

## 2 Cahier des charges

L'application a pour but de réguler le flux de circulation (véhicule, piéton) traversant un carrefour. Un carrefour routier<sup>1</sup> est au sens propre la zone comprise à l'intérieur du prolongement des bordures (ou, s'il n'y a pas de bordures, des rives) de deux chaussées qui se coupent à angle droit ou presque droit. Plus généralement il s'agit de la zone dans laquelle des véhicules se déplaçant sur des routes différentes qui se coupent à angle quelconque, peuvent se rencontrer. Une chaussée peut avoir plusieurs voies de circulation.

La branche est la portion de n'importe laquelle des chaussées raccordées au carrefour et située à l'extérieur de la zone du carrefour proprement dit.

On distingue deux parties dans une branche :

- l'entrée est la voie de cette branche que parcourt la circulation qui se dirige vers le carrefour ;
- la sortie est la voie de cette branche que parcourt la circulation lorsqu'elle quitte le carrefour ;

Une voie entrante peut être équipée des signalisations suivantes :

- feu tricolore : voyant rouge, voyant orange, voyant vert ;
- indicateur piéton : piéton rouge, piéton vert ;
- demande de passage piéton ;
- des détecteurs de présence de véhicules.

Pour l'application à développer, la topologie du carrefour est celle de la maquette présentée à la section 1.2 page 5. Il est quand même à prévoir une évolution de cette topologie pour supporter d'autres maquettes.

### 2.1 Les cas d'utilisation

Deux cas d'utilisation sont demandés comme montré par la figure ci-dessous : le premier permet de tester le bon fonctionnement de chaque constituant (capteur, actionneur, liaison RS-232) du carrefour, le second, régule le trafic traversant le carrefour en fonction du temps : cycles des feux, durées d'allumage des feux, jour-nuit), des piétons, des véhicules et de la police (contrôle manuel du trafic).

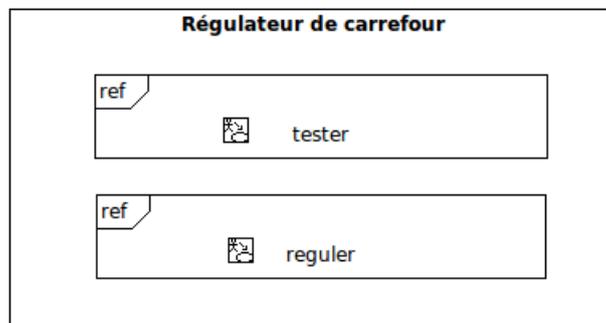


Figure 1 – Les besoins – Diagramme de cas d'utilisation

1. source Wikipedia

### 2.1.1 Cas d'utilisation tester

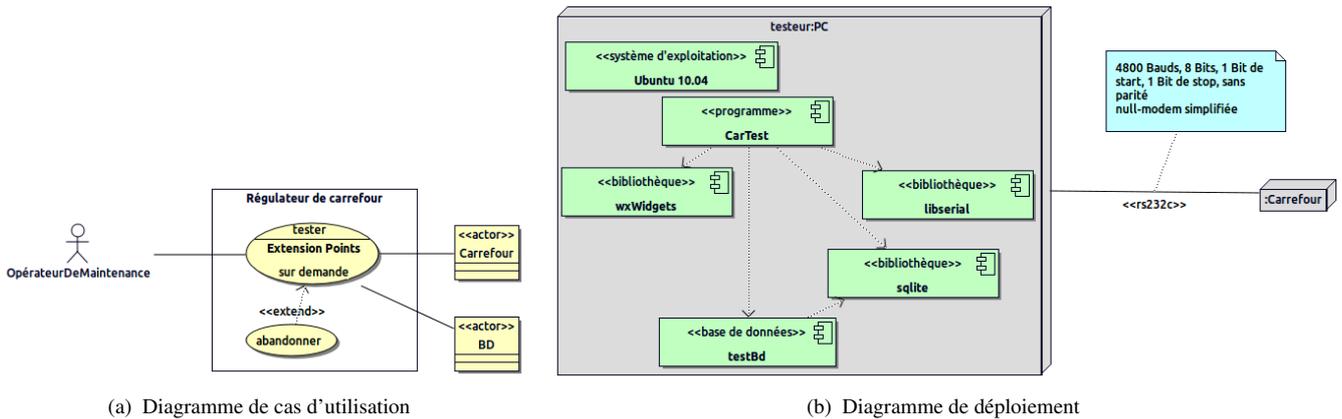


Figure 2 – Les besoins – Cas d'utilisation – tester

#### Objectif

Permettre à un opérateur de maintenance de valider le bon fonctionnement de chaque constituant (capteur, actionneur, liaison RS-232C du carrefour).

Listes des capteurs :

- 2 boutons "Appel piéton" : 1 par voie ;
- 1 bouton poussoir d'alarme ;
- 1 récepteur infra rouge (voie 1) ;
- 1 récepteur lumineux (voie 2) ;
- 1 détecteur jour/nuit.

Listes des actionneurs :

- 4 feux tricolores (rouge, orange et vert) : 2 par voie ;
- 2 feux bicolores (rouge et vert) : 1 par voie ;
- 1 émetteur infra rouge (voie 1) ;
- 1 émetteur lumineux (voie 2).

#### Pré conditions

- aucune.

#### Acteurs

- l'opérateur de maintenance agit sur les actionneurs et vérifie l'état des différents capteurs ;
- le carrefour reçoit les ordres via la liaison série et agit en fonction de l'ordre reçu ;
- la Base de données (BD) stocke la configuration et le protocole de communication avec le carrefour ainsi que les comptes rendus de test ; un compte rendu contient les résultats de test ainsi que la date et l'heure de la session de test.

#### Scénario nominal

1. le système extrait de la BD la configuration et le protocole de communication avec le carrefour
2. le système configure la voie série RS-232C avec les données de configuration extraites de la BD ;
3. le système établit la liaison avec le carrefour ;
4. le système indique à l'opérateur de maintenance que la connexion est établie ;

5. le système émet vers le carrefour des ordres de manière à positionner chaque actionneurs dans un état connu et de vérifier que tous les capteurs sont inactifs ;
6. l'opérateur de maintenance indique au système le début du test ;
7. l'opérateur de maintenance sélectionne un actionneurs ( voir § Listes des actionneurs page précédente) à tester ;
8. l'opérateur de maintenance sélectionne l'action à réaliser pour l'actionneurs sélectionné ;
9. le système émet l'ordre correspondant à l'action voulue pour l'actionneur sélectionné vers le carrefour ;
10. l'opérateur de maintenance vérifie visuellement sur le carrefour que l'état du actionneur sélectionné ;
11. l'opérateur de maintenance indique au système le résultat du test de l'action voulue pour l'actionneur sélectionné ;
12. le système retire l'action testée de l'actionneur sélectionné des tests possibles ;
13. les points de 7 à 12 sont répétées pour tous les tests possibles ( $card(tests\ actionneurs) = card(actionneurs) \times card(actions)$ ) ;
14. l'opérateur de maintenance sélectionne un capteur ( voir § Listes des capteurs page précédente) à tester ;
15. l'opérateur de maintenance sélectionne l'état à lire pour le capteur sélectionné ;
16. l'opérateur de maintenance agit sur le capteur sélectionné sur le carrefour pour obtenir l'état à tester ;
17. le système émet, sur demande de l'opérateur de maintenance, l'ordre de lecture du capteur sélectionné vers le carrefour ;
18. le système vérifié que l'état lue correspond à l'état à tester pour le capteur sélectionné ;
19. le système indique le résultat du test à l'opérateur de maintenance ;
20. le système retire le capteur testé de la liste des capteurs ;
21. le système retire l'état testée du capteur sélectionné des tests possibles ;
22. les points de 14 à 21 sont répétées pour tous les tests possibles ( $card(tests\ capteurs) = card(capteurs) \times card(etats)$ ) ;
23. le système enregistre dans la BD un compte rendu contenant les résultats de test ainsi que la date et l'heure de la fin de la session de test.

### Post conditions

- un compte rendu de test supplémentaire dans la BD.

### Scénarii alternatifs

**A1 - abandonner** : démarre à n'importe quel point du scénario nominal et sur confirmation de l'opérateur de maintenance.

- A1.1. le système ré initialise les tests effectués ;
  - A1.2. le système ré initialise la liste des tests ;
- Le scénario nominal reprend au point 5.

### Scénarii d'exception

**E1 - déconnexion** : apparaît à n'importe quel point du scénario nominal.

- E1.1. le système interrompt le test en cours ;
- E1.2. le système informe l'opérateur de maintenance ;
- E1.3. l'opérateur de maintenance vérifie la connexion ;
- E1.4. l'opérateur de maintenance indique au système la poursuite du test ou signifie au système l'abandon du test ; dans ce dernier cas les points 1 à 2 du scénario alternatif **A1 - abandonner** sont exécutés puis le scénario nominal reprend au point 3 page précédente.

## 2.1.2 Cas d'utilisation réguler

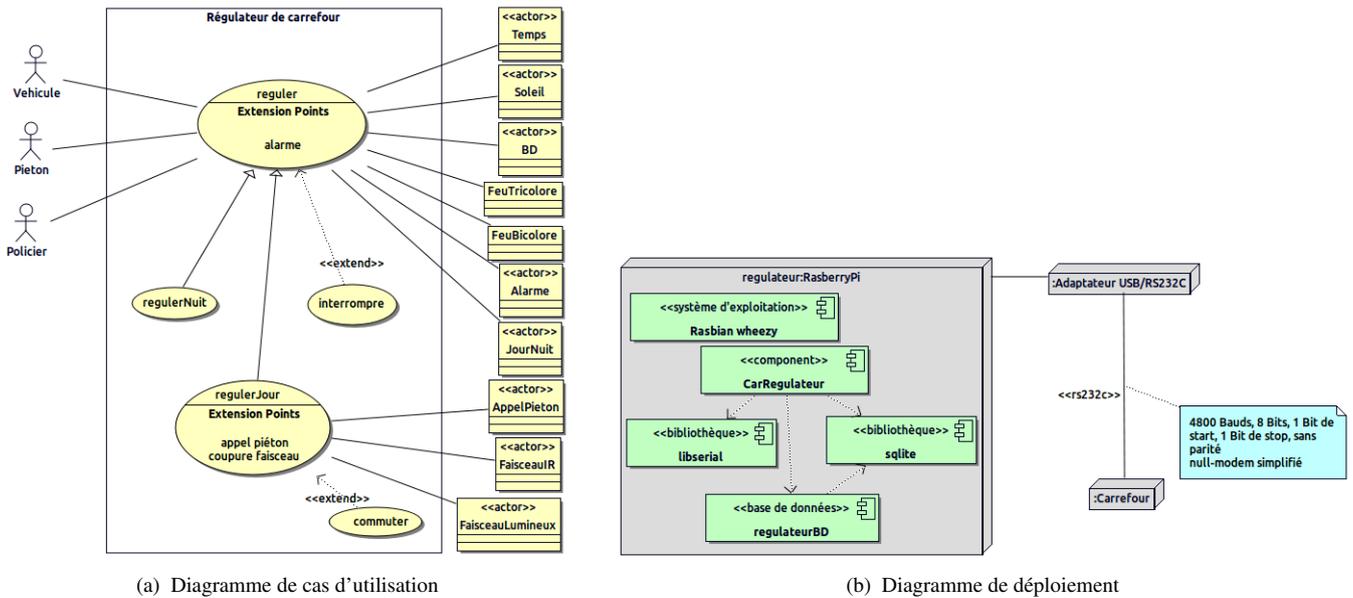


Figure 3 – Les besoins – cas d'utilisation – **reguler**

### Objectif

Réguler le trafic traversant le carrefour en fonction du temps : cycles des feux, durées d'allumage des feux, jour-nuit), des piétons, des véhicules et de la police (contrôle manuel du trafic). La régulation est réalisée par des feux tricolores, pour les véhicules, et bicolores, pour les piétons.

### Pré conditions

- aucune.

### Acteurs

- le véhicule circule sur la voie ; il peut être stoppé par un feu tricolore au rouge, son passage au niveau du feu tricolore est détecté par un faisceau infra rouge (*i.e.* IR) sur la voie 1 et par un faisceaux lumineux sur la voie 2 ;
- le piéton traverse la voie sur le passage qui lui est réservé ; le passage est rythmé par un feu bicolore qui prend alternativement les couleurs rouge et verte ; le changement dépend du cycle du feu tricolore ; le changement peut être également provoqué par appui sur le bouton « Appel piéton » ;
- le policier peut interrompre la régulation par appui sur le bouton « Alarme » ;
- le temps assure la durée d'allumage des différents feux du carrefour ;
- le soleil agit sur le capteur jour/nuit ; les cycles de régulations diffèrent le jour et la nuit ;
- la BD stocke :
  - la configuration et le protocole de communication du carrefour ;
  - le seuil de luminosité en lux ;
  - les différentes temporisations composant les cycles de régulation :
    - \* durée des feux tricolores au vert : 10 secondes ;
    - \* durée des feux tricolores à l'orange : 2 secondes ;
    - \* temps de dégagement : 2 secondes ;
    - \* durée des feux tricolores au rouge : somme des durées au vert , à l'orange et d'arrêt de la circulation ;
    - \* période de clignotement des feux tricolores à l'orange durant la nuit : 1 seconde ;
    - \* durée de décalage des feux tricolores à l'orange durant la nuit : 1 seconde ;

- \* période de clignotement des feux tricolores au rouges en alarme : 1 seconde ;
- \* durée des feux bicolores au vert : durée des feux tricolores au rouge moins 2 secondes.
- les informations des alarmes ; date, heure et durée ;
- les déconnexions de la liaison série : date, heure et durée.
- le feu tricolore régule la circulation des véhicules sur la voie :
  - rouge : arrête les véhicules ;
  - orange : prévient les véhicules du passage au rouge ;
  - vert : autorise le passage des véhicules .
- le feu bicolore régule la circulation des piétons sur la voie :
  - rouge :arrête les piétons ;
  - vert : autorise le passage des piétons .
- le capteur « JourNuit » prévient le système de la tombée de la nuit ou du levé du jour ; la détection dépend du seuil de luminosité mémorisé dans la BD ;
- le bouton « Alarme » permet un fonctionnement particulier des feux du carrefour ;
- le bouton « Appel piéton » permet au piéton de demander le passage ;
- le faisceau infra rouge, constitué d'un émetteur et d'un récepteur infra rouge, coupe la voie 1 au niveau du feu tricolore ; il permet de détecter le passage d'un véhicule ou la présence d'un véhicule arrêté par le feu tricolore ;
- le faisceau lumineux, constitué d'une Diode électro lumineuse (DEL) et d'un récepteur de lumière, coupe la voie 2 au niveau du feu tricolore ; il permet de détecter le passage d'un véhicule ou la présence d'un véhicule arrêté par le feu tricolore ;

**Scénario nominal**

1. le système extrait de la BD :
  - la configuration et le protocole de communication avec le carrefour ;
  - les cycles de régulation ;
  - le seuil de luminosité.
2. le système configure la voie série RS-232C avec les données de configuration extraites de la BD ;
3. le système établie la liaison avec la carrefour ;
4. le système émet les ordres au carrefour pour obtenir le cycle de régulation présenté par le tableau 1, ci-dessous, durant le jour et celui montré par le tableau 2, page suivante durant la nuit.
  - (a) le jour, le flux de circulation des véhicules est autorisé (feux vert ou orange) alternativement dans les deux voies. La traversée des piétons est possible lorsque le flux de circulation des véhicules est interdit.

Phase	Durée (seconde)	Voie 1		Voie 2	
		Feux tricolores	Feu bicolore	Feu tricolore	Feu bicolore
1	8	R	V	V	R
2	2	R	R	V	R
3	2	R	R	O	R
4	2 <sup>a</sup>	R	R	R	R
5	8	V	R	R	V
6	2	V	R	R	R
7	2	O	R	R	R

Tableau 1 – Cycle de régulation de jour

*a.* temps de dégagement

- (b) la nuit, le flux de circulation des véhicules est autorisé dans les deux voies. Un avertissement lumineux de prudence est mise en place ; il consiste à un imbrication des clignotement des feux oranges de chaque voie ; la traversée des piétons est interdites.

Phase	Durée (seconde)	Voie 1		Voie 2	
		Feux tricolores	Feu bicoloré	Feu tricolore	Feu bicoloré
1	1	O	R	E <sup>a</sup>	R
2	1	E	R	O	R

Tableau 2 – Cycle de régulation de nuit

a. tous les feux sont éteints

### Scénarii alternatifs

**A1 - commuter** : enclenché dans les phases 1 (voie 2 passante) ou 5 (voie 2 passante) du cycle de régulation de jour (table 1 page précédente) du point 4 page précédente du scénario nominal, par les deux événements suivants :

- appui du bouton « Appel piéton » de la voie passante ;
- coupure du faisceau (lumineux ou IR) de la voie bloquée.

A1.1. le système interrompt la phase courante (*i.e.* arrêt de la temporisation) du cycle de régulation de jour ;

A1.2. le système saute la phase suivante du cycle de régulation de jour (table 1 page précédente).

Le scénario nominal reprend au point 4 page précédente à la phase 3, si apparition à la phase 1, ou 7, si apparition à la phase 5 du cycle de régulation de jour (table 1 page précédente) .

**A2 - Alarme** : enclenché par l'appui sur le bouton « Alarme » à n'importe quel point du scénario nominal et du scénario alternatif « commuter » qu'il fasse jour ou nuit .

A2.1. le système mémorise l'état actuel du carrefour : cycle jour/nuit, phase dans le cycle ;

A2.2. le système émet les ordres au carrefour de manière à faire passer tous les feux (tricolores et bicolorés) au rouge ;

A2.3. le système lit l'état du bouton « Alarme » du carrefour jusqu'à son relâchement ;

A2.4. le système émet successivement au carrefour les ordres d'extension et allumage du feu rouge de tous les feux tricolores de manière à les faire clignoter ; la période de clignotement est de 1 seconde ; le clignotement dure jusqu'à un appui sur le bouton « Alarme » ;

A2.5. le système restaure l'état du carrefour au moment de l'alarme : cycle jour/nuit, phase dans le cycle ;

Le scénario interrompu reprend au point auquel l'alarme est apparue.

Le tableau 3, ci-dessous synthétise le cycle lumineux mise en œuvre lors d'une alarme.

	Durée (seconde)	Voie 1		Voie 2	
		Feux tricolores	Feu bicoloré	Feux tricolores	Feu bicoloré
état initial		R	R	R	R
après relâchement	1	E <sup>a</sup>	R	E	R
clignotement	1	R	R	R	R
"	1	E	R	E	R

Tableau 3 – Cycle lumineux en alarme

a. tous les feux sont éteints

### Scénarii d'exception

**E1 - déconnexion** : apparaît à n'importe quel point du scénario nominal ou des scénarii alternatifs.

E1.1. le système interrompt la régulation en cours ;

E1.2. le système mémorise l'état actuel du carrefour : cycle jour/nuit, phase dans le cycle ;

E1.3. le système mémorise la date, l'heure de la déconnexion dans la BD ;

E1.4. le système attend la reconnexion ;

E1.5. le système mémorise la durée de la déconnexion dans la BD ;

E1.6. le système restaure l'état du carrefour au moment de l'alarme : cycle jour/nuit, phase dans le cycle ;

Le scénario interrompu reprend au point auquel la déconnexion a été détectée.

### Post conditions

- aucune.

## 3 Contrainte des développement

- Personal computer (PC) Ubuntu 10.04 LTS x86 32 bits ;
  - chaînes de production GNU Compiler Collection (GCC) pour architecture Intel X86 ;
  - chaînes de production croisée GCC pour architecture ARM V7 ;
  - base de données SQLite – bibliothèque C SQLite ;
  - bibliothèque de gestion de la voie série RS-232 : *Libserial* ;
  - bibliothèques graphiques *WxWidgets*, *Simple directmedia layer (SDL)* ;
  - outils de développement d'interface homme-machine (IHM) : *Wxformbuilder*.
  - environnement de développement intégré – Eclipse C/C++ Development Toolkit (CDT) 3.7 ;
- langage de programmation C++ ;
- un mini ordinateur Raspberry Pi sous Rasbian Wheezy ;
  - base de données SQLite – bibliothèque C SQLite ;
  - bibliothèque de gestion de la voie série RS-232 : *Libserial* ;

L'application « CarRegulateur » est développée sur un PC sous Ubuntu 10.04 LTS x86 32 bits ; elle est générée par la chaîne de production croisée GCC pour architecture ARM V7 puis téléchargée dans le mini ordinateur Raspberry Pi via le protocole **ssh**. La configuration de mise à jour via ssh est présentée par le figure 4, ci-dessous.

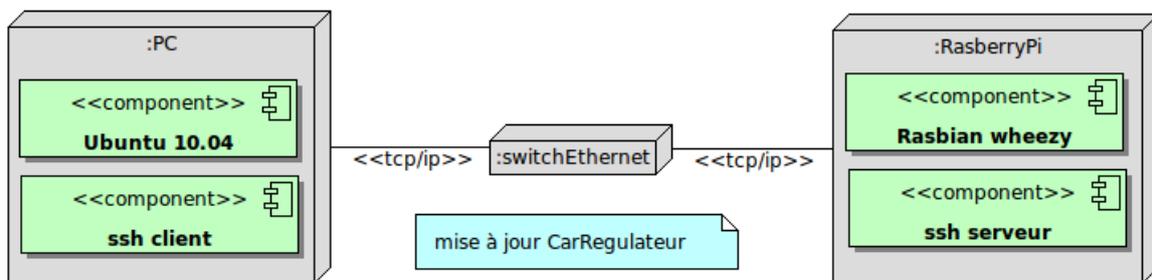


Figure 4 – Configuration de mise à jour du régulateur