



Haute école d'ingénierie et d'architecture Fribourg
Hochschule für Technik und Architektur Freiburg

Travail de Bachelor / 2019-2020

Filière Informatique

Accélérateur de tour télécom

Cahier des charges

30.05.2020 – Version 2

Nicolas Maier

Superviseurs : **Jacques Supcik**
Michael Mäder

Expert : **Frédéric Mauron**

Hes·SO

Haute Ecole Spécialisée
de Suisse occidentale

Fachhochschule Westschweiz

Table des versions

| Version | Date de publication | Auteur | Description |
|---------|---------------------|---------------|--|
| 1 | 22.05.2020 | Nicolas Maier | Ébauche du cahier des charges |
| 2 | 30.05.2020 | Nicolas Maier | Corrections selon commentaires séance 02 |

Table des matières

| | | |
|-----|------------------------|---|
| 1 | Présentation | 3 |
| 1.1 | Acteurs..... | 3 |
| 1.2 | Contexte | 3 |
| 1.3 | Système existant | 4 |
| 1.4 | Objectifs..... | 5 |
| 2 | Activités..... | 7 |
| 2.1 | Analyse | 7 |
| 2.2 | Spécification..... | 7 |
| 2.3 | Conception..... | 7 |
| 2.4 | Réalisation | 7 |
| 2.5 | Tests..... | 7 |
| 2.6 | Documentation | 7 |
| 3 | Dates clés..... | 7 |
| 4 | Planification..... | 8 |

1 Présentation

Cette section décrit le but du projet, la situation actuelle et les objectifs à réaliser.

1.1 Acteurs

Ce projet est suivi par les personnes suivantes :

- Jacques Supcik, *Superviseur*
- Michael Mäder, *Superviseur*
- Frédéric Mauron, *Expert*
- Nicolas Maier, *Étudiant*

1.2 Contexte

Depuis plusieurs années, un projet dénommé "Tour Télécom" est développé au sein de la filière Informatique et Télécommunications. L'objectif de ce projet est de faire la promotion de l'informatique et des systèmes de communication auprès du public lors des manifestations telles que les portes ouvertes.

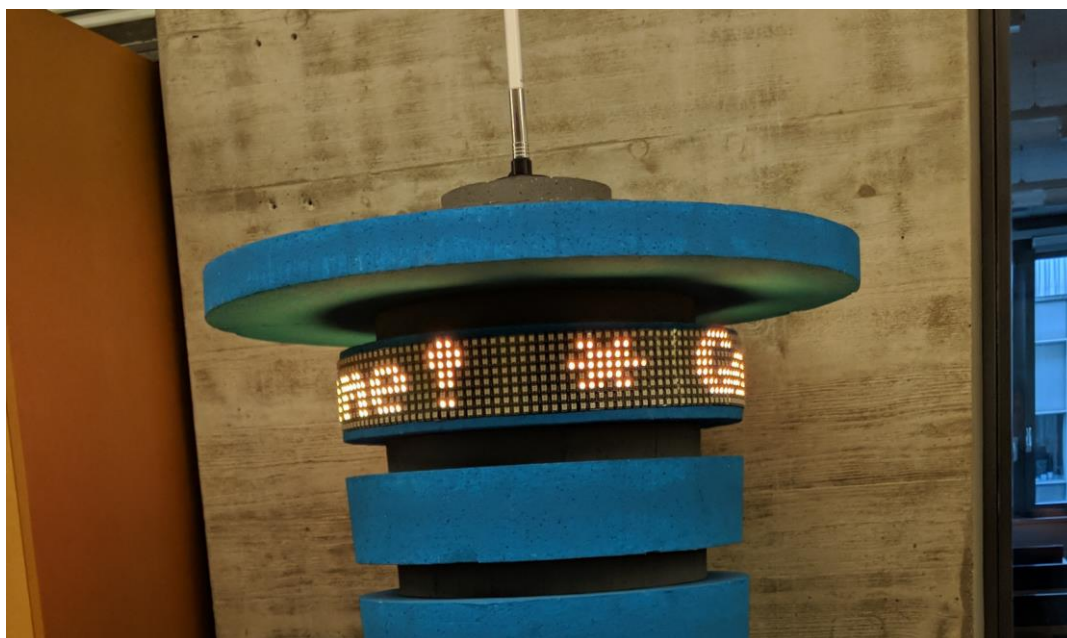


Figure 1 – Photo de la Tour Télécom

La Tour Télécom comporte une matrice de LEDs flexible permettant d'afficher du texte et des animations. C'est sur cet aspect de la tour que ce projet porte.

Ces matrices sont composées de contrôleurs **ws281x**. Chacun de ces contrôleurs gère une LED RGB. Ces contrôleurs sont conçus pour fonctionner avec autant de LEDs que nécessaire : ils sont organisés en chaîne. Pour ajouter une LED, il suffirait d'ajouter un contrôleur **ws281x** à la chaîne. Cette organisation en cascade offre une grande flexibilité, tout en permettant d'adresser chaque LED individuellement. Pour ce faire, un microcontrôleur doit envoyer les valeurs des couleurs de chaque LED (24 bits par LED RGB) au premier contrôleur **ws281x** de la chaîne. Chacun des contrôleurs lit et affiche la première valeur qu'il reçoit, et transmet toutes les autres au suivant. Cela permet de donner la couleur désirée à chacune des LEDs. Notre microcontrôleur doit ensuite envoyer un signal particulier ("reset code"), qui indique à chaque contrôleur **ws281x** qu'il devra à nouveau prendre une valeur, dans le but d'afficher l'image suivante.

Ce système simple et extensible demande toutefois un timing très précis, c'est pourquoi un système temps réel est nécessaire pour contrôler les matrices de LEDs.

Voici une photo montrant en détail une partie d'une bande de LEDs, on y voit de nombreux contrôleurs **ws281x** :

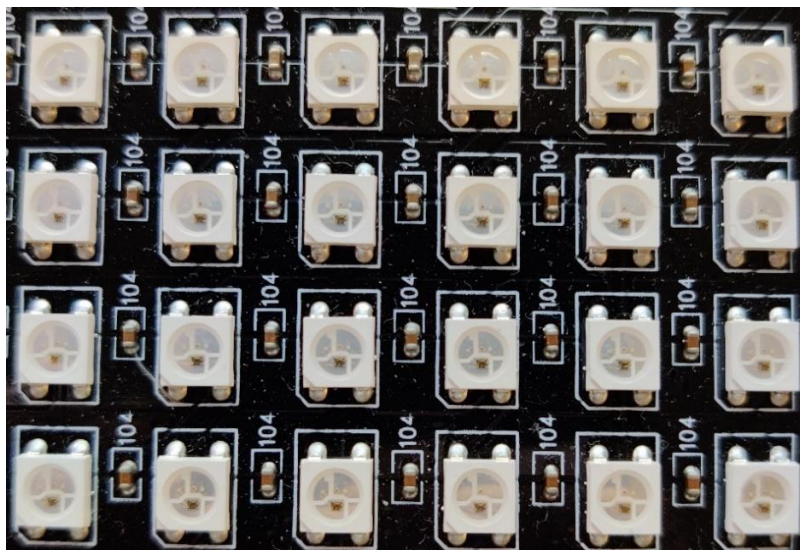


Figure 2 – Détails d'une bande de 8x32 LEDs

Et voici un schéma montrant comment ces contrôleurs sont câblés entre eux pour fonctionner en série :

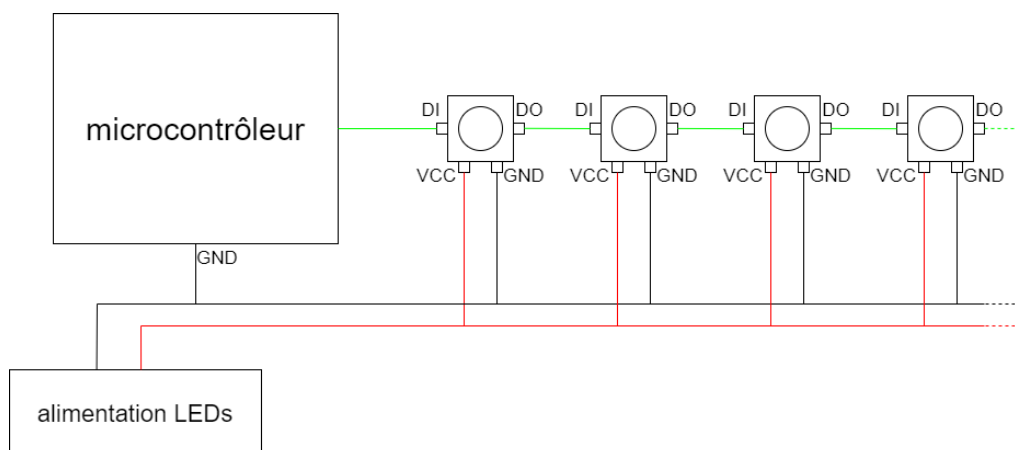


Figure 3 – Câblage des contrôleurs **ws281x**

1.3 Système existant

Actuellement, le système est composé d'un Raspberry Pi qui contrôle une matrice de 4 bandes de 32x8 LEDs RGB NeoPixel. Pour contrôler ces LEDs, le Raspberry Pi doit combiner un DMA et un PWM afin de générer un signal ayant un timing très précis, ce qui ne serait pas possible directement en contrôlant les GPIOs car le système Linux ne permettrait pas de résoudre cette contrainte de temps réel.

Les 4 bandes de 32x8 sont branchées en série, le Raspberry Pi contrôle donc 1024 LEDs. Chaque LED attend 24 bits afin d'obtenir une couleur RGB où chaque couleur est encodée sur 8 bits. Selon le protocole défini dans la datasheet du **ws281x**, les données doivent être transmises à 800 Kb/s (sans compter le "reset code"). La solution actuelle permet donc d'afficher environ 30 images par seconde.

Cette solution est limitée en nombre de LEDs ou en images par seconde. En ajoutant des LEDs en série, on serait obligé de réduire le nombre d'images par secondes, et inversement.

Voici un schéma montrant la situation actuelle, avec 4 bandes de 8x32 LEDs contrôlées par un Raspberry Pi :

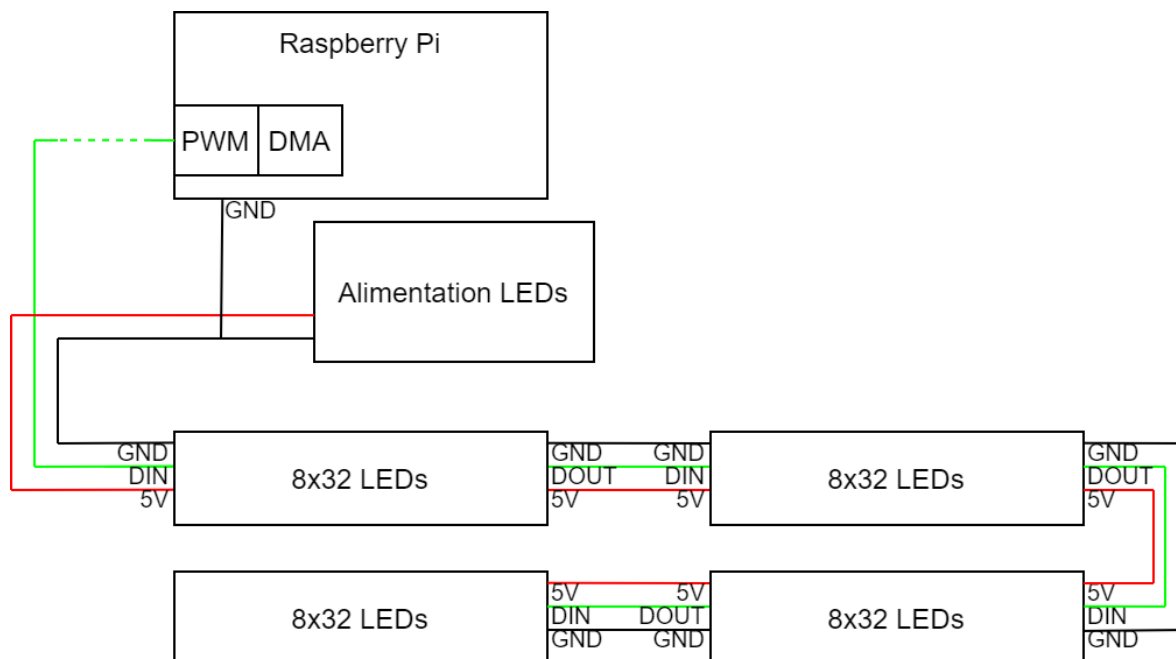


Figure 4 – Situation actuelle

1.4 Objectifs

On aimerait étendre le système avec plus de LEDs, et les rendre plus rapide. Pour ce faire, il faudrait contrôler plusieurs bandes de LEDs en parallèle, en ajoutant au système un/des "coprocesseur(s)" qui s'occuperai(en)t du travail temps réel que réalise actuellement le Raspberry Pi. Par exemple, contrôler 4 groupes de 2 bandes de 32x8 LEDs avec 4 lignes indépendantes permettrait d'obtenir 60 images par seconde, tout en doublant le nombre de LEDs de la Tour Télécom.

Le but de ce projet est de créer une solution permettant de gérer au minimum 4 bandes NeoPixel en parallèle, et fournissant une interface permettant au Raspberry Pi de contrôler les LEDs.

Voici les différents objectifs du projet :

- Un minimum de 4 bandes de LEDs en parallèle est nécessaire, et une cinquième serait utile mais pas absolument nécessaire afin de compléter le tour complet de la tour télécom avec deux carrés de 8x8 LEDs
- Étudier et décrire les limites théoriques de la solution (combien de bandes sont contrôlable au maximum, performances possibles)
- Le système réalisé doit être simple à mettre en œuvre (l'idéal serait de pouvoir simplement brancher la tour télécom en USB à un Raspberry Pi, de façon "plug-and-play")
- L'interface fournie au Raspberry Pi devrait être inspirée de celle qui est actuellement utilisée (https://github.com/jgarff/rpi_ws281x), afin de faciliter la portabilité des projets existants

Voici un schéma montrant l'objectif de ce projet, avec 5 bandes de LEDs contrôlées en parallèle par un microcontrôleur qui communique avec un Raspberry Pi :

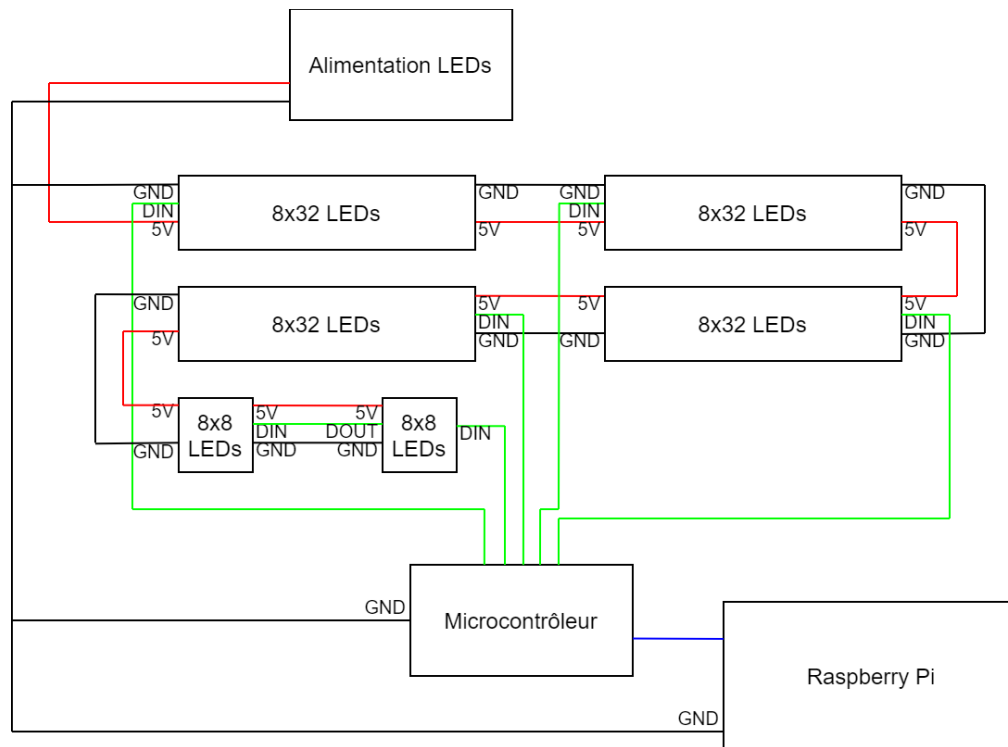


Figure 5 – Objectif du projet

2 Activités

Ce chapitre décrit les différentes activités qui devront être effectuées pour atteindre les objectifs du projet.

2.1 Analyse

- Étude des bandes de LEDs NeoPixel
- Étude des microcontrôleurs existants permettant de contrôler les LEDs
- Étude de l'interfaçage entre le Raspberry Pi et le(s) microcontrôleur(s)
- Choix du matériel adapté pour ce projet

2.2 Spécification

- Définir l'interfaçage fourni par la solution qui permettra au Raspberry Pi de contrôler le système

2.3 Conception

- Définir l'architecture du système, et créer les diagrammes correspondants

2.4 Réalisation

- Implémenter le système capable de contrôler au minimum 4 bandes NeoPixel en parallèle
- Implémenter l'interface permettant au Raspberry Pi de contrôler les LEDs

2.5 Tests

- Définir des tests appropriés
- Effectuer les tests et corriger la réalisation en conséquence

2.6 Documentation

- Réaliser un cahier des charges
- Décrire le contenu et le déroulement du projet dans un rapport technique
- Écrire un manuel permettant la mise en place du système et l'utilisation de ce dernier depuis le Raspberry Pi

3 Dates clés

Dans ce document, des références aux semaines P18 à P21 sont faites bien qu'elles n'existent pas dans le calendrier HEIA-FR. Elles désignent les semaines du 22.06.2020 au 19.07.2020, en suivant la même logique que les semaines P13 à P17 qui désignent les semaines du 18.05.2020 au 21.06.2020.

Voici les différentes dates clés du projet :

- 29.05.2020 (P14) : Validation du cahier des charges
- 04.06.2020 (P15) : Choix du matériel
- 17.06.2020 (P17) : Validation de la spécification
- 19.05.2020 (P17) : Validation de la conception et choix des technologies
- 15.07.2020 (P21) : Validation de la réalisation
- 17.07.2020 (P21) : Rendu du rapport
- 26.08.20 : Rendu du poster
- 28.08.20 ? (À définir) : Exposition du travail
- 31.08.2020 – 02.09.2020 (À définir) : Défense orale

4 Planification

Voici la répartition planifiée des activités sur la durée du projet :

| | P13 | P14 | P15 | P16 | P17 | P18 | P19 | P20 | P21 |
|--|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Documentation | | | | | | | | | |
| Cahier des charges | | | | | | | | | |
| Validation cahier des charges | | Ve | | | | | | | |
| Rapport technique | | | | | | | | | |
| Manuel d'utilisation | | | | | | | | | |
| Rendu du rapport | | | | | | | | Ve | |
| Analyse | | | | | | | | | |
| Étude des bandes de LEDs NeoPixel | | | | | | | | | |
| Étude des microcontrôleurs | | | | | | | | | |
| Étude de l'interfaçage entre le RPi et le(s) µp(s) | | | | | | | | | |
| Choix du matériel | | | Je | | | | | | |
| Spécification | | | | | | | | | |
| Spécification de l'interface entre le Rpi et le(s) µp(s) | | | | | | | | | |
| Validation de la spécification | | | | | Me | | | | |
| Conception | | | | | | | | | |
| Définition de l'architecture, diagrammes correspondants | | | | | | | | | |
| Validation de la conception et choix des technologies | | | | | Ve | | | | |
| Réalisation | | | | | | | | | |
| Implémentation du contrôle des bandes NeoPixel | | | | | | | | | |
| Implémentation de l'interface RPi/µp(s) | | | | | | | | | |
| Tests | | | | | | | | | |
| Tests appropriés | | | | | | | | | |
| Validation de la réalisation | | | | | | | | Me | |
| Défense orale | | | | | | | | | |

(Semaine des examens)