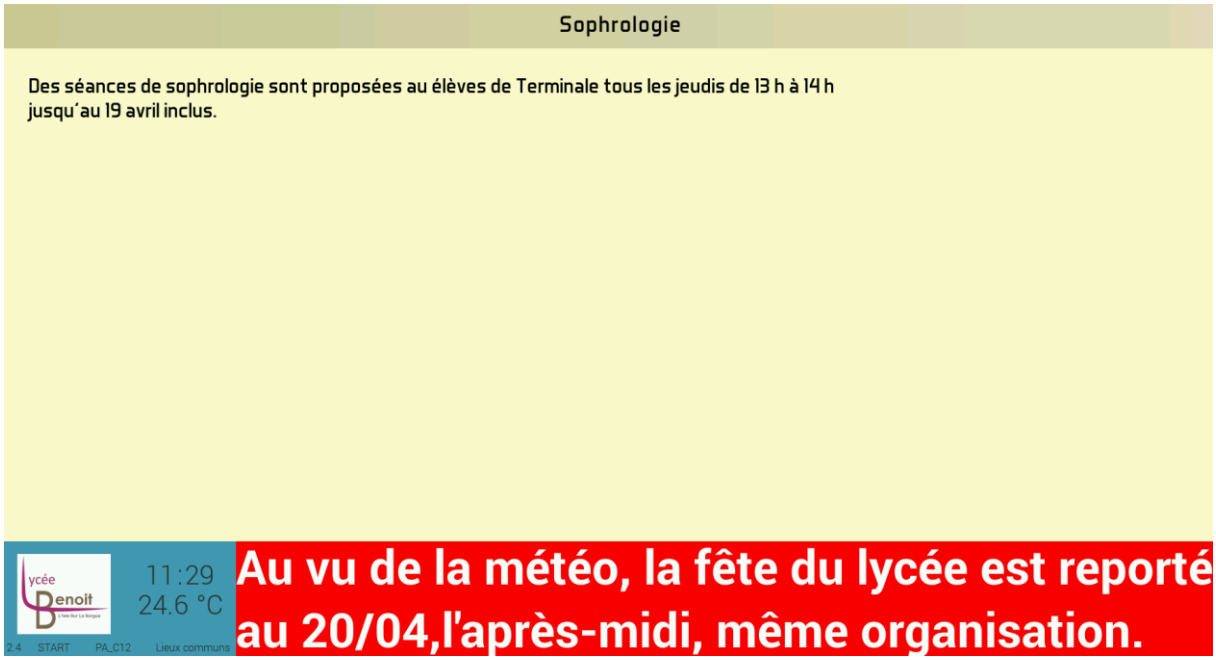
- La deuxième partie en bas de l'écran, divisé en deux, a gauche des informations (Version, État de l'infrarouge, Heure…) et un bandeau affichant les messages flashs.

8.4 Mise en évidence de ma partie

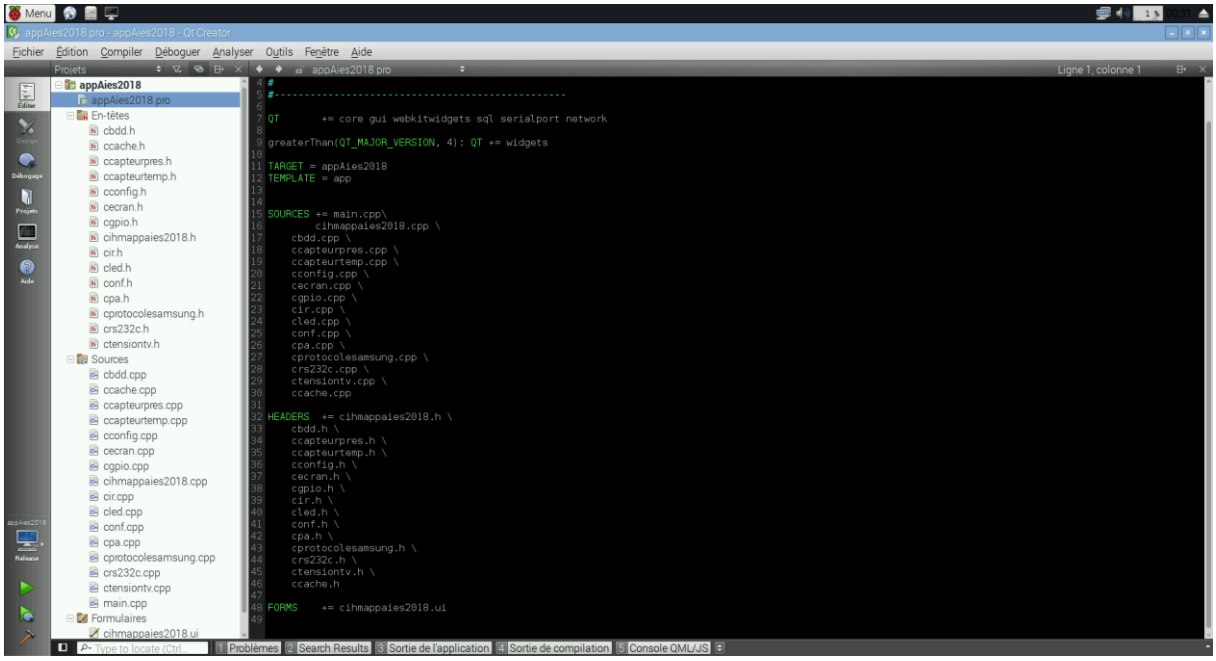


J'interviens sur cette partie encadrée, principalement sur la Raspberry Pi qui est en liaison avec l'écran, et les différents capteurs.

8.5 Visuel de l'application AIES2018



8.6 Programme Principal AIES2018

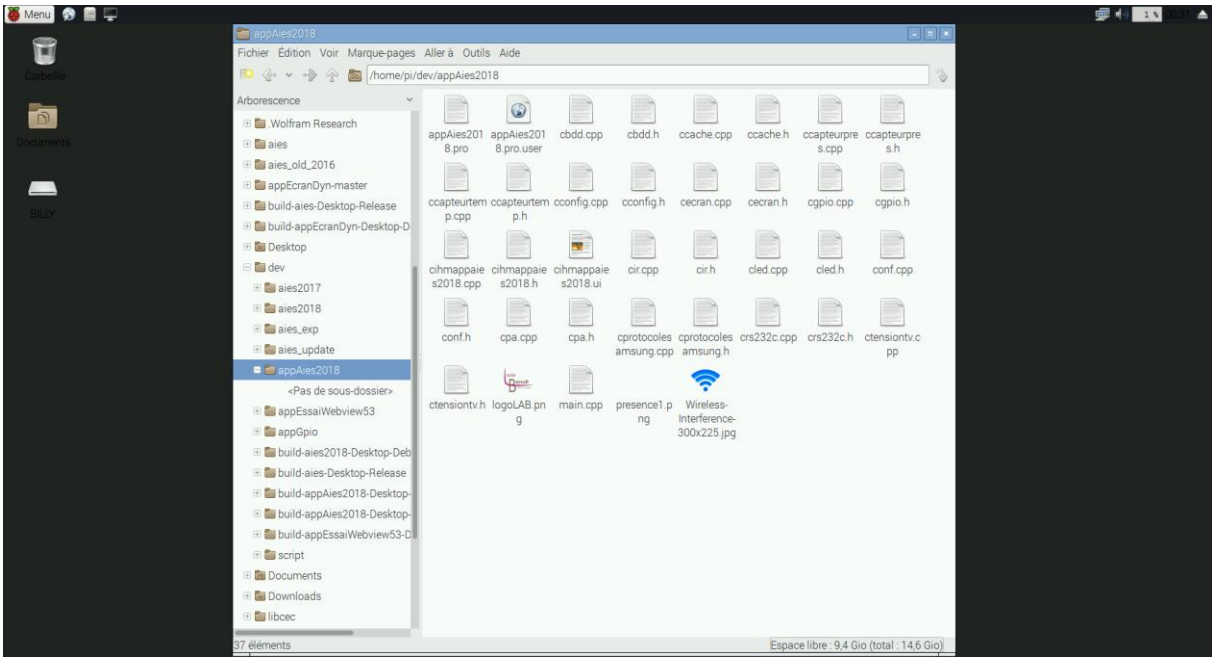


Sur cette capture d'écran, fait sur le logiciel QtCreator 5.3, on voit les différentes classes* qui composent l'application AIES2018.

*La classe déclare, d'une part, des attributs représentant l'état des objets et, d'autre part, des méthodes représentant leur comportement.

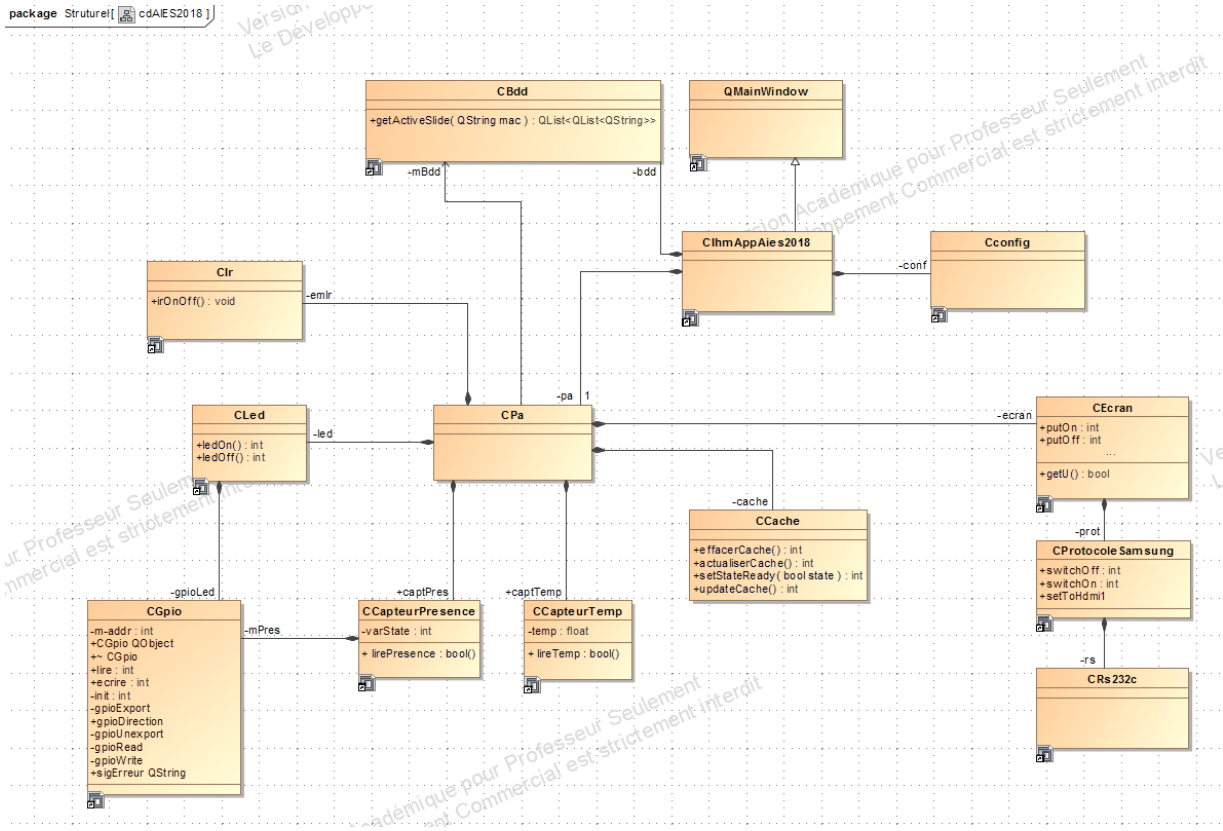
Une classe représente donc une catégorie d'objets.

Toutes renommées pour la version 2018.



Il y a d'abord l'IHM (Interface homme machine) **ClhmAppAie2018** qui regroupe tout ce qui concerne l'interface, la classe **CCapteurPres** qui concerne la partie du capteur de présence, de même que la classe **CCapteurtemp. CBdd** concernant la relation avec la base de données, lorsque nous avons besoin d'aller y chercher une information. **CEcran** traite de tout ce qui ce passe avec les différents types d'écrans. **CGpio** traite les entrées/sorties des leds, cette dernière est en relation avec **CLed.** La classe **CRs232c** concerne une norme standardisant une voie de communication de type série.

8.7 Diagramme de classe AIES2018



Sur ce diagramme de classe on peut y voir les différentes classes qui composent l'application AIES 2018.

La classe principale est **ClhmAppAies2018,** c'est celle qui s'occupe de l'IHM (Interface Homme Machine), l'utilisateur transmet des données vers le programme qui les traite et retourne les résultats à l'humain. Le formulaire est l'intermédiaire entre l'**homme** et la **machine**, ce qui caractérise le fait que l'on nomme les formulaires d'une application : **interface** « **homme machine** » (**IHM**).

Dans la classe **CPa** on retrouve les différentes méthodes qui interagissent avec les écrans et les raspberry.

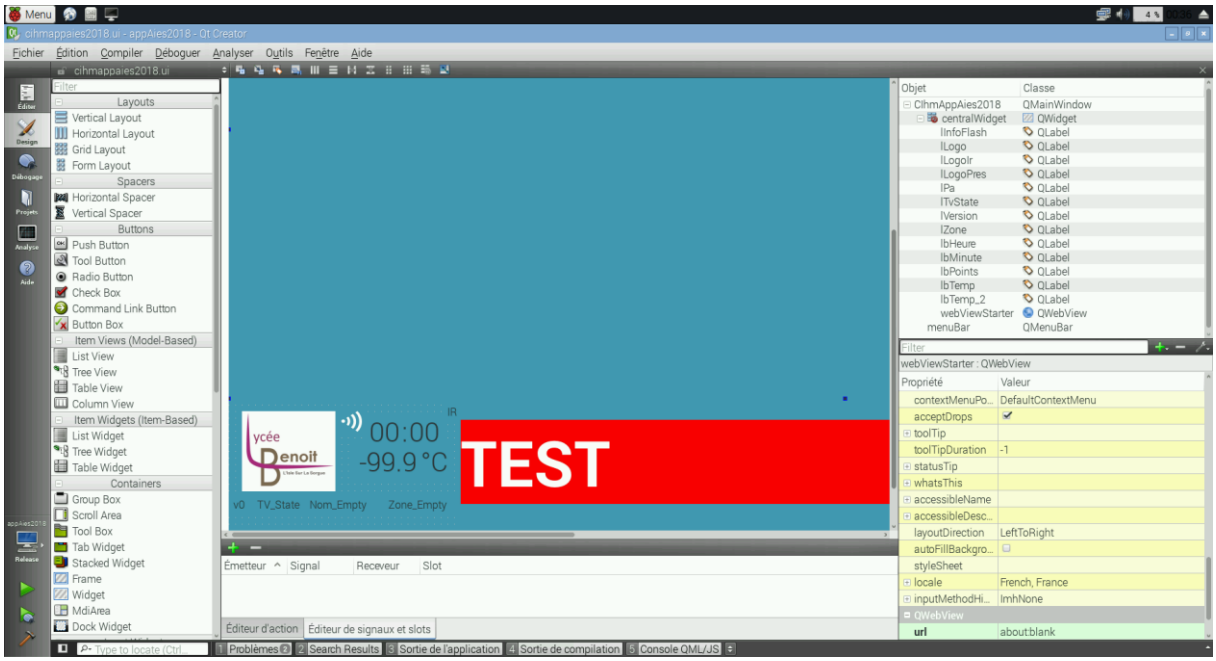
La classe **CCache,** permettant de gérer la mise en cache est en cours de création.

Deux classes concernent les capteurs on a le capteur de présence et le capteur de température qui sont **CCapteurPresence** et **CCapteurTemp**.

La classe **CEcran** concerne la partie écran du projet.

Quand a la classe **CLed et CGpio** on y retrouve les différents GPIO utilisés, les ports GPIO sont des ports d'entrée/sortie utilisés dans le monde des microcontrôleurs, en particulier dans le domaine de l'électronique embarquée.

8.8 Formulaire AppAIES2018



Il y a différents éléments qui sont visible ici, avec des **QLabels** et **QWebView**

Le **widget QLabel** fournit un affichage de texte ou d'image.

QLabel est utilisé pour afficher du texte ou une image. Aucune fonctionnalité d'interaction utilisateur n'est fournie. L'apparence visuelle de l'étiquette peut être configurée de différentes manières, et elle peut être utilisée pour spécifier une touche mnémonique de mise au point pour un autre widget.

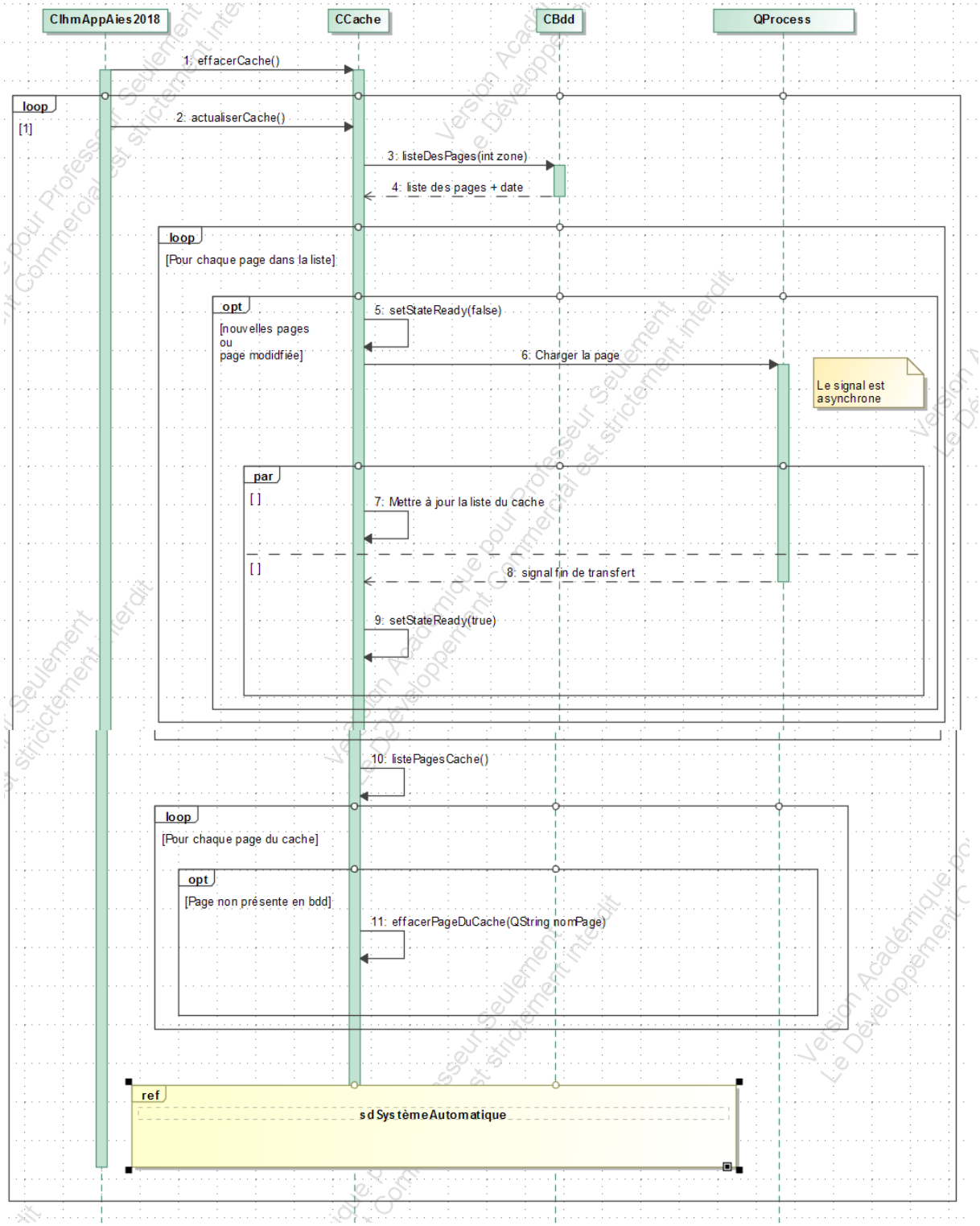
La classe **QWebView** fournit un **widget** utilisé pour afficher et modifier des documents Web.

QWebView est le composant de **widget** principal du module de navigation **Web QtWebKit**. Il peut être utilisé dans diverses applications pour afficher du contenu Web en direct sur Internet.

Un site Web peut être chargé sur **QWebView** avec la fonction load (). Comme tous les widgets **Qt**, la fonction show () doit être invoquée pour afficher **QWebView**.
On peut le voir ici:

```
QWebView *view = new QWebView(parent);
view->load(QUrl("http://qt-project.org"));
view->show();
```

8.9 Diagramme de séquence Mise en Cache



Ce diagramme de séquence montre le fonctionnement de la mise en cache.

Plusieurs classes sont concernés, **AppAies2018**, **CCache**, **CBdd** et **QProcess***

*utilisée pour démarrer des programmes externes et pour communiquer avec eux.

On actualise le cache pour voir si de nouvelles diapos sont disponibles,

Ensuite on liste les pages avec leur zone,

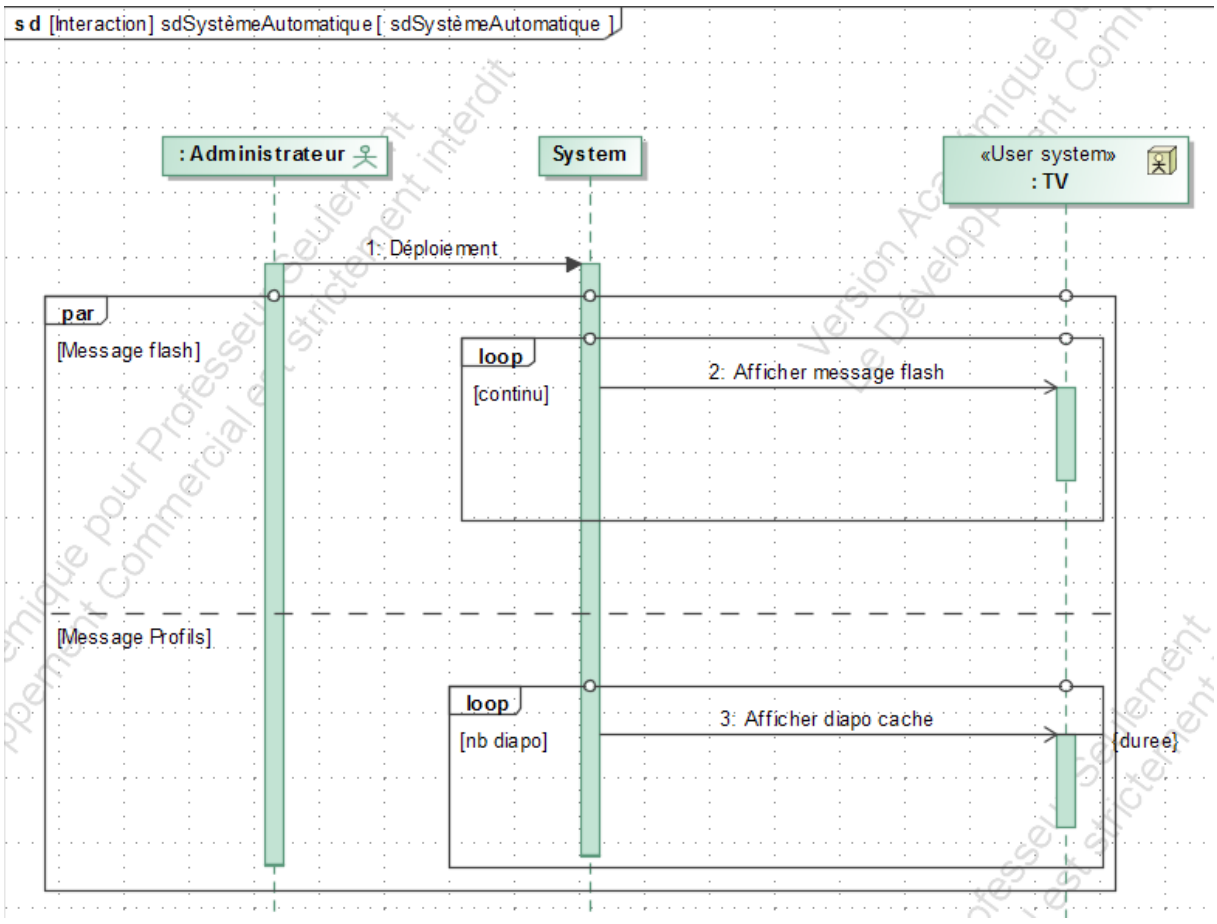
Pour chaque page dans la liste on les passe en état false, pour dire quelles ne sont prêtes,

La liste du cache est mise a jour et l'état de la diapo passe a true, puis nous est retourné.

Enfin pour chaque page du cache si la date est plus récente la diapo est effacée.

De plus ce signal est asynchrone car cela ne se passe pas à la même vitesse, que ce soit dans le temps ou dans la vitesse, donc ce n'est pas synchrone.

8.10 Diagramme de séquence Système Automatique



Dans le diagramme précédent, deux boucles, une pour les messages flash et une pour les Profils.

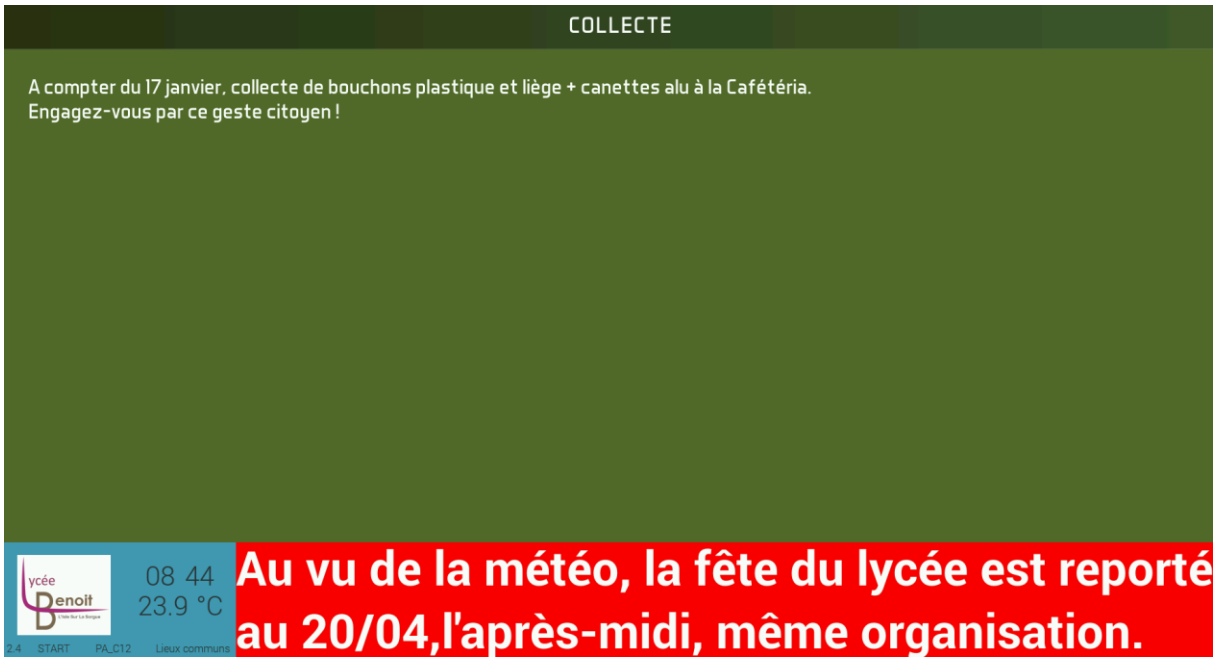
8.11 Bandeau Flash

Pour créer le bandeau flash rouge, j'ai ajouté une ligne a l'interface graphique dans le CIhmAppAies2018 :

```
188 void CIhmAppAies2018::onTimerFlash()
189 {
190     ui->lInfoFlash->setText("<html><style> h2 {background-color:red;} h2{color:white;}</style><h2>"+bdd->getActiveFlash()+"</h2></html>");
191 }
192
```

J'ai utilisé du code HTML, les éléments **<h1>** à **<h6>** représentent six niveaux de titres dans un document, <h1> est le plus important et <h6> est le moins important, dans notre cas h2 convient a l'utilisation du bandeau flash.

Le résultat est le suivant :



8.12 Problèmes et solutions retenues

Au début du projet, la version qui tournait sur les Raspberry pi des bâtiments, n'était pas trouvable dans la Raspberry de l'atelier, j'ai donc pris celle qui était le plus compatible.

Les classes existantes n'étaient pas bien organisées, alors j'en ai créé des nouvelles afin d'alléger le contenu.

QtCreator version 4.8 ne prenait pas en compte le serialport, j'ai donc décidé de migrer vers une version plus récente 5.3.

Pour installer la dernière version de Qt j'ai utilisé les commandes suivantes :

sudo apt-get install qt5default

sudo apt-get install qtcreator

sudo apt-get install libqt5serialport5

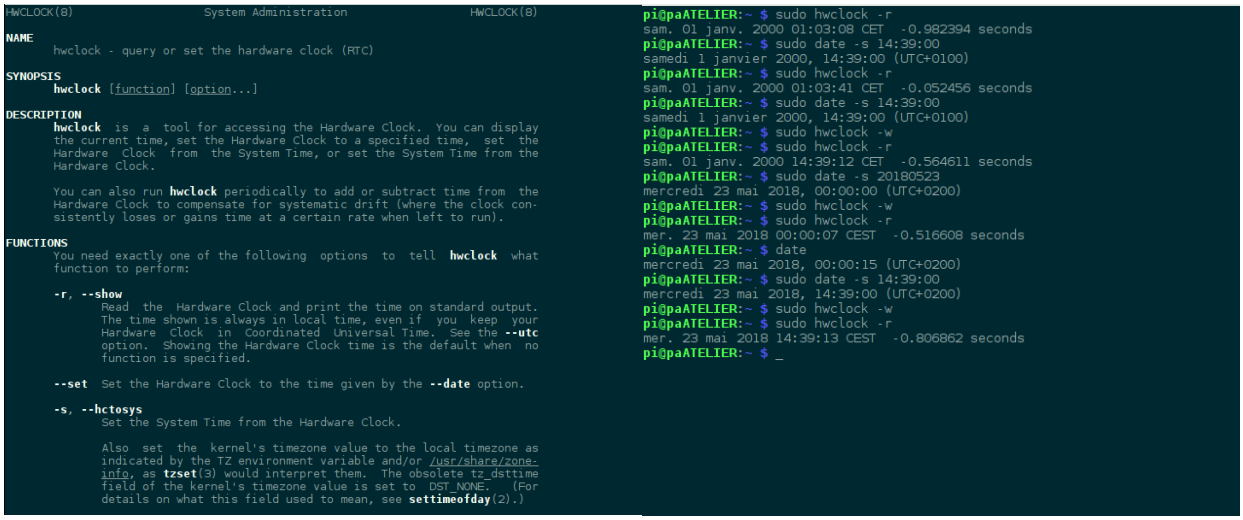
sudo apt-get install libqt5sql5

sudo apt-get install libqt5sql5-mysql

Un problème autre se situé dans le code, faisait planter le programme lors de son lancement, la source a donc était progressivement trouvée, il s'agissait d'une ligne qui ne fonctionnait pas et qui a donc était remplacée.

Le mode plein écran de l'application ne fonctionnait pas car il manquait une ligne dans le main.cpp : w.showFullScreen() ;

Un problème d'horloge était présent mais grâce au man de **hwclock** le problème a était réglé.



```
HWCLOCK(8)                  System Administration                  HWCLOCK(8)

NAME
       hwclock - query or set the hardware clock (RTC)

SYNOPSIS
       hwclock [function] [option...]

DESCRIPTION
       hwclock  is  a  tool for accessing the Hardware Clock.  You can display
       the current time, set the Hardware Clock to a specified time,  set  the
       Hardware  Clock  from  the System Time, or set the System Time from the
       Hardware Clock.

       You can also run hwclock periodically to add or subtract time from  the
       Hardware Clock to compensate for systematic drift (where the clock con-
       sistently loses or gains time at a certain rate when left to run).

FUNCTIONS
       You need exactly one of the following  options  to  tell  hwclock  what
       function to perform:

       -r, --show
              Read  the  Hardware Clock and print the time on standard output.
              The time shown is always in local time, even if  you  keep  your
              Hardware  Clock  in  Coordinated  Universal Time.  See the --utc
              option.  Showing the Hardware Clock time is the default when  no
              function is specified.

       --set  Set the Hardware Clock to the time given by the --date option.

       -s, --hctosys
              Set the System Time from the Hardware Clock.

              Also  set  the  kernel's timezone value to the local timezone as
              indicated by the TZ environment variable and/or /usr/share/zone-
              info, as tzset(3) would interpret them.  The obsolete tz_dsttime
              field of the kernel's timezone value is set to  DST_NONE.   (For
              details on what this field used to mean, see settimeofday(2).)
```

```
pi@paATELIER:~ $ sudo hwclock -r
sam. 01 janv. 2000 01:03:08 CET  -0.982394 seconds
pi@paATELIER:~ $ sudo date -s 14:39:00
samedi 1 janvier 2000, 14:39:00 (UTC+0100)
pi@paATELIER:~ $ sudo hwclock -r
sam. 01 janv. 2000 01:03:41 CET  -0.052456 seconds
pi@paATELIER:~ $ sudo date -s 14:39:00
samedi 1 janvier 2000, 14:39:00 (UTC+0100)
pi@paATELIER:~ $ sudo hwclock -w
pi@paATELIER:~ $ sudo hwclock -r
sam. 01 janv. 2000 14:39:12 CET  -0.564611 seconds
pi@paATELIER:~ $ sudo date -s 20180523
mercredi 23 mai 2018, 00:00:00 (UTC+0200)
pi@paATELIER:~ $ sudo hwclock -w
pi@paATELIER:~ $ sudo hwclock -r
mer. 23 mai 2018 00:00:07 CEST  -0.516608 seconds
pi@paATELIER:~ $ date
mercredi 23 mai 2018, 00:00:15 (UTC+0200)
pi@paATELIER:~ $ sudo date -s 14:39:00
mercredi 23 mai 2018, 14:39:00 (UTC+0200)
pi@paATELIER:~ $ sudo hwclock -w
pi@paATELIER:~ $ sudo hwclock -r
mer. 23 mai 2018 14:39:13 CEST  -0.806862 seconds
pi@paATELIER:~ $ _
```

8.13 Evolution au cour du projet

Pendant la durée du projet il a fallu trouver des solutions aux problématiques qui se présentés, certaines se sont révélées plus simple ou plus compliqué que prévu.

8.14 Conclusion

Dans la première partie du projet, mon but était surtout de créer la nouvelle version de l'application, de la rendre fonctionnelle.

Puis dans la seconde de créer les nouvelles classes pour nos besoins tels que le cache.

Et enfin dans la dernière, la fonction getU qui va nous permettre de récupérer l'information de la tension sur un port USB de l'écran, pour nous permettre d'éteindre l'écran. Finir aussi la gestion du cache pour le format vidéo car ce dernier est le plus gourmand en espace de stockage.

9. HUET Valentin : Technicien EC

9.1 Introduction :

Le projet AIES (Affichage Information Etablissement Scolaire) est un projet lancé par le lycée qui va servir à mettre toute sorte d'information a la disposition du personnel et des élèves d'un établissement scolaire comme par exemple :

- Afficher les professeurs absents de la journée ou de la semaine
- Afficher des informations importantes

Je m'occupais de tout ce qui est partie EC, comme gérer l'émetteur IR, le détecteur de présence, le capteur de température, l'horloge en temps réelle, l'installation de produit des années précédentes, leur amélioration et leur développement.

Je travaillais avec 2 Techniciens IR qui ont contribués a leurs partie sur le projet, et avec qui j'ai pu partager quelques idées.

Ce compte rendu prend compte du travail effectué durant la période du 10/01/2018 au 25/05/2018.

9.2 Travail effectué :

- Réalisation d'un Gantt
- Saisie de schéma sur ISIS et ARES
- Commande de la carte et son pochoir
- Duplication d'une carte AIES2017
- Défaut horloge temps réelle (RTC)
- Explication de la mémorisation et restitution du signal IR

9.3 Réalisation d'un Gantt :

Le Gantt fut réaliser en deux temps, d'abord de manière séparer j'ai réalisé un Gantt général pour mes activités. Voici ce Gantt :

Mode Tâche	Nom de la tâche	Durée	Début	Fin
	Projet 2018 AIES	34,86 jours	Mer 10/01/1	Ven 25/05/1
	Lecture de document et mise en place des logiciels	4 jours	Mer 10/01/18	Ven 19/01/18
	implantation des capteur IR et capteur de mouvement	4 jours	Mer 24/01/18	Ven 02/02/18
	Création de schéma Isis et Ares	4 jours	Mer 07/02/18	Ven 16/02/18
	Test des différents schémas .	2 jours	Mer 14/03/18	Ven 16/03/18
	Mesure d'intensite des différente tv utiliser	2 jours	Mer 21/03/18	Ven 23/03/18
	Recherche de matériel.	2 jours	Mer 28/03/18	Ven 30/03/18
	Commande du matériel	1 jour	Mer 04/04/18	Mer 04/04/18
	Vérification du bon fonctionnement du matériel.	3 jours	Jeu 05/04/18	Ven 13/04/18
	Devellopement application en mode console	7 jours	Mer 18/04/18	Ven 11/05/18
	Evolution vers une version graphique de l'application	3 jours	Jeu 17/05/18	Ven 25/05/18

Dans un second temps il fut repris par les IR afin d'être ajouter et de posséder un Gantt complet.

9.4 Saisie de schéma sur ISIS et ARES :

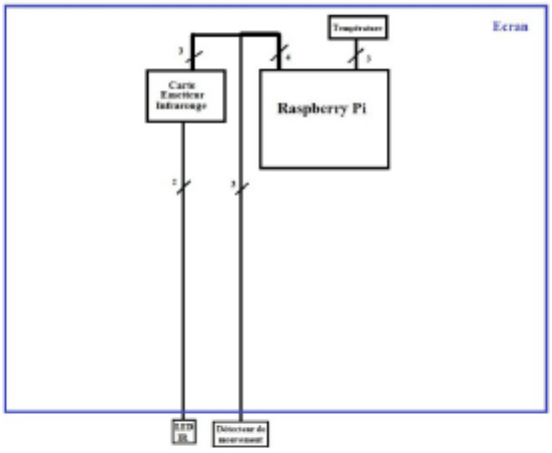
9.4.1 Fiche de demande de modification :

2. Fiche de demande de modification.		*Réf.* FDM-AIES-01
Projet	*Demandeur*	*Date*
AIES	Lycée Benoit	10/01/2018

Description de la modification

Modifications apportées au cahier des charges suite aux demandes de M. Claveyrolle *(magasinier du LAB et installateur du système)* **et de M. Antoine**

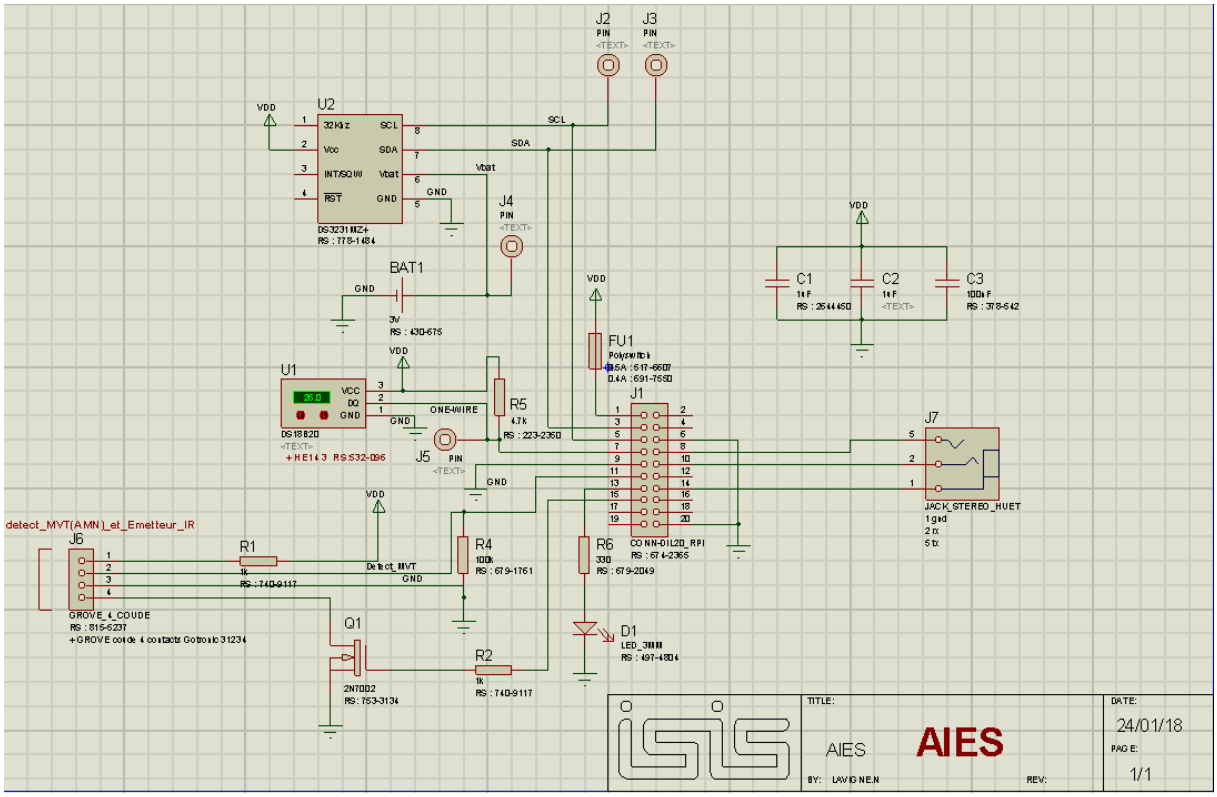
- Afin de fixer plus facilement sur l'écran la LED IR permettant le basculement en mode marche/arrêt du téléviseur, cette LED sera dissociée de la carte qui la pilote (liaison filaire), et cette carte sera fixée à proximité de la carte Rpi lors du déploiement du système. Dans la version 2018 les composants de cette petite carte devront être intégrés sur le shield de la carte Rpi.



- Afin de vérifier la bonne prise en compte de la commande marche/arrêt générée par la LED IR, il est souhaitable qu'un dispositif électronique permettant un retour sur l'état du téléviseur soit mis en œuvre. Une possibilité est d'effectuer une mesure du courant d'alimentation du téléviseur, pour distinguer les 2 états de l'appareil. D'autres solutions peuvent être envisagées, à l'initiative des candidats du projet. Seule une solution viable (donc testée en situation) et ne majorant pas excessivement le coût du dispositif, sera retenue. Dans tous les cas les essais devront figurer dans le dossier. Si une solution est retenue, la structure correspondante devra être ajoutée au shield de la carte Rpi.

9.4.2 Choix d’adaptation du signal RS232

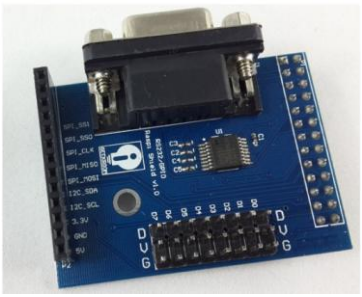
Dans un premier temps, j'ai pensez simplement réaliser le schéma de la carte en fessant les modifications demander et en ajoutent l'embase pour la nouvelle télévision.



Cependant, j'ai réalisé le besoin d'adapter le signal reçus de la prise jack qui est en +13, –13V est un signal RS232 tandis que la carte Rpi fonctionnent en +3,5V, 0V, un signal TTL.

Après quelque recherche j'ai trouvé plusieurs solutions.

La première l'ajout d'une carte qui convertie le signal reçus RS232 en Signal TTL communication série.



Cette solution possède c'est avantage, achat direct, pas besoin de prendre de risque quand au fonctionnement, connexion simple et direct.

Cependant elle possède des inconvénients aussi, ajout d'un certain poids et plutôt encombrant.

La carte déjà bien pleine cette solution ne fût pas retenue finalement.

La seconde solution fût un Adaptateur TTL / RS232.

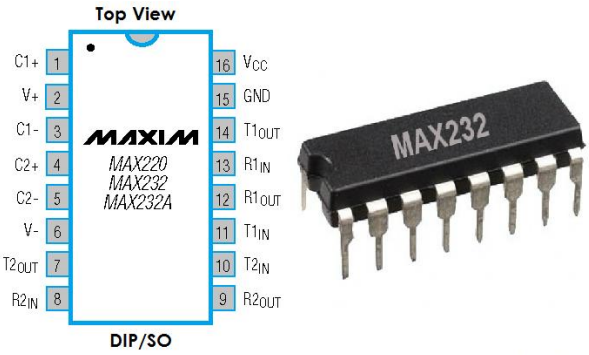


C'est avantage plutôt simple, peut de branchement très facile d'utilisation et sans besoin de modification de la part de l'installateur.

Cependant, le coût élever et le besoin d'adaptions de la carte malgré sont utilisation l'on rendu presque inutile dans notre situation.

La seconde solution fût également rejeter.

La troisième solution fût l'ajout d'un MAX232.



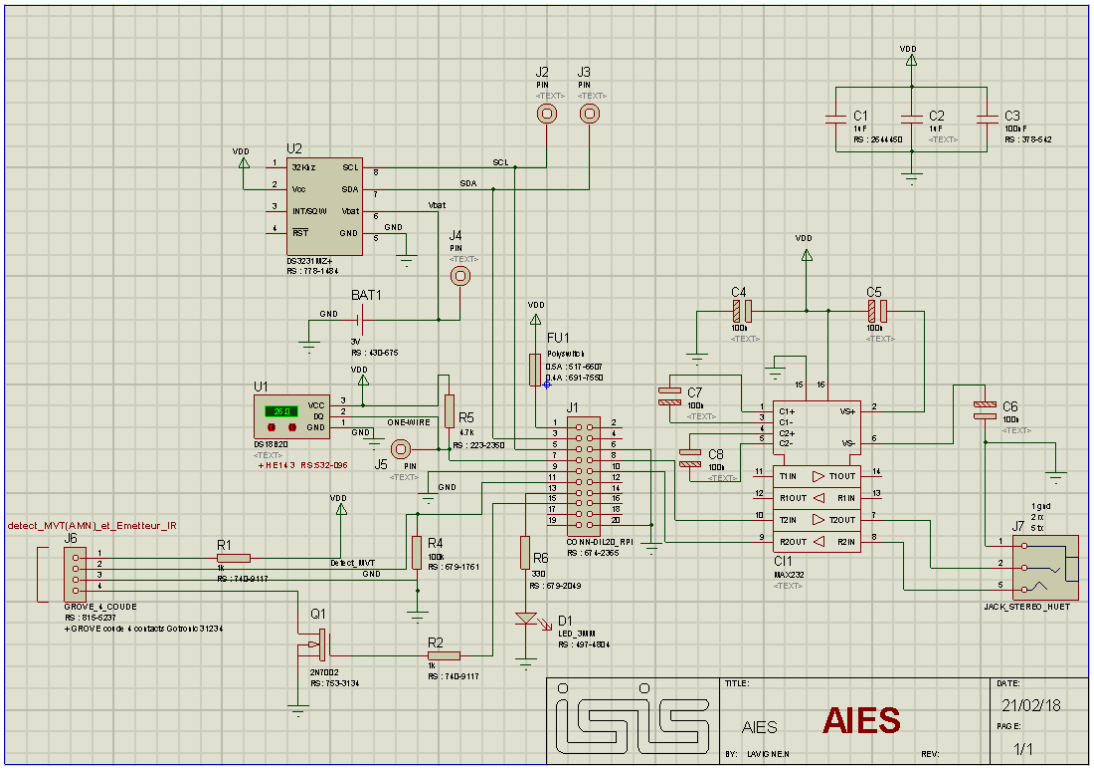
MAX232: Construction, Operation, Types and Application

Le MAX232 convertie un signal RS232 reçus en signal TTL, c'est avantage son sont faible coût, la possibilité de l'ajouter directement sur une carte déjà existante, il prend peut de place.

C'est inconvénient son le besoin d'ajouter un certain nombre de transistors pour son bon fonctionnement ainsi que le besoin de connaître le matériel sur lequel on travail.

La troisième solution fût celle choisie.

J'ai joins le dernier schéma réaliser avant la première revue de projet ci-dessous :



9.4.3 Modification apporter a la suite de la première revue de projet :

Après la première revue de projet plusieurs modification mon été demander ils sont listé ci-dessous :

-Ajout d'un port mini-USB servant à la vérification du mode veille des écrans.

- L'ajout d'un pont de résistance en sortie du max232 broche 9.

- Le déplacement des condensateurs afin qu'ils soient placé sur le schéma.

- La vérification des empreintes et des valeurs des composants.

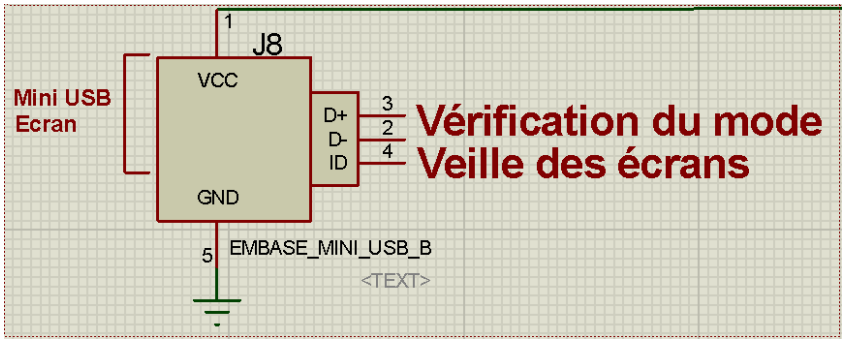
- Changement du texte du cartouche du schéma.

- Ajout d'un diagramme de bloc.

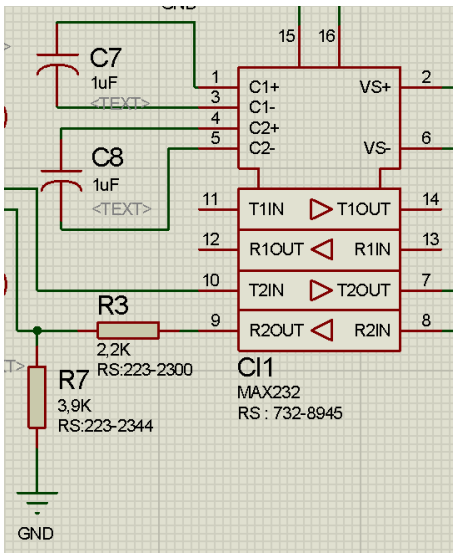
-Ajout de 3 points test (Gnd, Tx, Rx)

J’ai réalisé chacune des modifications notées comme indiquer ci-dessous :

Le port mini-USB servant a la vérification du mode veille des écrans.



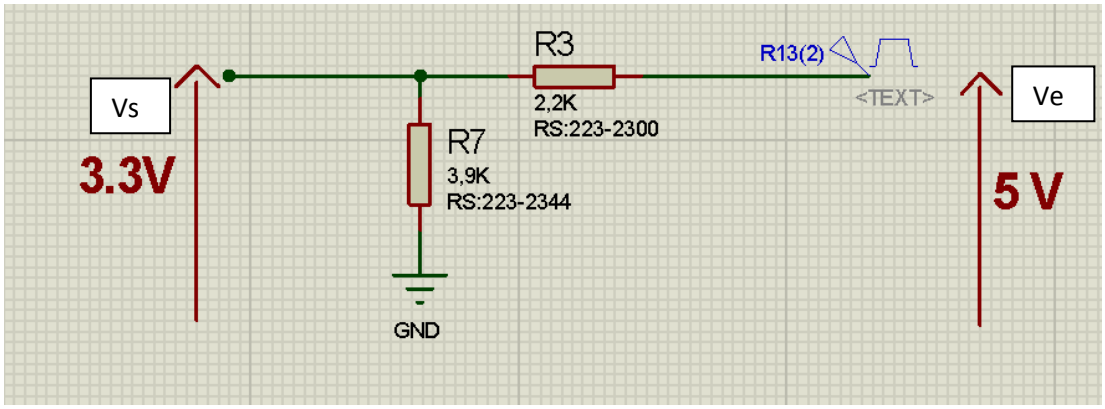
L'ajout d'un pont de résistance en sortie de max232 broche 9.



L'entré UART utilise des impulsions allant de 0 à 5V tandis que l'entrée GPIO de la carte RPI fonctionne en 0 à 3,3V.

Afin de protéger le circuit on met en place un pont de résistance afin d'abaisser la tension reçue (5V) en tension utilisable (3,3V).

Les résistances on les valeurs suivante : R3 (2,2KΩ) et R7 (3,9KΩ).



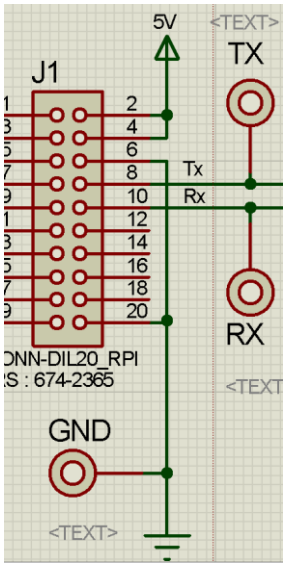
Afin de vérifier les valeurs des résistances j'ai appliqué la loi des ponts diviseurs de tensions :

$$VS = VE \times \left(\frac{R7}{(R3 + R7)}\right)$$

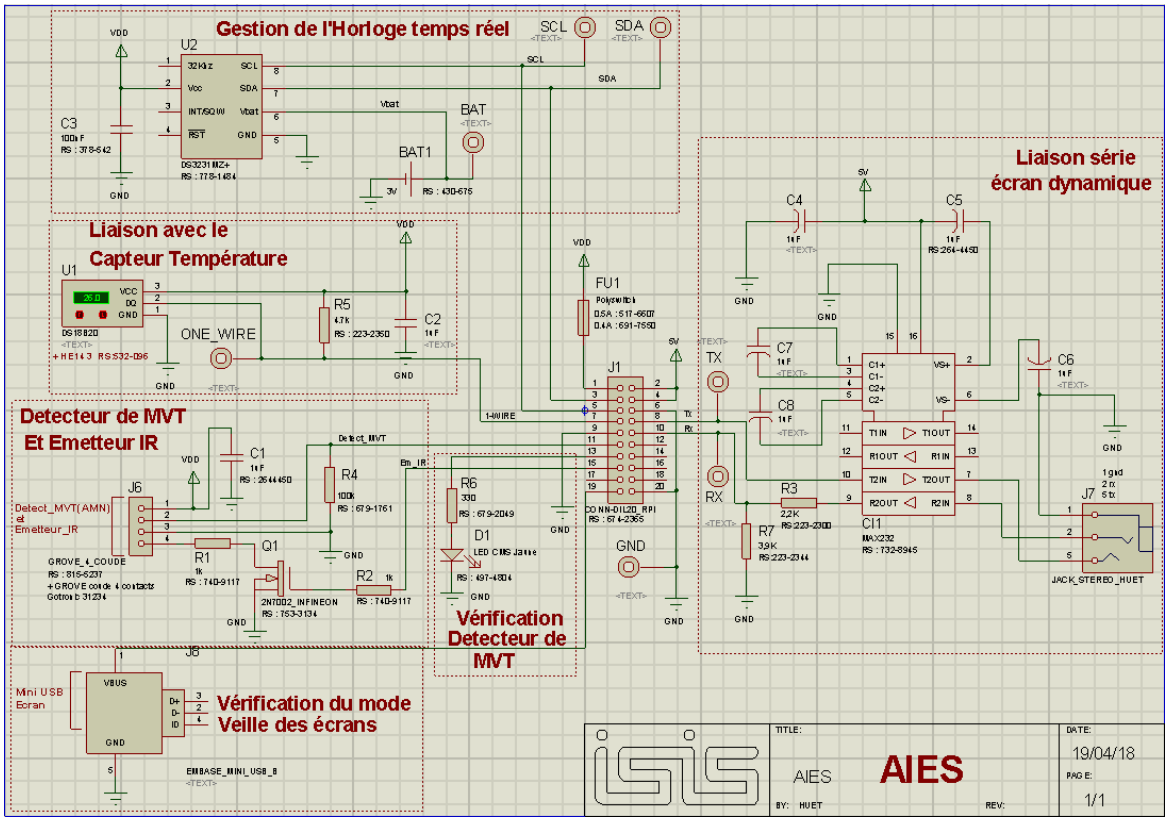
$$VS = 5 \times \left(\frac{3.9}{(2.2 + 3.9)}\right)$$

$$VS = 5 \times \left(\frac{3.9}{6.1}\right) \qquad VS = 3.2V$$

Ajout de 3 points test (Gnd, Tx, Rx).



Voici le schéma lors de son envoie commande.



Un diagramme de bloc fût ajouté également. Pour faciliter la lecture du schéma.

9.4.4 Création du schéma de routage :

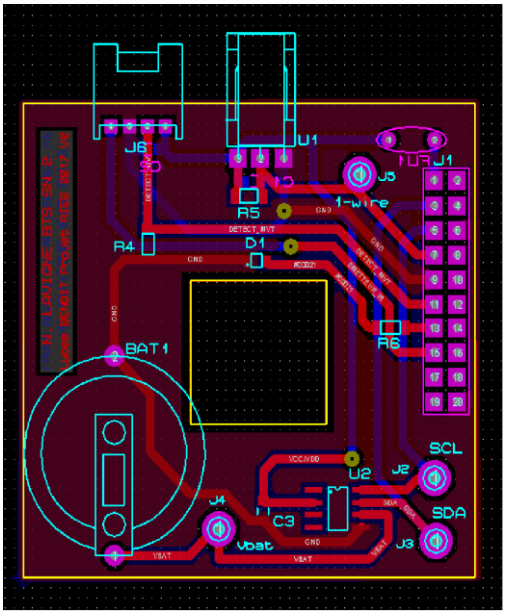
A la suite de la validation du schéma, le schéma de routage a été réalisé. Certaines conditions notées ci-dessous devaient être remplies.

- Le composant J1 ne devait pas changer de position.

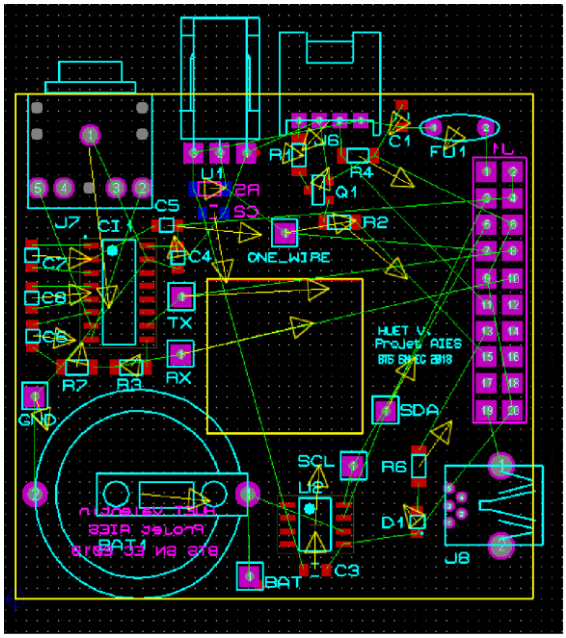
- Les connecteurs U1 et J6 devaient rester sur le même coté de la carte que sur les versions antérieurs.

- Les condensateurs de découplage devaient être proches de leur composant.

Voici la version antérieure du schéma de routage de la carte :

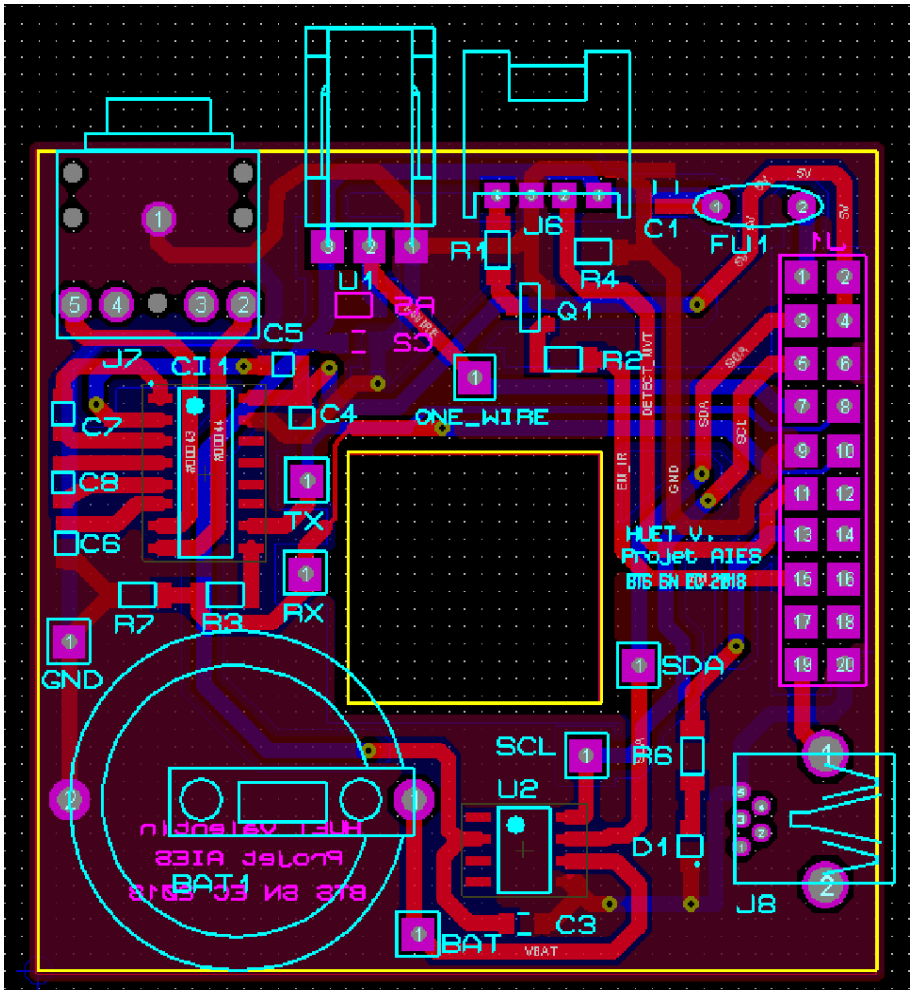


En respectant les consignes les composants ont été placés de la façon suivante :



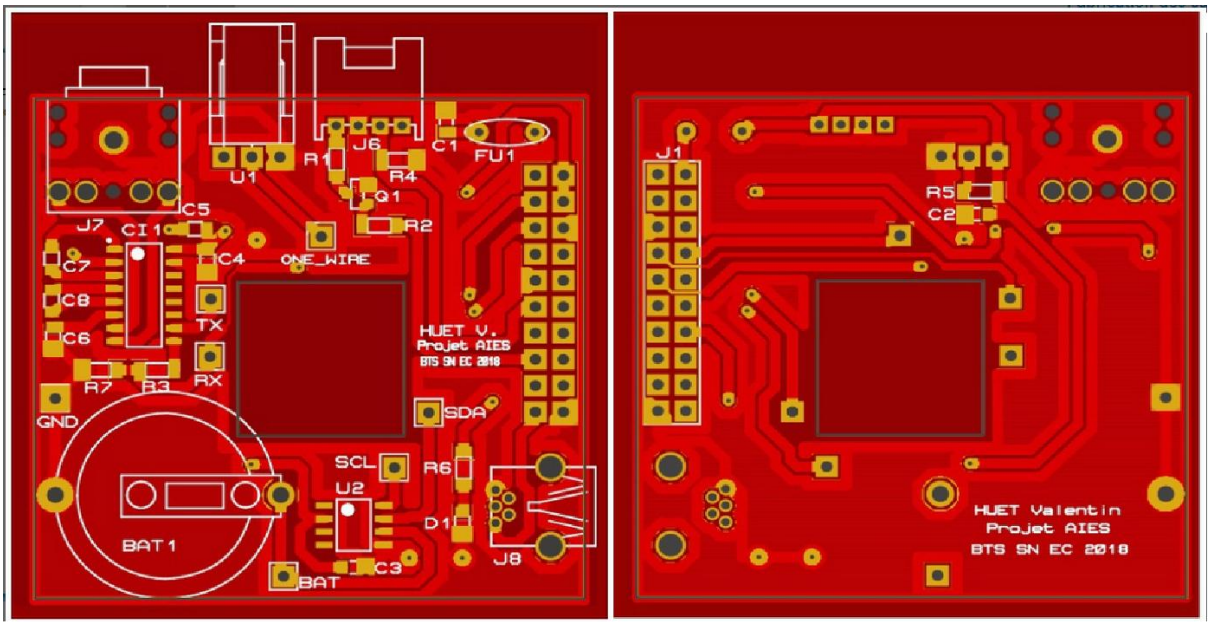
On voit que les consignes ont bien été respectés J1 n'a pas été déplacé, les composants J6, U1 sont bien sur le même coté que sur la version antérieure et les condensateurs sont proches de leur composant.

Ensuite les pistes on été tracées cependant il faut faire des pistes d'angles 45° on a utilisé des pistes de taille T40 sauf pour les interconnexions de J1 ou on a utilisé du T8.



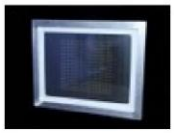
9.5 Commande de la carte et son pochoir

La création du schéma électrique et de routage terminés le schéma de routage fût envoyé sur seedstudio afin d'indiquer l'apparence de la carte à sa réception avant de passer commande. Voici le fichier converti :



Puis en attente de la commande, il est possible de vérifier les informations et le statut de la commande comme prouvé ci-dessous :

Information du pochoir :



PCB Stencil Ser Status: Traceable

SKU: 503000001

$8.90 × 1

PCB Dimensions - 10.0cm * 15.0cm Frameless - eff.8*13
Fiducial Mark - No Fiducial
Thickness - 0.12
Quantity - 1
Polished - Polished
File - D_798984_aies_huet_2018_04_19cadcam.ZIP

Status: Traceable

Traceable	2018-05-07 09:02:48
Stencil Awaiting Packing	2018-04-28 05:31:55
Stencil Processing	2018-04-21 11:21:39

Information de la carte :



Fusion PCB **Status: Traceable**

SKU: 501010001

$4.90 × 1

PCB Dimensions - 49.6mm*51mm
Impedance Control - No
Surface Finish - HASL
Minimum Solder Mask Dam - 0.4mm
Copper Weight - 1oz.
Plated Half-holes / Castellated Holes - No
Minimum Drill Hole Size - 0.3mm
PCB Color - Red
Base Material - FR-4 TG130
No. of Layers - 2 layers
PCB Quantity - 10
Blind or Buried Vias - No
PCB Thickness - 1.6
Trace Width / Spacing - 6/6 mil
No. of Different Designs - 1
Expedited Option - No
File - D_798983_aies_huet_2018_04_19cadcam.ZIP

Status: Traceable

Traceable	2018-05-07 09:02:48
PCB Awaiting Packing	2018-05-02 04:59:07
PCB In Production	2018-04-21 20:13:35
PCB Processing	2018-04-21 11:21:34

Ainsi qu'un suivit de commande détaillé tel que :

Shipping Method: **DHL** Tracking Number: **9077652975**

2018-05-09 07:36:00	Departed Facility in MARSEILLE - FRANCE
2018-05-09 07:21:00	Processed at MARSEILLE - FRANCE
2018-05-09 07:17:00	Clearance processing complete at MARSEILLE - FRANCE
2018-05-09 06:58:00	Arrived at Sort Facility MARSEILLE - FRANCE
2018-05-09 04:26:00	[MARSEILLE - FRANCE]Customs status updated
2018-05-09 04:22:00	Departed Facility in LEIPZIG - GERMANY
2018-05-08 21:50:00	Processed at LEIPZIG - GERMANY

Voici la facture de la commande :

Item		Qty	Unit Price
PCB Stencil Service	$8.90		
Fusion PCB	$4.90		
Fusion PCB	$4.90		
Fusion PCB	$4.90		
Fusion PCB	$4.90		
PCB Stencil Service	$8.90		
Fusion PCB	$4.90		
PCB Stencil Service	$8.90		
PCB Stencil Service	$8.90		
PCB Stencil Service	$8.90		
Fusion PCB	$9.90		
Fusion PCB	$9.90		
Fusion PCB	$9.90		
	Sub Total:		$98.70
	Shipping:		$55.97
	Discount:		-$0.00
	Grand Total:		$154.67

9.6 Câblage d'une carte AIES2017

Une carte AIES2017 à été câblé les composants utilisés sont listés dans la nomenclature ci-dessous :

3 Resistors

Quantity:	References	Value	Order Code
1	R4	100k	
1	R5	4.7k	
1	R6	330	

3 Capacitors

Quantity:	References	Value	Order Code
2	C1, C2	1uF	
1	C3	100nF	

2 Integrated Circuits

Quantity:	References	Value	Order Code
1	U1	DS18B20	
1	U2	DS3231MZ+	

1 Diodes

Quantity:	References	Value	Order Code
1	D1	LED_3MM	

8 Miscellaneous

Quantity:	References	Value	Order Code
1	BAT1	3V	
1	FU1	Polyswitch	
1	J1	CONN-DIL20_RPI	
4	J2-J5	PIN	
1	J6	GROVE_4_COUDE	

Dans un premier temps les composants CMS furent soudés sur la carte dans l'ordre suivant :

R4, R5, R6

D1

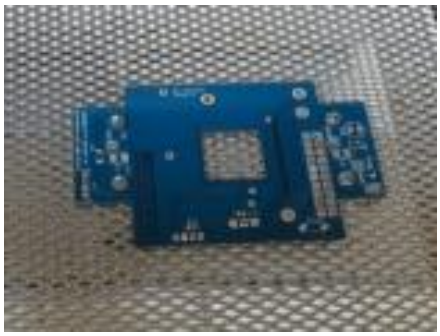
C3

U2

Tous ces composants sont présents sur la même face de la carte après leur cuisson, j'ai ajouté à l'autre face de la carte les composants CMS dans l'ordre suivant :

C1, C2

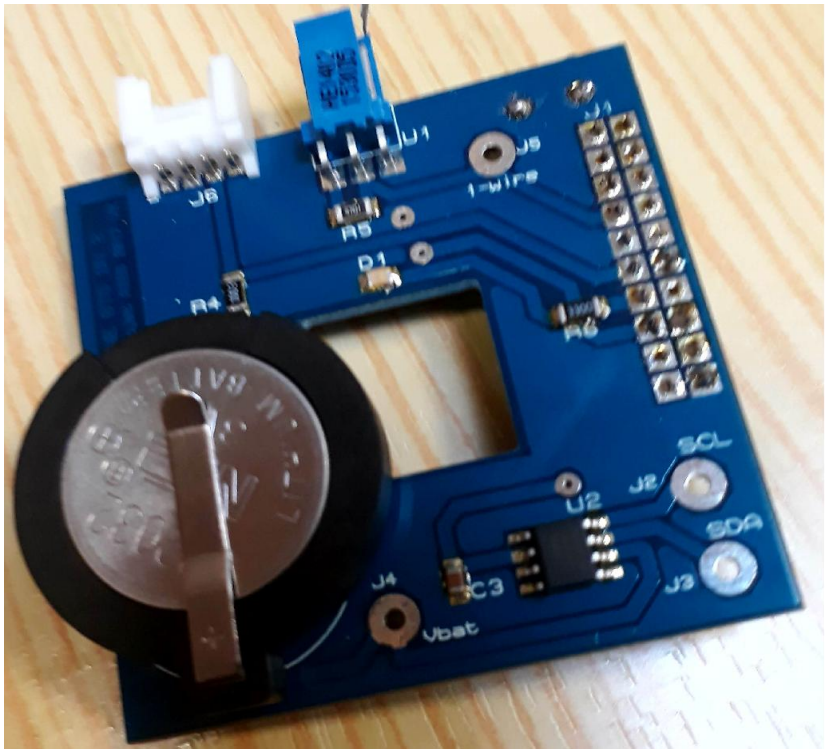
Il a fallut installer deux cale avant la cuisson, cela permet d'éviter également de toucher le support métallique chaud et permet d'éviter que la carte ne sois disposée de travers comme indiquer sur l'image ci-dessous:



Enfin ajout des composants traversant sur carte dans l'ordre suivant:

U1, J6, J1, FU1, Bat1

Comme indiqué sur l'image ci-dessous :



9.7 Défaut sur l'horloge concernant l'heure

L'Horloge DS3231MZ+ avait un défaut en effet l'heure était affichée cependant après avoir déconnecté la carte du réseau pendant un certain temps, l'heure était remise à 0H00.

Afin de résoudre le problème la procédure d'installation a été réalisée a nouveau. La procédure suivie est indiquée ci-dessous:

- On passe la Rpi en mode Super User :

pi@raspberrypi:~$ sudo su

- On modifie le fichier /etc/module avec l'éditeur nano :

root@raspberrypi:/home/pi# nano /etc/module

- On ajoute les lignes :

i2c-bcm2708

i2c-dev

- Quittez puis redémarrez la carte :

root@raspberrypi:/home/pi# reboot

- Afin de vérifier le bon fonctionnement utiliser les 2 commandes suivantes :

root@raspberrypi:/home/pi# dmesg|grep i2c

root@raspberrypi:/home/pi# ls /dev/i2c

- Pour s'assurer que l'on détecte bien la connexion entre le busI2C et le montage on utilise la commande :

root@raspberrypi:/home/pi# apt-get install i2c-tools

- On vérifie l'adresse avec

root@raspberrypi:/home/pi# i2cdetect –y 1

Le résultat doit ressembler à :

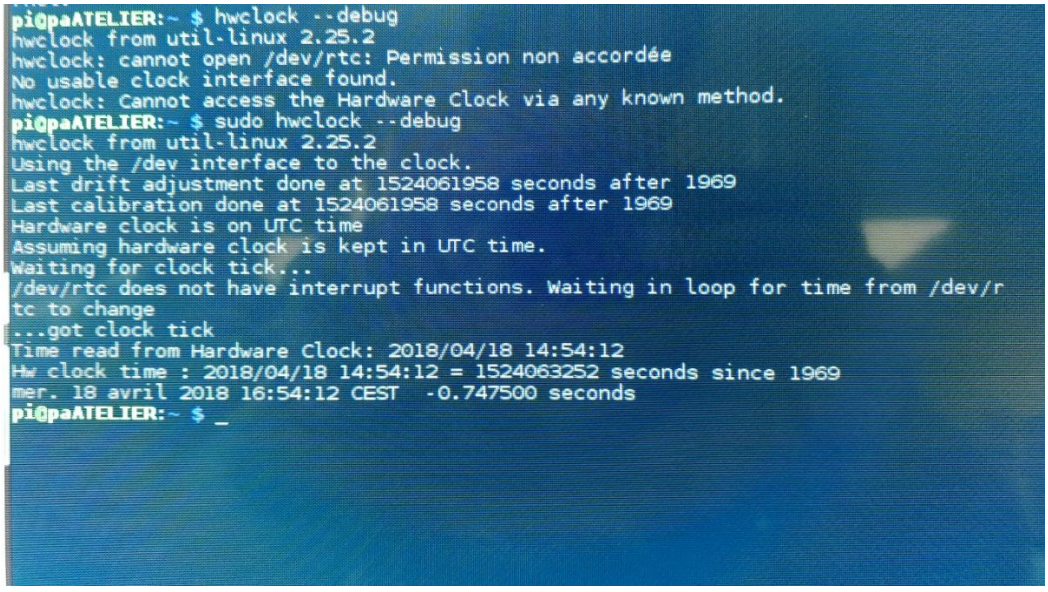
```
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- 68 -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
```

Cependant on obtient UU qui signifie que le bus est en cours d'utilisation.

Le problème n'étant toujours pas résolu des tests sur la pile afin de vérifier l'intensitée quelle fournit est suffisante, après une mesure de tension avec un multimètre, la tension est bien de 3,5volts.

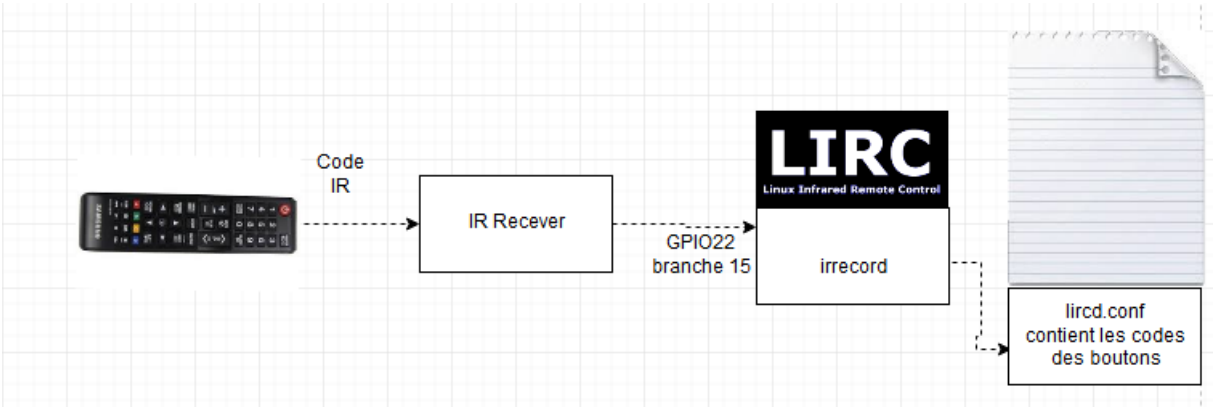
La raison du défaut était dût a un enregistrement de l'heure mal réaliser en effet l'heure après avoir été entré restait influençable par l'heure internet. Il faut rajouter une commande "sudo hwclock -u " afin de repasser l'utc en local time afin de remédier au problème.

A l'aide du terminal nous avons résolu l'erreur puis nous nous sommes assurés du bon fonctionnement de la RTC et de la récupération correct de l'heure en utc.



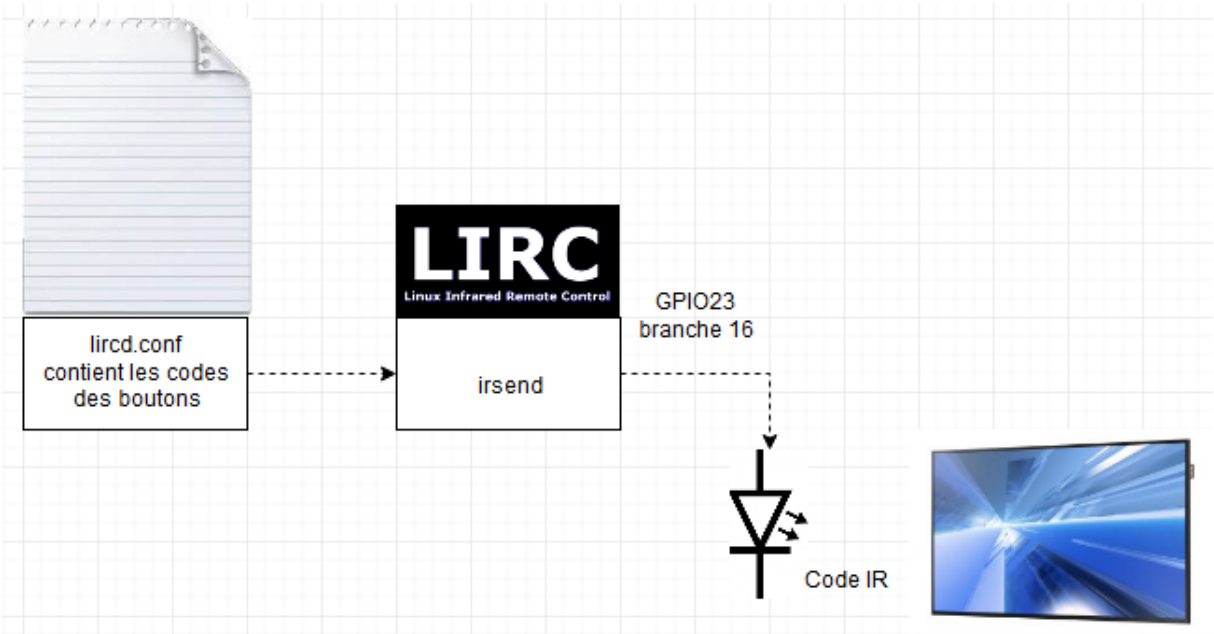
9.8 Explication de la mémorisation et restitution du signal IR

9.8.1 Schéma explicatif de la mémorisation du signal IR



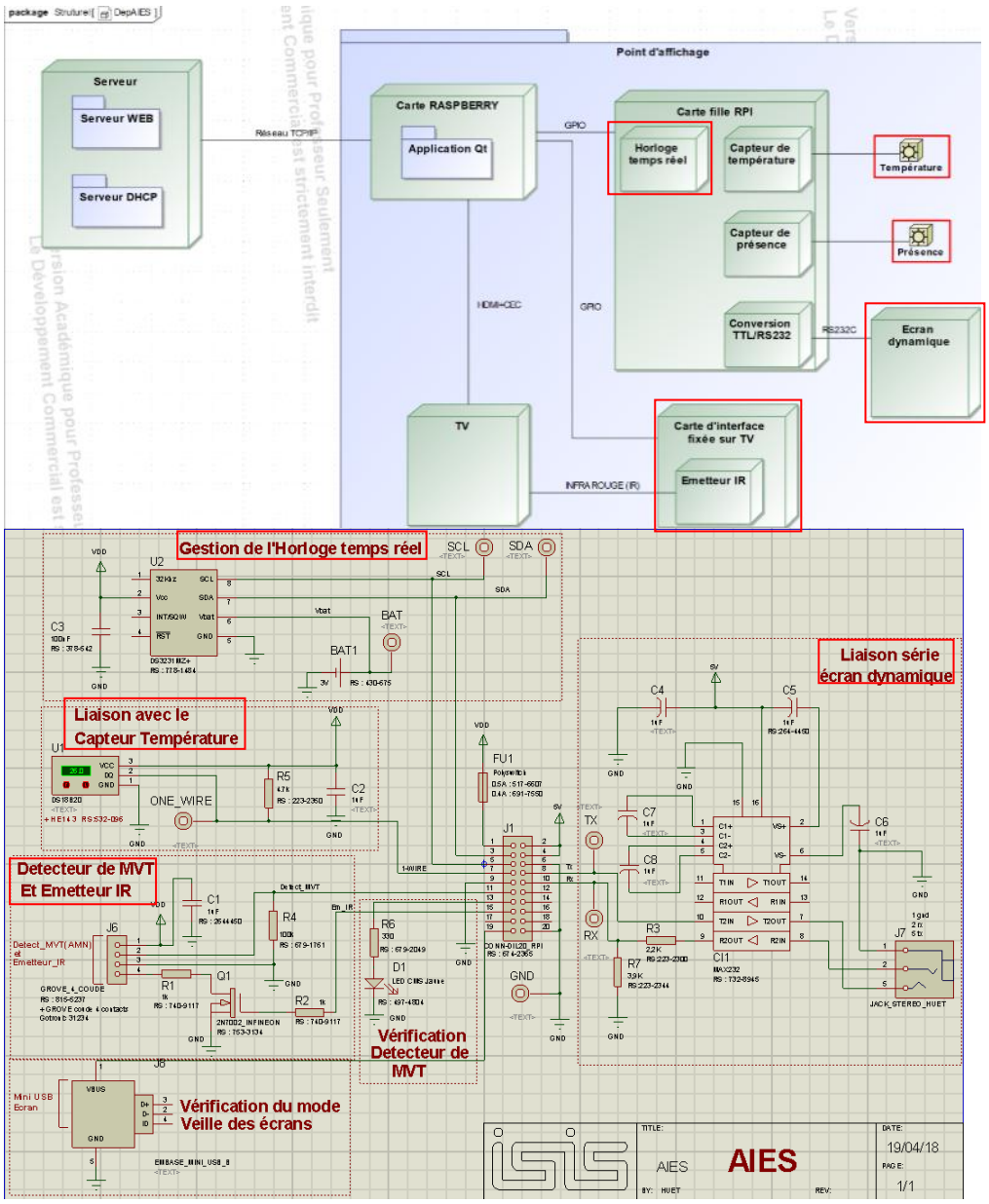
On lance le logiciel LIRC avec la commande « `irrecord -d /dev/lirc0 ~/lircd.conf` » Après la page de présentation du programme, l'explication de l'enregistrement de la télécommande a lieu. Il faut appuyer sur plusieurs boutons de la télécommande pendant 1s environs en direction du récepteur. Après il faut nommer la commande d'après la liste que l'on a dans « `irrecord --list-namespace` ».Il faut finalement maintenir la touche que l'on souhaite enregistré.

9.8.2 Schéma explicatif de la restitution du signal IR



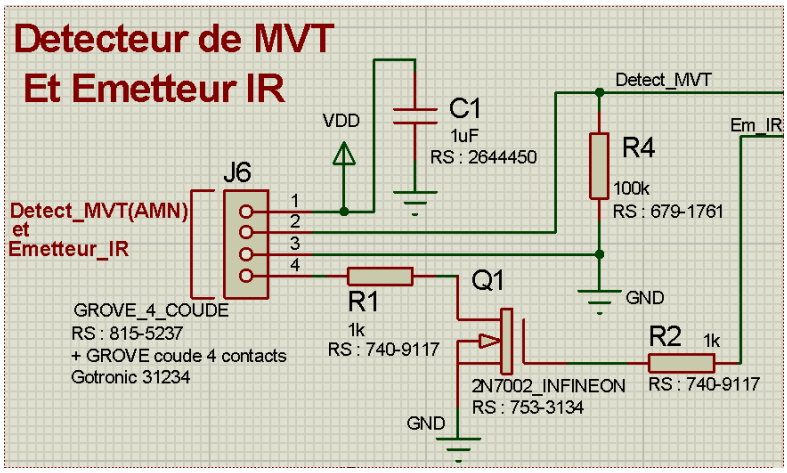
Il suffit d'envoyer la commande enregistré en utilisant « `irsend` » suivit du nom enregistré pour la commande.

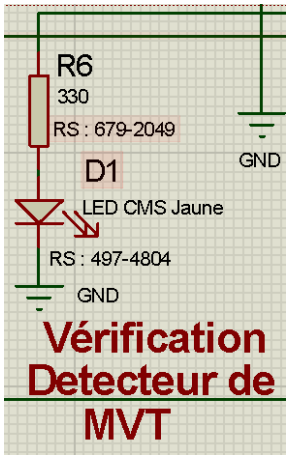
9.9 Parallèle entre le diagramme de déploiement et le schéma structurel :



9.10 Partie Physique

9.10.1 Détecteur de mouvement :

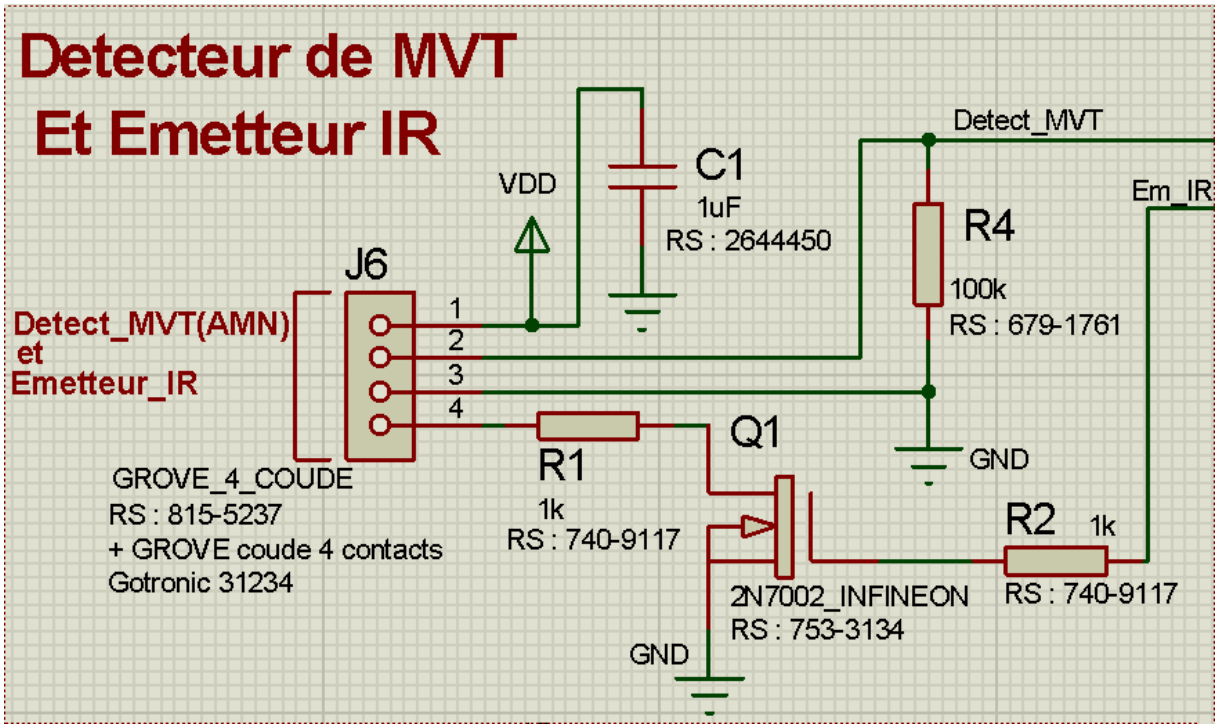




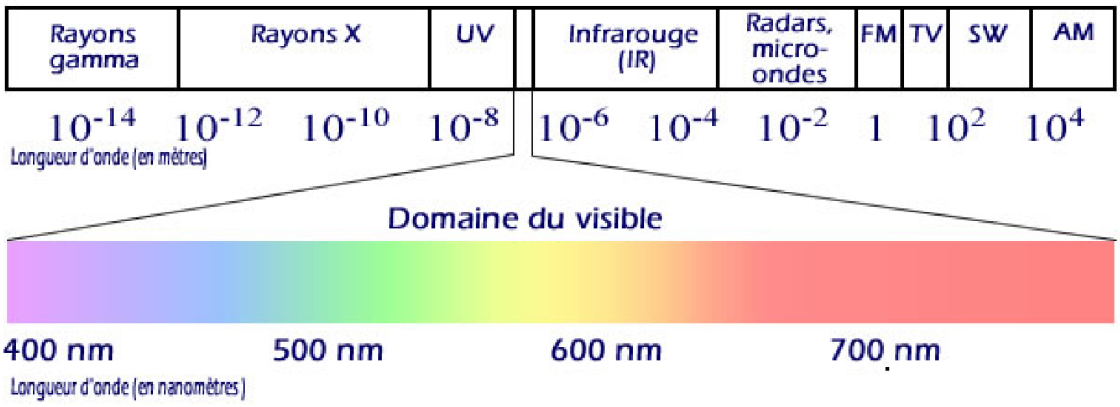
On utilise un détecteur de mouvement qui fonctionne grâce au changement de température car un corps rayonne une onde électromagnétique en fonction de sa température. L'onde électromagnétique du corps humain est ici dans le domaine des infrarouges ainsi les changements de température émise lors de l'intrusion d'une personne dans sa zone de détection entraine un changement des ondes infrarouge et font réagir le détecteur de mouvement qui envoie une impulsion.



9.10.2 Emetteur Infrarouge :



Infrarouge : L'infrarouge son des ondes d'une certaine longueur d'onde, elles sont de l'ordre des ondes invisibles et la température est émise en infrarouge. Ce sont les ondes d'une longueur comprise entre 1µm et 100µm, la fréquence est comprise entre f (max) = $3*10^{17}$ HZ et f (min) = $4,3*10^{14}$ Hz .Elle trouve leur nom du latin qui signifie "en dessous du rouge ". L'Emetteur Infrarouge sert à reproduire les signaux de la télécommande enregistre au préalable à distance.



Longueur d'onde = Célérité / Fréquence

m = $3*10^{8}$ m/s / $3*10^{17}$ Hz

m = $1*10^{-9}$nm

D’ici la soutenance oral j’ai l’intention de réaliser les points suivants :

-Faire le montage des composants sur la carte 2018.

-Réaliser des tests afin de vérifier le bon fonctionnement.

-Réaliser un programme afin de faire fonctionner les nouvelles fonctionnalités.