UNIVERSITE DE LORRAINE
IUT NANCY-BRABOIS
Département GEII

RAPPORT D'ETUDE ET REALISATION INFORMATIQUE

Compte-rendu projet Carrefour

Thomas MERTZ

Année universitaire 2016-2017

UNIVERSITE DE LORRAINE
IUT NANCY-BRABOIS
Département GEII

RAPPORT D'ETUDE ET REALISATION INFORMATIQUE

Compte-rendu projet Carrefour

Thomas MERTZ

Année universitaire 2016-2017

SOMMAIRE

INTRODUCTION

Les carrefours sont des lieux dangereux pour les voitures et les piétons. Il est donc important qu'ils fonctionnent parfaitement et sans accros. Ainsi les programmes qui commandent les feux de ces carrefours doivent être complets et respecter les différentes règles de circulation présente en France.

.

1. Première partie : Présentation du projet

Il est important de définir les différents acteurs autour de ce projet et surtout son fonctionnement

1.1. Acteurs du projet

Ici, le client peut être une ville, une agglomération ou plus simplement une entreprise qui a besoin de gérer de la circulation et plus précisément un carrefour routier.

Les différentes ressources : librairies, maquette et simulation ont été fournis par le département GEII ce qui permettra d'effectuer simplement et sans risque ce projet.

1.2. But du projet

Le but est de créé un programme en langage C# qui gérera automatique les feux d'un carrefour en fonction des différents paramètres définies par le client. On a ici à gérer un croisement de routes (4 feux) avec des détecteurs de voiture et des boutons pour piétons.

2. Deuxième partie : Analyse du besoin

2.1. Analyse de l'existant

Nous avons actuellement accès à une librairie C# permettant de contrôler jusqu'à quatre feux, permettant de détecter des voitures et des piétons.

Il existe également une maquette physique et une simulation d'un carrefour à quatre feux. On y trouve également des détecteurs de voitures (il suffit de cliquer sur la route pour la simulation) et de boutons piétons (Il faut appuyer sur les feux orange de la route)

2.2. Analyse du besoin

Il faut d'abord gérer les feux convenablement en utilisant les valeurs entrées dans un fichier texte par l'utilisateur. Il faut ensuite bien gérer l'utilisation des capteurs (voitures, piétons). On pourra aussi enregistrer les valeurs des capteurs.

3. Troisième partie : Analyse fonctionnelle et technique

3.1. Fonctionnement général

Le programme se déroule assez simplement : On va d'abord récupérer les différents variables définis par l'utilisateur (la durée des feux et l'adresse de connexion). A partir du moment où on est connecté on va rester en permanence dans un cycle pour changer la couleur des feux tant que l'utilisateur ne choisit pas d'arrêter les feux.

Le cycle des feux fonctionne simplement : un feu reste toujours au vert au minimum le temps qui lui est défini. Mais il peut rester plus longtemps au vert s'il n'y a pas de piétons qui demandent à passer, qu'il n'y a pas de véhicule sur la voie opposée et que l'on n'a pas dépassé la durée maximum du vert. Ensuite on va simplement le passer à l'orange puis au rouge l'autre voie pouvant ainsi passer au vert.

On a aussi une fonction d'archivage qui va enregistrer ce qui fait passer le feu au rouge (un piéton, une voiture).

3.2. Fonctionnement principale

Une fonction Main qui appel 5 fonctions : trois de mise en place et deux pour faire fonctionner un cycle de feux.

On commence par récupérer les temps maximum et minimum de durée de feux dans un tableau en appelant la fonction « LectureFichier ». Ensuite on demande à l'utilisateur de choisir s'il veut se connecter à la maquette ou à la simulation avec la fonction « choixIP ». Enfin on met en place l'état initiale des feux (fonction « setup ») : voie 1 au vert, voie 2 au rouge. On arrive maintenant dans un boucle infini qui ne peut être brisée qu'en appuyant sur la touche 'échap' et qui permet de faire fonctionner les feux.

3. Troisième partie : Analyse fonctionnelle et technique

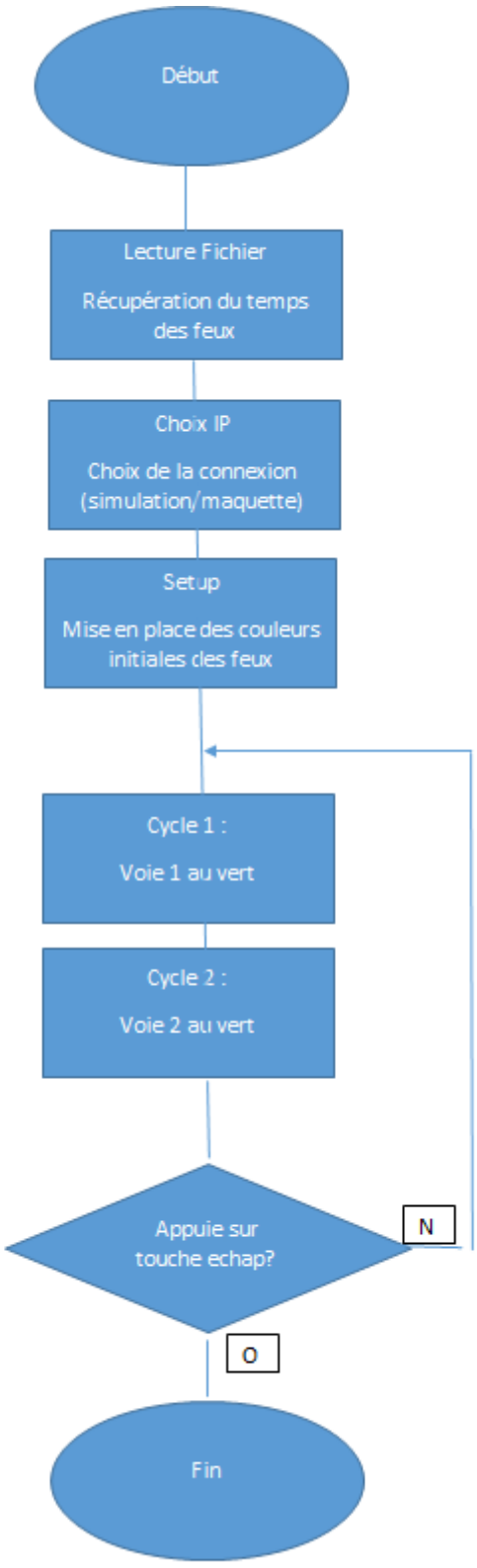

Figure 1 Algorithme principal

3. Troisième partie : Analyse fonctionnelle et technique

3.3. Fonctionnement du cycle

Le fonctionnement est séparé en deux parties :

- Partie 1 : la voie 1 est au vert
- Partie 2 : la voie 2 est au vert

Ces deux fonctions sont vraiment très semblables, et on peut les décrire simplement.

On va commencer par créer des variables :

- « localDate» permet de mesurer le temps
- « pieton » de savoir si un piton a appuyé sur un bouton
- « voiture » pour savoir si une voiture est détectée
- « mesure » un tableau permettant d'enregistrer les valeurs des capteurs

On va alors entrer dans une boucle appelant de manière régulière (toutes les 100 millisecondes) les fonctions permettant de connaitre l'état des capteurs (piétons et voitures).

On pourra sortir de cette boucle uniquement si le feu est au vert depuis au moins le temps minimum ou que l'on a atteint la durée maximum de vert. Si l'on détecte un piton qui veut traverser ou une voiture sur la voie d'en face et que le feu est vert depuis suffisamment de temps (le temps minimum) alors on pourra aussi sortir de cette boucle.

C'est à la sortie que l'on va écrire dans le fichier les mesures que l'on a effectué et que l'on va changer la couleur des feux.

3. Troisième partie : Analyse fonctionnelle et technique

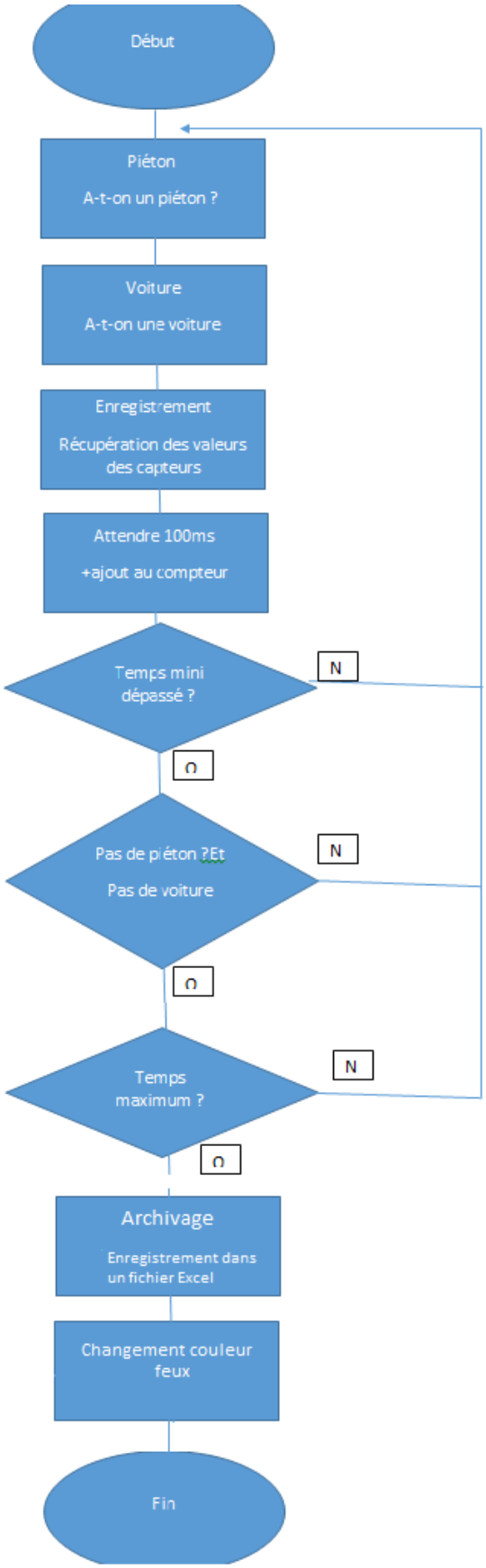

Figure 2 Algorithme boucle Feux

3. Troisième partie : Analyse fonctionnelle et technique

3.1. Fonctionnement du changement de couleur

Cette partie du code est très simple :

Si on est au vert, on passe à l'orange pendant 3 secondes (cela correspond à la durée classique pour le feu orange). Puis on passe au rouge.

Et si on est au rouge, on passe au vert.

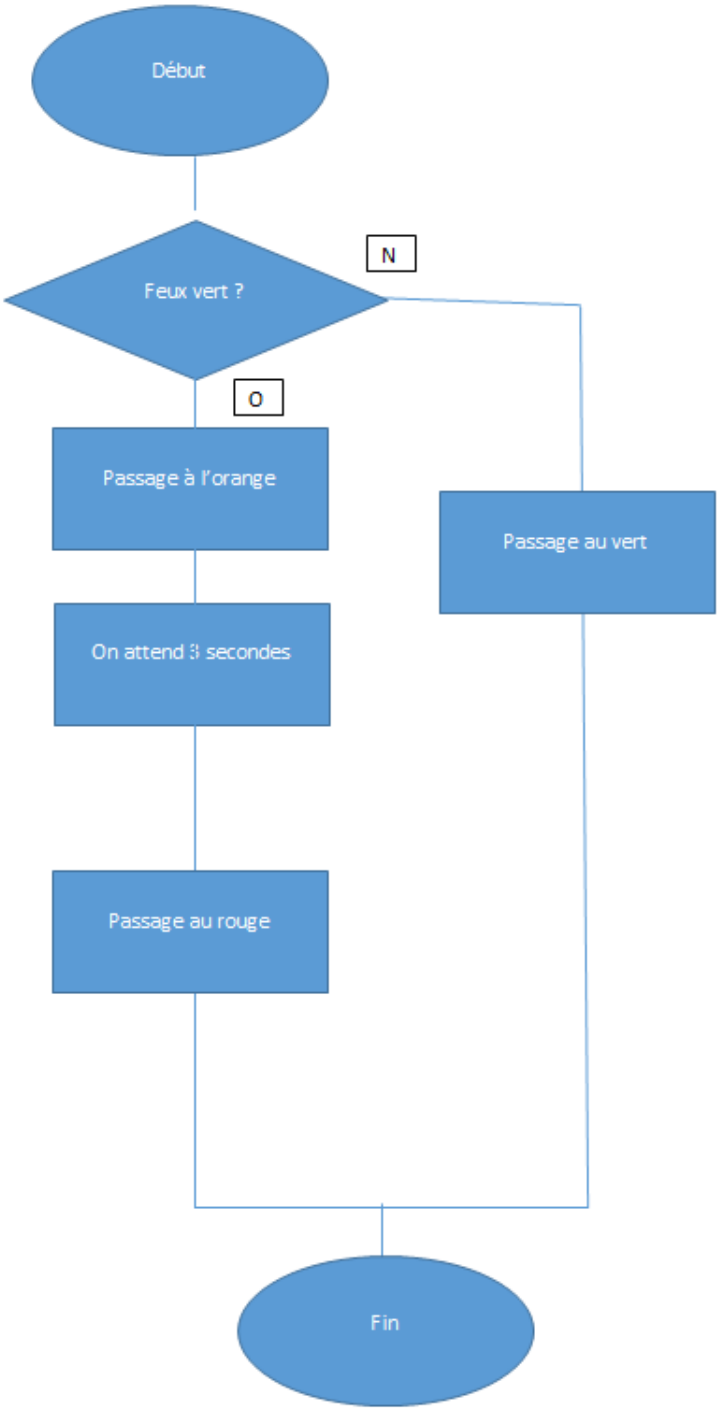

Figure 3 Algorithme changement de couleur

3. Troisième partie : Analyse fonctionnelle et technique

3.2. Fonctionnement de la lecture du fichier

Pour lire les informations qui se trouvent dans le fichier texte, on récupère tout le texte du fichier que l'on sépare en fonction des espaces et des retours à la ligne.

On obtient ainsi un tableau et il suffit de récupérer les valeurs 3,4,5,6 pour obtenir les temps maximum et minimum des feux.

On les multiplie par 1000 pour obtenir un temps en millisecondes plus facilement exploitable.

3.3. Fonctionnement de l'écriture dans un fichier

Pour écrire dans le fichier on va simple récupérer les mesures que l'on a effectué sur les capteurs.

Ainsi lorsque qu'on a détecté quelque chose on a enregistré la date et la voie où l'événement à eu lieu. On enregistre donc dans le fichier .csv ces événement sans écraser les données qu'on l'on a déjà.

CONCLUSION

A l'aide des documents mis à disposition, j'ai pu écrire un programme qui gère un carrefour a 4 feux en prenant en compte les véhicules et les piétons. On a une interaction avec l'utilisateur car c'est lui qui choisit s'il veut se connecter à la maquette ou à la simulation, et qui choisit les différents temps des feux.

Pour compléter mon programme il aurait été possible de rajouter des tests pour vérifier si le fichier Config.txt était bien présent. J'aurais aussi pu découper mon programme en différents form.

TABLE DES FIGURES