

**REPUBLIQUE DEMOCRATIQUE DU CONGO**

**ENSEIGNEMENT SUPERIEUR ET UNIVERSITAIRE**

**UNIVERSITE LIBRE DE PAYS DES GRANDS LACS**

**Département de Génie Electrique et Informatique**



BP. 368 GOMA

[www.ulpgl.net](http://www.ulpgl.net)

# **CONTROLE ET SURVEILLANCE DU TRAFIC ROUTIER URBAIN**

Par **RUKUNDO RWANDARUGARI Bienfait**

Travail de fin de cycle présenté en vue de l'obtention  
du diplôme d'Ingénieur Civil.

**Option : Génie Informatique**

**Directeur : Prof. Dr. Baraka MUSHAGE Olivier**

**Encadreur : Ir. MBUYAMBA KALUBI Célestin**

**Année académique 2016 – 2017**

# **EPIGRAPHE**

L'imagination est la meilleure compagnie de transport au monde

**Roger Fournier**

# **DEDICACE**

*A mes parents Augustin RWANDARUGARI et Bernadette NTWAMWEMERA.*

**RUKUNDO RWANDARUGARI Bienfait**

# REMERCIEMENTS

A Dieu, le Tout Puissant, le Miséricordieux, qui nous a donné l'opportunité de mener à bien ce travail.

Les plus sincères au Professeur Dr. OLIVIER BARAKA MUSHAGE qui, en dépit de ses multiples occupations, a accepté de diriger ce travail. Sa disponibilité inconditionnelle, ses conseils éclairants et son savoir-faire nous ont permis de mener à bon port ce travail, à l'Ir Célestin MBUYAMBA qui a accepté d'être l'encadreur de ce travail. Sans oublier l'Ir KAMY MUISA, l'Ir Raoul IRENGE pour leurs contributions.

Les plus vifs s'adressent aussi à tout le cadre professoral et Administratif de la faculté des sciences et Technologies appliquées.

Toute ma gratitude à mes chérîmes parents Augustin RWANDARUGARI et Bernadette NTAMWEMERA, pour leurs conseils et sacrifices en qualité des vaillants parents, dignes et soucieux de l'avenir de leurs enfants; sans votre amour, sans votre tendresse, sans votre affection parentale, sans votre génie inventif et l'oubli de soi, nous ne serions jamais ce que nous sommes et cette œuvre ne serait jamais mise au point.

Ma gratitude est grande vis-à-vis de tous mes collègues et compagnons de lutte pour leurs encouragements, je citerai plus particulièrement : Tripa VINGI, Elie MAYOGHA, Pierre MUKISA, Alain MAPENDO, Cédric BEBYO, Enock KATEGHE, Fidèle OBUTU et Christian KAKASU.

Je ne terminerai pas sans avoir exprimé mes remerciements envers toutes les personnes qui ont contribué de près ou de loin à la réalisation de ce travail.

**RUKUNDO RWANDARUGARI *Bienfait***

# TABLE DES MATIERES

<b>EPIGRAPHE</b> .....	<b>i</b>
<b>DEDICACE</b> .....	<b>ii</b>
<b>REMERCIEMENTS</b> .....	<b>iii</b>
<b>SIGLES ET ABREVIATIONS</b> .....	<b>vii</b>
<b>LISTE DES FIGURES</b> .....	<b>viii</b>
<b>LISTE DES TABLEAUX</b> .....	<b>viii</b>
<b>RESUME</b> .....	<b>x</b>
<b>ABSTRACT</b> .....	<b>xi</b>
<b>CHAPITRE 0: INTRODUCTION GENERALE</b> .....	<b>1</b>
0.1. PROBLEMATIQUE .....	1
0.2. HYPOTHESES .....	2
0.3. OBJECTIFS ET DELIMITATION DU TRAVAIL.....	2
0.4. METHODES ET TECHNIQUES DE TRAVAIL .....	3
0.5. INTERET ET SUBDIVISION DU TRAVAIL.....	4
<b>CHAPITRE I: ETAT DE L'ART ET CONCEPTS DE BASE</b> .....	<b>5</b>
I.1. ETAT DE L'ART .....	5
I.2. NOTRE APPROCHE .....	6
I.3. CONCEPTS .....	6
I.4. PRESENTATION D'ARDUINO [19].....	10
I.5. CONCLUSION PARTIELLE.....	11
<b>CHAPITRE II: GENERALITES SUR LE TRAFIC ROUTIER URBAIN</b> .....	<b>12</b>
II.1. INTRODUCTION SUR LE TRAFIC ROUTIER .....	12
II.2. COMPOSANTES DU TRAFIC ROUTIER.....	13
II.3. LE DIAGRAMME FONDAMENTAL DU TRAFIC .....	14
II.4. MESURES USUELLES DU TRAFIC .....	15
II.4.1. Les capteurs pneumatiques .....	15
II.4.2. Les boucles électromagnétiques .....	16
II.4.3. Principe de fonctionnement d'un capteur à ultrason HC-SR04 :	16
II.4.4. Le capteur vidéo.....	18

II.5. MOTIVATION ET JUSTIFICATION D'UTILISATION D'UN CAPTEUR VIDEO .....	18
II.6. LE CAS PARTICULIER DES FEUX DE CIRCULATION .....	19
II.7. SYSTEME DE CONTROLE OU DE REGULATION DU TRAFIC EXISTANT .....	20
II.7.1. Système de contrôle Manuel [28] .....	20
II.7.2. Système de contrôle à temps fixe (Système de contrôle traditionnel) .....	20
II.7.3. Système de contrôle à temps dynamique .....	21
II.7.4. Système de signalisation utilisant le traitement d'image.....	21
II.8. GESTION DU TEMPS DE FEUX VERT [1].....	22
II.9. CONCLUSION PARTIELLE .....	23
<b>CHAPITRE III. CONCEPTION DU SYSTEME .....</b>	<b>24</b>
III.1. ARCHITECTURE DU SYSTEME.....	24
III.2. REGULATION DES FEUX DE CIRCULATION .....	25
III.2.1. Détection du véhicule .....	25
III.2.2. Processus décisionnel pour le contrôle des feux de signalisation.....	26
III.3. GESTION DES VEHICULES DE PRIORITE .....	28
III.3.1. Mesures de sécurité .....	30
III.4. GESTION D'INFRACTIONS AU NIVEAU DES FEUX DE CIRCULATION.....	31
III.5. GESTION D'INFRACTIONS AU NIVEAU DE LA VOIE DE CIRCULATION .....	32
III.6. DETECTION D'ACCIDENT AU NIVEAU DE LA VOIE DE CIRCULATION.....	35
<b>CHAPITRE IV: ANALYSE FONCTIONNELLE ET MODELISATION DU SYSTEME .....</b>	<b>36</b>
IV.1. ANALYSE FONCTIONNELLE DU SYSTEME .....	36
IV.1.1. Les différents éléments graphiques de la méthode SA--RT [36] .....	36
IV.1.2. Diagramme de contexte.....	37
IV.1.3. Diagramme préliminaire .....	37
IV.1.4. Diagramme de cas d'utilisation .....	38
IV.1.5. Documentation des UCs (User Cases): .....	39
IV.2. MODELISATION DU SYSTEME.....	41
IV.2.1. Les syntaxes graphiques de la méthode DARTS [36].....	42
IV.2.2. Description du diagramme d'activité du cas d'utilisation « Réguler Feu Circulation »..	44
IV.2.3. Description du diagramme d'activité du cas d'utilisation « Détecter Véhicule de Priorité » .....	45
IV.2.4. Description du diagramme d'activité du cas d'utilisation « Détecter Infraction» .....	46

IV.2.5. Description du diagramme d'activité du cas d'utilisation « <i>Détecter Accident</i> » .....	47
IV.2.6. Diagramme Multitâche DART du système .....	48
IV.3. CONCLUSION PARTIELLE.....	49
<b>CHAPITRE V. EXPERIMENTATION ET ESTIMATION DU COUT DU PROJET .....</b>	<b>50</b>
V.1. EXPERIMENTATION .....	50
V.1.1. Préparation du CPU:.....	50
V.1.2. DataSet (Base d'apprentissage).....	51
V.1.3. Fixation de la caméra (positionnement de la caméra) .....	52
V.2. SCHEMA ELECTRONIQUE DU PROJET .....	54
V.3. QUELQUES RESULTATS: .....	55
V.4. ESTIMATION DU COUT DU PROJET .....	58
V.5. CONCLUSION PARTIELLE .....	64
<b>CHAP VI. CONCLUSION GENERALE ET PERSPECTIVES.....</b>	<b>65</b>
<b>BIBLIOGRAPHIE .....</b>	<b>68</b>
<b>ANNEXE A .....</b>	<b>71</b>
<b>ANNEXE B .....</b>	<b>76</b>
<b>ANNEXE C.....</b>	<b>80</b>

# SIGLES ET ABBREVIATIONS

CG	: Chien de Garde
COCOMO	: Constructive Cost Model
CPU	: Unité Centrale de Traitement
DARTS	: Design Approach for Real-Time Systems
FPS	: Frame Per Second
GSM	: Global System for Mobile SMS: Short Message System
HM	: Homme Mois
HTR	: Horloge Temps Réel
IDE	: Environnement de Développement Intégré
IR	: Infrarouge
IT	: Interruption
ITS	: Intelligent Transport System
KLOC	: 1000 lignes de code
OS	: Operating System
RAM	: Random Access Memory
RFID	: Radio Frequency Identification
SA-RT	: Structured Analysis-Real Time
SMS	: Short Message Service
SCSTR	: Système de Contrôle et Surveillance du Trafic Routier
TDEV	: Temps de Développement
UC	: User Case
UML	: Unified Modeling Language
USD	: United States Dollard

# LISTE DES FIGURES

<i>Figure I. 1 : Carte arduino MEGA 2560 Rev3 [20]</i> .....	11
<i>Figure II. 1 Illustration d'un carrefour Isolé [21]</i> .....	13
<i>Figure II.2: Diagrammes débit- densité [6]</i> .....	15
<i>Figure II. 3: Illustration des différents temps de feu vert pour une phase [1]</i> .....	22
<i>Figure IV.1: Diagramme de contexte générique du système</i> .....	37
<i>Figure III.1: Architecture du Système</i> .....	24
<i>Figure III.2: Schéma global du processus de détection du système proposé</i> .....	26
<i>Figure III.3: Gestion de véhicule de Priorité</i> .....	29
<i>Figure III.4: Gestion d'Infraction au niveau de feux de Circulation</i> .....	31
<i>Figure III.5: Schéma du système d'acquisition de données</i> .....	32
<i>Figure III. 6: Schéma du processus de détection de la vitesse d'un mobile</i> .....	34
<i>Figure III.7: Schéma de configuration pour la déduction de la vitesse du Mobile</i> .....	34
<i>Figure IV. 2: Diagramme préliminaire du système par la méthode SA-RT</i> .....	38
<i>Figure IV. 3: Diagramme de cas d'utilisation du système</i> .....	39
<i>Figure IV.4: Représentation des tâches</i> .....	42
<i>Figure IV. 5 : Communication entre taches</i> .....	42
<i>Figure IV. 6: Activation des tâches</i> .....	43
<i>Figure IV. 7: Module de donnée</i> .....	43
<i>Figure IV. 8: Description du diagramme d'activité du cas d'utilisation « Réguler Feu Circulation »</i> .....	44
<i>Figure IV. 9: Description du diagramme d'activité du cas d'utilisation « Détecter Véhicule de Priorité »</i> .....	45
<i>Figure IV. 10: Description du diagramme d'activité du cas d'utilisation « Détecter Infraction»</i> .....	46
<i>Figure IV.11: Description du diagramme d'activité du cas d'utilisation « Détecter Accident»</i> .....	47
<i>Figure IV.12: Diagramme multitâche DARTS du système</i> .....	48
<i>Figure V.1: quelques images positives</i> .....	51
<i>Figure V.2: quelques images négatives</i> .....	51
<i>Figure V. 3: Fixation de la caméra</i> .....	52
<i>Figure V. 4: Estimation du champ de vision de la caméra</i> .....	53
<i>Figure V. 5: schéma électronique du projet</i> .....	54
<i>Figure V.6: Image temps réel du trafic avant détection</i> .....	55
<i>Figure V.7: Image temps réel du trafic avant après détection</i> .....	55
<i>Figure V. 8: Output console du système de détection de véhicule</i> .....	56
<i>Figure V.9: Image du trafic après détection d'un accident</i> .....	56
<i>Figure V. 10: Output console du système après détection d'un accident</i> .....	57
<i>Figure V. 11: image du trafic pendant la détection de la vitesse</i> .....	57
<i>Figure V. 12: Output console du système pendant la détection de la vitesse</i> .....	58
<i>Figure ANNEXE-A. 1: Caractéristiques pseudo-har à seulement deux caractéristiques</i> .....	72
<i>Figure ANNEXE-A.:2: caractéristiques pseudo-haar avec différentes orientations</i> .....	74

## **LISTE DES TABLEAUX**

Tableau III.1: Processus décisionnel pour le contrôle le feu de signalisation.....	27
Tableau V.2: Liste des matériels .....	58
Tableau V.3: Découpage en Module.....	59
Tableau V.4: Dépôts internes .....	59
Tableau V.5: entrés externes .....	60
Tableau V.6: Comptage des points de fonction non ajusté .....	60
Tableau V.7: fonctions d'ajustement.....	60
Tableau V.8: Conversion de points de fonction aux lignes de code.....	61
Tableau V.9: Estimation COCOMO de base .....	62
Tableau V.10: Distribution de l'effort par phase en pourcentage .....	63

# RESUME

De nos jours, de nombreux travaux de recherche à travers le monde portent sur le transport routier. La gestion du trafic en milieu urbain nécessite le déploiement de solutions et techniques innovantes pour améliorer la fluidité des déplacements et la qualité environnementale des transports des personnes et marchandises.

Cependant, malgré les nombreuses recherches dans le domaine de la vision par ordinateur, l'analyse et la compréhension automatique du trafic reste un défi. C'est pour cela qu'un capteur vidéo doit être équipé de toutes les données nécessaires à la réalisation de nombreuses fonctions (telle que celle du comptage de véhicule, de la détection d'un incident ainsi que celle de la détection du dépassement de la vitesse limite); ce capteur vidéo est à la fois très flexible et économiquement intéressant.

Dans ce travail nous proposons un système de contrôle et surveillance du trafic routier urbain par utilisation d'un capteur vidéo intégrant différents sous-systèmes notamment:

- ✓ un système qui utilise la technique de vision par ordinateur pour réguler le feu de circulation d'une manière intelligente, adaptative et efficace en prenant des images en temps réelle de l'état du trafic à une jonction.
- ✓ un système de contrôle de feu de circulation intégrant la gestion de véhicules d'urgence ou prioritaires par utilisation de la technologie GSM,
- ✓ un système de notification et de la détection du dépassement de la vitesse limite ainsi que de la détection des accidents observés au niveau de la voie de circulation,
- ✓ un système de détection d'infractions observées au niveau du feu de circulation et sur la voie de circulation,

Dans ce système proposé, tout le processus de traitement d'image est effectué par une unité centrale de traitement et la logique de régulation de feux de circulation est implémentée au niveau du microcontrôleur. L'unité centrale de traitement analyse les données du trafic en temps réelle, prend une décision et envoie les résultats au microcontrôleur (ATMega2560) et par la suite le microcontrôleur ajuste le feu de circulation en fonction de cela.

Mot clés : Arduino, SA-RT, DARTS, UML, openCv, vision par ordinateur, capteurs à ultrasons HC-SR04, SMS.

# ABSTRACT

Today, there is a lot of research around the world on road transport. Traffic management in urban areas requires a deployment of innovative solutions and techniques to improve the travel fluidity and the environmental quality of the people and goods transport.

However, despite of the many researches in the field for computer vision, analysis and automatic understanding of traffic remains a challenge. For this reason, a video sensor must be equipped with all the data necessary to perform many functions (such as vehicle counting, incident detection and over range speed limit detection); this video sensor is both very flexible and economically interesting.

In this work we propose a control and surveillance system of the urban road traffic by use of a video sensor integrating different subsystems in particular:

- ✓ a system that uses the computer vision technique to regulate the traffic light with intelligent, adaptive and efficient manner by taking real-time images of traffic status at a junction.
- ✓ a traffic light control system incorporating the management of emergency or priority vehicles using GSM technology,
- ✓ a notification and a system detection of the exceeding speed limit as well as accidents detection observed on the road way,
- ✓ a system for detecting infringements observed at the traffic light and on the road way,

In this proposed system, the entire image processing process is performed by a central processing unit and the traffic light logic control is implemented on the microcontroller (ATMega2560) level. The central processing unit analyses the traffic data in real time, takes a decision and sends the results to the microcontroller and then according to that the microcontroller adjusts the traffic light.

Key words: Arduino, SA-RT, DARTS, UML, openCv, computer vision, ultrasonic sensor HC-SR04, SMS.

# CHAPITRE 0: INTRODUCTION GENERALE

## 0.1. PROBLEMATIQUE

De nos jours, se déplacer est devenu un aspect essentiel de la vie quotidienne : qu'il s'agisse de transports en commun ou de véhicules personnels, le vaste réseau formé de ces moyens de locomotion est immensément complexe à gérer. Sa gestion recouvre l'ensemble des techniques humaines et automatisées permettant d'assurer la surveillance, l'analyse et le contrôle des transports, au mieux un gain de performance dans l'acheminement des différents flux.

La gestion du trafic routier, surtout en milieu urbain, est un aspect essentiel de la vie quotidienne; elle est même au cœur de nombreuses problématiques. En effet, il y a de plus en plus des véhicules, de personnes, mais avec le même espace à partager et peu de budget pour aménager les infrastructures. Jouer sur les leviers technologiques pour rendre l'existant plus efficace est donc nécessaire [1].

Cette gestion du trafic routier s'est amplifié en l'espace de quelques années, engendrant de nombreux problèmes qui coûtent quotidiennement en temps, argent, santé et qualité environnementale, que ce soit par l'intermédiaire des embouteillages, des tracasseries, des accidents ou encore des infractions.

Il y a une chose qui pose problème dans toutes les villes du monde : la circulation ; or on ne peut pas régler ce problème en augmentant la taille ou le nombre de routes, par manque de place.

Pour la gestion du trafic routier, les problèmes de circulation sont généralement traités au niveau d'un seul véhicule ou d'un sous-système (par exemple, dans un couloir spécifique artériel ou une partie d'une route urbaine). Or cette vue parcellaire est inefficace pour améliorer la mobilité [2]. Il est donc nécessaire d'avoir recours à l'automatisation pour prendre en charge les fonctions évoluées maintenant disponibles. A cela se rajoute un trafic plus intense et un environnement plus dense. Il faut alors pouvoir compter sur des systèmes rapides et robustes pour prolonger la perception humaine.

Dès lors, le développement d'un système automatique de contrôle, d'analyse et de surveillance du trafic routier nous impose toutefois d'acquérir des informations en temps-réel sur l'infrastructure routière. C'est pour cela que nous sommes en droit de nous poser différentes questions fondamentales:

- *Comment utiliser la technologie pour rendre le trafic plus fluide ?*
- *Quels mécanismes beaucoup plus économiques et efficaces pouvons-nous proposer afin de pouvoir bien assurer le contrôle, l'analyse, la surveillance et rendre certaines composantes du système autonomes ?*

Dans ce travail, nous nous intéressons à l'utilisation des technologies de la vision par ordinateur et d'un réseau fixe de capteurs vidéo afin d'améliorer la surveillance, l'analyse et le contrôle du trafic routier dans un environnement urbain. En effet, le cadre urbain est celui qui engendre le plus de problèmes (ralentissements, engorgements, etc.), mais est aussi le plus complexe. En outre, il offre la plus grande possibilité de surveillance, d'analyse et de contrôle par un système de transport intelligent.

## 0.2. HYPOTHESES

Comme nous l'avions mentionné dans la problématique, nous pensons que les embouteillages, la pollution, les tracasseries, ou encore les accidents seraient, à priori réglables si un meilleur contrôle leur était réservé. Ceci nous laisse supposer que:

- ✓ une meilleure gestion des feux de circulations routières intelligentes nous permettrait d'améliorer l'expérience des usagers et de réduire les coûts globaux.
- ✓ La nécessité de la fourniture de l'état du trafic en temps réel contribuerait à la surveillance et au contrôle du trafic routier.

## 0.3. OBJECTIFS ET DELIMITATION DU TRAVAIL

L'un des principaux enjeux de l'information routière est la surveillance, l'analyse, le contrôle et la gestion de la formation des phénomènes de congestion (bouchons), qu'ils soient occasionnels ou récurrents.

*D'une manière spécifique, notre travail se propose de pouvoir :*

- ✓ Visualiser le trafic routier ;
- ✓ Compléter et étendre les moyens existants ;
- ✓ Proposer une méthode de détection du dépassement de la vitesse limite d'un véhicule par vision d'ordinateur ;

- ✓ Proposer une méthode de détection d'infraction et d'accident observé sur la voie de circulation ;
- ✓ Augmenter les fréquences de surveillance qui permettrait notamment de donner des informations en quasi-temps réel.

*D'une manière générale, notre travail se propose de pouvoir :*

- ✓ parvenir à optimiser et gérer les phénomènes de congestion par le moyen d'un feu de signalisation intelligent;
- ✓ Réduire les coûts financiers ;

D'une manière beaucoup plus simple, l'objectif du présent travail est de pouvoir développer un système temps réel d'aide à l'analyse, au contrôle et à la surveillance du trafic routier. Le système développé doit réagir en fonction de l'environnement à partir de la vision par ordinateur et de l'utilisation des quelques capteurs.

#### 0.4. METHODES ET TECHNIQUES DE TRAVAIL

Pour un meilleur aboutissement aux objectifs de notre travail nous nous sommes servi de quelques techniques et méthodes de recherche scientifique dont notamment :

***Méthodes:***

- Le *processus unifié* « Unified Process » à l'aide du langage UML pour la modélisation du système,
- La gestion du projet et son estimation en cout et durée avec *COCOMO*,
- *Viola et Jones* pour la détection d'objet dans une image numérique,
- *SA-RT et DARTS* pour l'analyse et la conception du système,
- *La soustraction de l'arrière-plan* est l'opération qui suit logiquement la modélisation de l'arrière-plan afin d'obtenir une détection de mouvement.

### ***Techniques:***

- *L'étude documentaire* approfondie sur l'implémentation des quelques algorithmes de vision par ordinateur
- *Les simulations et les tests*
- *L'observation* qui nous a permis de constater les faits tel qui se déroulent et ceci sans pour autant modifier les phénomènes. Etant un moyen pour arriver à la connaissance du milieu, l'observation nous servira de sens et la conscience à un objet afin d'acquérir une connaissance claire et précise.
- *L'analyse systématique* de toutes les informations ainsi que les données récoltées.

## 0.5. INTERET ET SUBDIVISION DU TRAVAIL

Ce travail a un double intérêt :

Premièrement, du point de vue personnelle, notre centre d'intérêt est de pouvoir nous familiariser et d'enrichir nos connaissances dans la combinaison du développement d'un système d'acquisition de données en temps réel par vision d'ordinateur et d'un système embarqué à travers une mise en situation pratique de l'objet de ce présent travail.

Deuxièmement, du point de vue communautaire l'intérêt est de pouvoir inciter la conscience de la communauté à pouvoir respecter le code de la route, soulager et renfoncer les tâches de la police de circulation routière, contribuer aussi à l'accroissement des recettes de la trésorière publique par l'enregistrement et l'archivage des différentes infractions observées sur la voie de circulation routière.

Outre l'introduction et la conclusion, notre travail s'articule sur cinq chapitres :

CHAPITRE I: ETAT DE L'ART ET CONCEPTS DE BASE

CHAPITRE II: GENERALITE SUR LE TRAFIC ROUTIER URBAIN

CHAPITRE III: CONCEPTION DU SYSTEME

CHAPITRE IV: ANALYSE FONCTIONNELLE ET MODELISATION DU SYSTEME

CHAPITRE V. EXPERIMENTATION ET ESTIMATION DU COUP DU PROJET

# CHAPITRE I: ETAT DE L'ART ET CONCEPTS DE BASE

## I.1. ETAT DE L'ART

De nos jours, de nombreux travaux de recherche à travers le monde portent sur le transport routier. La motivation vient du fait que ce secteur nécessite de nouvelles applications de méthodes permettant de contrecarrer l'augmentation accrue des problèmes issus du trafic routier, que ce soit, en vue de réduire le nombre d'accidents, de diminuer le trafic dans les grandes agglomérations ou de réduire la pollution.

En raison de l'importance du transport et du trafic routier, plusieurs travaux ont presque tous convergé vers des solutions basées sur les systèmes de transport intelligents («*Intelligent Transport System*», ITS).

Dans [3], les auteurs présentent un système de suivi d'images et ses applications pour la surveillance du trafic et la détection des accidents aux intersections routières. Les emplacements des motos ainsi que des automobiles sont obtenus en temps réel en utilisant l'approche du modèle de contour actif. Le traitement d'image est en outre incorporé aux techniques de filtrage de Kalman pour suivre le mouvement de chaque véhicule.

Dans [4], les auteurs proposent un algorithme adaptatif de contrôle de feux de circulation pour ajuster la séquence et la durée des feux de signalisation en fonction de l'état du trafic détecté en temps réel. Leur algorithme prend en considération un certain nombre de facteurs de trafic tels que le volume de trafic, le temps d'attente, la densité des véhicules pour déterminer la séquence et la durée optimale du feu vert.

Dans [5], l'auteur propose un système de contrôle du trafic routier pour contrôler automatiquement le feu de signalisation ; la détermination du nombre de véhicule se fait en utilisant un approche basée sur un ensemble des capteurs IR (Infrarouge) placés au bord de la route et d'un module RFID (Radio Frequency Identification) pour pouvoir détecter la présence d'un véhicules d'urgence. Cependant, compte tenu du volume de

trafic en temps réel et de la capacité des capteurs IR, ce système ne peut pas être considéré comme très fiable.

Dans [6], les auteurs ont proposé un système permettant de gérer deux problèmes de circulation routière : détection de véhicules volés et changement du feu de circulation en vert pour le véhicule d'urgence. Différents composants matériels ont été utilisés, tels que ZigBee, RFID incorporé dans chaque véhicule. Lorsque le véhicule d'urgence se retrouve à la jonction de la circulation, le module émetteur ZigBee du véhicule envoie un signal au récepteur ZigBee du module de contrôle de feu de circulation pour que le feu de circulation change en vert. En outre, le RFID est utilisé pour détecter les véhicules volés.

## I.2. NOTRE APPROCHE

Après avoir analysé les quelques travaux mentionnés dans l'état de l'art, nous constatons que la plus part de solutions proposées par les auteurs nécessitent d'utiliser beaucoup de capteurs sur la voie de circulation, d'intégrer dans le véhicule certains modules tel que : le RFID, ZigBee,... Le système est donc dépendant des différents dispositifs matériels intégrés dans des véhicules ou sur la voie de circulation.

Notre approche vise à pouvoir rendre le système indépendant des différents dispositifs matériels intégrés dans le véhicule, réduire considérablement la quantité des capteurs intégrés sur la voie de circulation tout en utilisant des méthodes et de technologies disponibles et moins coûteuses en ressource dont notamment le GSM, les capteurs vidéo (la vision par ordinateur) et quelques capteurs à ultrasons HC-SR04 pour automatiser la gestion du trafic routier (feux de circulation, véhicule d'urgence, détection du dépassement de la vitesse limite du véhicule, détection d'infraction, détection d'accident).

## I.3. CONCEPTS

Dans cette partie nous allons présenter quelques notions et outils d'analyse qui seront ensuite utilisés dans notre système de contrôle et de surveillance du trafic routier urbain.

**Les systèmes de transport intelligents (STI)** apparaissent comme étant "*l'application des technologies de l'information et de la communication au domaine des transports*". Le terme système est vague et se décline en un ensemble de moyens mis en place pour gérer au mieux les

contraintes liées au trafic routier, telles que les embouteillages, la sécurité ou même la pollution [7]. En particulier, ces systèmes offrent un caractère réactif à une infrastructure pourtant fixe, mais dont la population peut grandement varier.

Le *trafic* est constitué de deux composantes majeures [8]: **l'infrastructure et les mobiles**.

L'*infrastructure* est composée, entre autres, de trottoirs, de bâtiments et de mobiliers urbains (bancs publics, arrêts de bus, etc.) qui sont généralement négligés par la modélisation du trafic. En revanche, la chaussée et les signalisations, qui en font partie, sont les principaux éléments considérés.

Un *mobile* est défini comme étant un usager qui conduit son véhicule au sein de l'infrastructure. Ils sont au cœur de notre considération. Par contre, les piétons (vaquant à pied) ne sont pas considérés car leurs mouvements sont supposés dépendre de celui des mobiles.

Les **Carrefour** sont les lieux de croisement d'au moins deux routes. Un tel croisement peut revêtir de nombreuses formes, tant dans l'infrastructure que dans les règles de franchissement. Une **route** est caractérisée par sa longueur, son nombre de voies, ainsi que par le sens de circulation et la vitesse maximale autorisée sur la voie [9].

**Modélisation de l'arrière-plan** (Background modeling) : cette catégorie regroupe toutes les méthodes de détection de mouvement qui consistent à créer un modèle de l'arrière-plan de la scène filmée (sans aucun objet mobile). Ce modèle peut être une image créée à partir des pixels observées à différents instants de la séquence vidéo, comme dans [10], un modèle statistique décrivant la fonction de distribution des niveaux de gris ou des couleurs en tout point (c'est de loin l'approche la plus fréquente., depuis [11] ), ou encore une base d'images caractéristiques qui constitue un sous-espace vectoriel dans lequel on considère que les pixels représentant l'arrière-plan vont se trouver [12].

**Soustraction de l'arrière-plan (Background subtraction)** : La soustraction de l'arrière-plan est l'opération qui suit logiquement la modélisation de l'arrière-plan afin d'obtenir une détection de mouvement. Si le modèle de l'arrière-plan est une image, une différence en valeur absolue entre ce modèle et l'image courante est effectuée afin d'obtenir une détection de mouvement.

Quand il s'agit d'un modèle statistique, on calcule la probabilité que chaque pixel appartienne à l'arrière-plan en testant la valeur observée dans le modèle; l'importance du mouvement observé varie dans le sens opposé à la probabilité calculé [13].

**Vision par ordinateur** : (aussi appelée vision artificielle, vision numérique ou plus récemment vision cognitive) est une branche de l'intelligence artificielle dont le but est de permettre à une machine de comprendre ce qu'elle «voit » lorsqu'on la connecte à une ou plusieurs caméras [14].

**OpenCV** est une bibliothèque libre de vision par ordinateur. Cette bibliothèque est écrite en C et C++ et peut être utilisée sous Linux, Windows et Mac OS X. Des interfaces ont été développées pour Java, Python, Ruby, Matlab et autre langage. Open CV est orienté vers des applications en temps réel [14].

Le **tracking** est un procédé de localisation d'un (ou plusieurs) objet en mouvement en temps réel en utilisant une caméra. Un algorithme analyse les photogrammes de la vidéo et localise les cibles en mouvement sur la vidéo. La principale difficulté dans le tracking sur une vidéo est d'associer la localisation des cibles dans les photogrammes successifs, particulièrement lorsque les objets bougent rapidement par rapport au frame rate. Les systèmes de tracking sur une vidéo utilisent normalement un modèle en mouvement qui décrit comment l'image de la cible peu changer en tenant compte du mouvement possible de l'objet traqué [14].

Le rôle d'un algorithme de tracking est d'analyser les photogrammes dans l'ordre et d'estimer les paramètres de mouvement. Ces paramètres caractérisent la localisation de la cible.

**Pixel** [15] : Une Image est constituée d'un ensemble de points appelés pixels.

Le mot **pixel** provient d'une abréviation de l'expression britannique **PIC**ture **E**lement. Le pixel représente ainsi le plus petit élément constitutif d'une image numérique. L'ensemble de ces pixels est contenu dans un tableau à deux dimensions constituant l'image finalement obtenu. Etant donné que l'écran effectue un balayage de gauche à droite et de haut en bas, on désigne généralement par les coordonnées [0,0] le pixel situé en haut à gauche de l'image.

**Image** [16] : Une image est avant tout un signal 2D (x, y), qui représente souvent une réalité 3D (x, y, z). D'un point de vue mathématique, une image est une matrice de nombres représentant un signal; plusieurs outils permettent de manipuler ce signal. Une image contient plusieurs informations sémantiques, il faut en interpréter le contenu au-delà de la valeur des nombres.

Il existe trois principaux types d'images :

1. les images en niveaux de gris, dont la valeur appartient à l'ensemble  $\{0,1,\dots,255\}$ .
2. les images binaires (uniquement en noir et blanc) et dont la valeur égale soit 0 soit 1.
3. les images couleurs dont plusieurs représentations existent.

**Détecteur** : Un détecteur est plus globalement une technologie permettant de récolter et transmettre des informations à un nœud traitant [1]. Ce terme englobe donc les capteurs électromagnétiques, mais également les autres technologies existantes et détaillées plus loin dans le document.

**Un contrôleur de feux** est un équipement permettant de veiller au bon fonctionnement des feux de circulation. Il applique une stratégie de signalisation en enclenchant séquentiellement les feux. Il est traditionnellement rattaché à un seul carrefour mais peut, dans certains cas, en gérer plusieurs [1].

**Une LED** (Light-Emitting Diode) est un composant électronique qui émet de la lumière lorsqu'il est parcouru par un courant électrique [17].

**Un capteur à ultrasons** est un dispositif de contrôle industriel qui utilise les ondes sonores supérieures à 20 000 Hz, trop élevées pour être captées par l'oreille humaine, pour mesurer et calculer la distance du capteur à un objet cible spécifié [18].

## I.4. PRESENTATION D'ARDUINO [19].

Tel que l'a toujours dit John NUSSEY, Arduino est un outil, une communauté et une façon de penser qui transforme notre regard sur la technologie et l'usage que nous en faisons. C'est une plateforme et un outil permettant de construire des dispositifs qui peuvent interagir avec l'environnement qui les entoure.

Avant d'aller loin, notons que le nom Arduino désigne un ensemble de choses. Il peut s'agir de la carte Arduino - le matériel - ou de l'environnement Arduino - c'est-à-dire, un logiciel qui tourne sur notre ordinateur – ou encore d'Arduino en tant que sujet de recherche. La carte Arduino est un circuit imprimé, spécifiquement conçu pour héberger un microcontrôleur (d'architecture Atmel AVR comme l'Atmega328p, et d'architecture ARM comme le Cortex-M3 pour l'Arduino Due) et donner accès à toutes ses entrées et sorties.

Elle comprend également quelques autres composants électroniques qui permettent de faire fonctionner le microcontrôleur ou d'en étendre les fonctionnalités. Il n'existe pas de modèle unique de la carte Arduino ; on en trouve une pléthore de variations, chacune étant conçue pour convenir à différents usages. Cependant, il existe plusieurs types des cartes arduino par exemples : arduino UNO, arduino MEGA, arduino DUE, arduino LEONARDO, etc. Toutefois, une carte peut être considérée comme la pierre angulaire de toute aventure dans le monde Arduino, c'est celle qui est adaptée à la plupart des usages. Comme dans ce travail nous avons utilisé une carte arduino MEGA, nous allons nous limiter uniquement à la présentation d'arduino MEGA.

La carte Arduino Mega 2560 est basée sur un ATmega2560 cadencé à 16 MHz. Elle dispose de 54 E/S dont 14 PWM, 16 analogiques et 4 UARTS. Des connecteurs situés sur les bords extérieurs du circuit imprimé permettent d'enficher une série de modules complémentaires. Le contrôleur ATmega2560 contient un bootloader qui permet de modifier le programme sans passer par un programmeur.

Caractéristiques principales:

- version: Rev. 3
- via port USB ou de 7 à 12 V sur connecteur alim
- microprocesseur: ATmega2560
- mémoire flash: 256 kB

- mémoire SRAM: 8 kB
- mémoire EEPROM: 4 kB
- 54 broches d'E/S dont 14 PWM
- 16 entrées analogiques 10 bits
- intensité par E/S: 40 mA
- cadencement: 16 MHz
- 3 ports série
- bus I2C et SPI
- gestion des interruptions
- fiche USB
- dimensions: 107 x 53 x 15 mm

Un exemple d'image d'une carte arduino MEGA est illustré à la figure ci-dessous :



*Figure I.1: Carte arduino MEGA 2560 Rev3 [20]*

## I.5. CONCLUSION PARTIELLE

Dans ce chapitre, nous avons présenté brièvement l'état de l'art en rapport avec les systèmes de transport intelligents («*Intelligent Transport System*», ITS); ensuite nous avons présenté notre approche de gestion du contrôle et de surveillance du trafic routier urbain et par la suite nous avons présenté quelques notions et matérielles utilisés dans notre système de contrôle et de surveillance du trafic routier urbain.

# **CHAPITRE II: GENERALITES SUR LE TRAFIC**

## **ROUTIER URBAIN**

Dans ce chapitre, nous présentons les éléments de base composant un trafic routier urbain. Ensuite, nous introduisons quelques méthodes émergentes et ainsi que quelques notions importantes utilisées pour la régulation du trafic au niveau des carrefours.

### **II.1. INTRODUCTION SUR LE TRAFIC ROUTIER**

Le trafic urbain peut être assimilé à un système dynamique hybride (interactions de phénomènes continus et discrets). L'évolution des files d'attente est décrite par des équations continues et l'occurrence d'évènements (feux de signalisation, incidents,...) par des automates discrets [8].

La gestion du trafic routier s'inscrit dans le domaine des STI (Système de Transport Intelligent), qui visent à proposer des outils et modèles afin de gérer les aléas de ce dernier, ceci par le biais ou non d'équipements réactifs dits dynamiques. L'application de tels systèmes va avoir de multiples objectifs, parmi lesquels la fluidification du trafic, la surveillance temps-réel du trafic, la qualité environnementale des transports de personnes et marchandises, la diffusion d'informations ou de consignes variables aux automobilistes ou encore la réduction en conséquence de la pollution et des bruits.

Des avancées significatives dans les domaines de l'informatique, de l'automatique et de l'instrumentation, permettent de calculer des paramètres de la commande des feux en temps réel à partir de mesures transmises par des boucles électromagnétiques ou des caméras numériques. Ces stratégies de commandes adaptatives régulent l'état des feux en fonction de la présence ou non de véhicules à l'intersection.

Le système qui surveille, contrôle et maintient les feux de circulation d'une manière complètement automatisée est appelée système automatisé de contrôle de la circulation.

## II.2. COMPOSANTES DU TRAFIC ROUTIER

Un réseau routier est conçu afin de permettre à ses usagers de se déplacer d'un point à un autre. Dans ce contexte, il est composé d'un ensemble de routes avec plus ou moins de voies, en fonction des zones d'activités et des lieux d'habitation. Les croisements entre les routes étant inévitables en milieu urbain, de par leur concentration, des intersections permettent de gérer les flux de véhicules entrant en conflit, pour leur sécurité et afin d'éviter les inters blocages et de permettre aux automobilistes de changer de route. Ces intersections, ou dans certains cas des carrefours giratoires, régulent le trafic et appliquent généralement des règles de priorité à droite ou à gauche dans les pays anglais où sont équipés de feux de circulation pour gérer les situations particulièrement dangereuses [1].

Le réseau routier urbain peut donc être vu comme étant une collection d'intersections, liées les unes aux autres par des routes, auxquelles peuvent venir se greffer des éléments opérationnels (panneaux, feux de circulation, etc.).

Comme nous pouvons le voir sur la Figure II.1, un carrefour simple est composé de quatre routes (ou quatre directions). Une route est caractérisée par sa longueur, son nombre de voies. Cette figure montre également les zones fonctionnelles : la « zone de conflit », commune aux deux routes et la « zone de stockage ». On peut aussi définir la « longueur de stockage » comme la longueur sur laquelle se stockent tous les véhicules pouvant pénétrer le carrefour pendant la durée du feu vert.

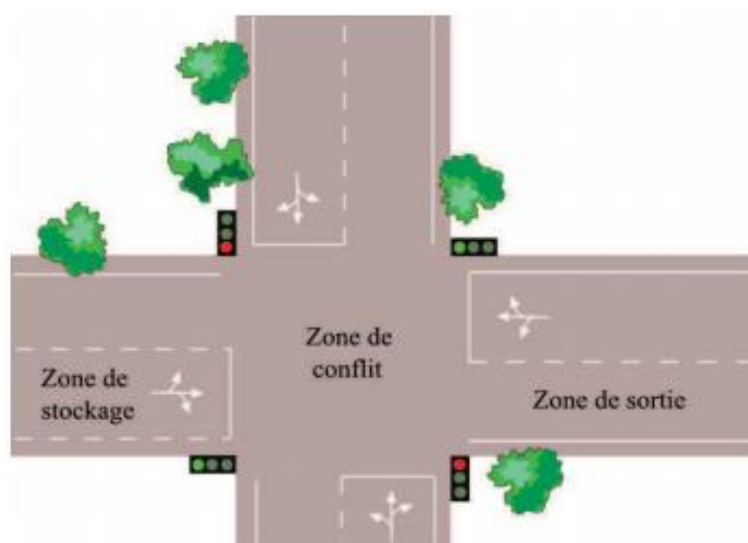


Figure II. 1 Illustration d'un carrefour Isolé [21].

Un « **mouvement** » est défini par son origine et sa destination. Une voie n'est pas allouée à chaque type de mouvement, mais ceux-ci sont au contraire regroupés en flux sur une même voie mixte. Un flux de mobiles est donc l'ensemble des mouvements des véhicules qui proviennent d'une même branche d'accès à un carrefour et qui sont admis simultanément au moyen d'un même groupe de signaux.

Les indications des feux – vert, orange, rouge – se succèdent à l'intérieur d'un « **cycle** », défini comme étant la durée constante séparant deux passages successifs de l'ensemble des signaux par le même état. Le cycle est divisé en « phases », temps pendant lequel un ou plusieurs flux sont admis dans le carrefour.

*Une phase est dite spéciale quand elle a pour seul but de favoriser l'écoulement d'un mouvement tournant. Elle est dite saturée lorsqu' au moins un véhicule de cette phase est contraint d'attendre plus d'un cycle pour franchir le carrefour. Le carrefour est lui-même saturé quand au moins une de ses phases est saturée.*

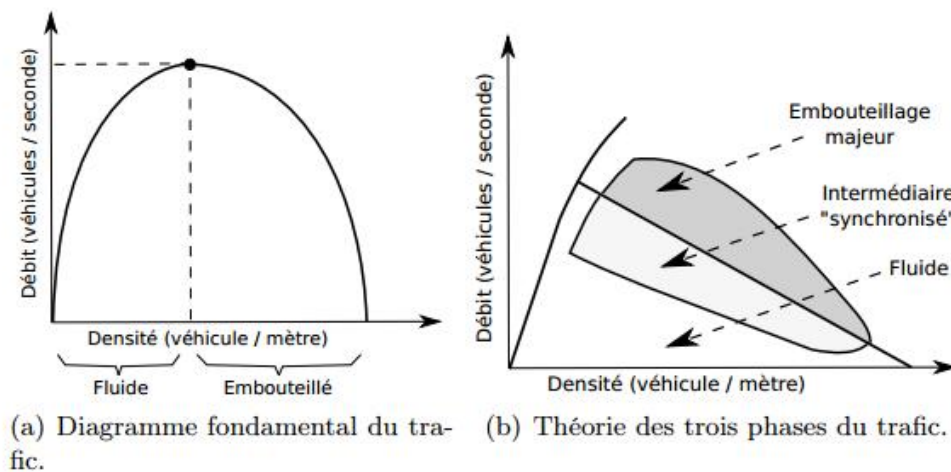
### II.3. LE DIAGRAMME FONDAMENTAL DU TRAFIC

Le diagramme fondamental du trafic permet de décrire la relation entre le débit et la densité d'un réseau routier [22]. On distingue deux états de trafic différents : le cas où le trafic est fluide (c.-à-d., vitesse proche de la vitesse libre) et le cas où il est embouteillé (c.-à-d., les véhicules se déplacent à une vitesse inférieure à la vitesse libre).

La figure II.2(a) représente la forme générale de ce diagramme. Des variantes existent, telles que celle suggérée par la théorie des trois phases du trafic [23].

Elle considère un état supplémentaire d'embouteillage, comme représenté sur la figure II.2(b).

Compte tenu de la relation liant les deux variables (le débit et la densité), la loi d'écoulement du trafic sur une section de route peut donc être formulée par une fonction liant deux des trois variables : débit, concentration et vitesse. Le diagramme fondamental est sensible à de nombreux facteurs comme la géométrie de la route, la nature et la composition du trafic, les conditions météorologiques, les mesures d'exploitation, ...



*Figure II.2: Diagrammes débit- densité [6].*

## II.4 MESURES USUELLES DU TRAFIC

Il existe de nombreux types de capteurs permettant la mesure directe ou indirecte des variables de circulation. Ces capteurs sont généralement des éléments transducteurs, sensibles à la grandeur physique que l'on veut saisir : présence, passage, vitesse d'un véhicule [25].

Le capteur peut changer l'information en un signal élémentaire, transmis au détecteur. Le signal reçu est transformé en une information électrique simple, significative du paramètre de circulation. Compte tenu des évolutions technologiques rapides dans le domaine de la métrologie routière, seuls les principaux types de capteurs d'emploi courant sont ici mentionnés [26], à savoir :

- Les pneumatiques
- Les boucles électromagnétiques
- Les ultrasons
- Le capteur vidéo

### II.4.1. Les capteurs pneumatiques

Ils permettent d'effectuer des comptages routiers et donc de mesurer les débits. Ils sont constitués d'un câble en caoutchouc, tendu en travers de la chaussée et relié à un détecteur. L'écrasement du câble lors du passage d'un véhicule provoque une surpression, détectée par un manomètre actionnant un relais. Il est alors possible de compter de nombre

d'essieux passant sur le capteur en cumulant les impulsions dans un compteur. Les comptages sont par la suite exprimés en u.v.p (unité de voiture particulière): 1 u.v.p. = 2 essieux.

Ces capteurs, encore répandus pour les comptages routiers, présentent quelques avantages, notamment :

- ✓ La facilité de pose
- ✓ La bonne portabilité de l'ensemble capteur-détecteur
- ✓ La possibilité de fonctionnement sur batterie procurant une autonomie de plusieurs jours

En revanche, le système est d'un coût moyen élevé. De plus, le câble peut être arraché lors du passage de véhicules lourds. En régime saturé, l'imprécision peut parfois dépasser les 20%.

#### **II.4.2. Les boucles électromagnétiques**

C'est aujourd'hui le dispositif de mesure des paramètres de circulation le plus répandu dans de nombreux pays, tant en ville que sur les voies rapides et les autoroutes urbaines.

Le capteur est constitué d'une boucle inductive, noyée dans le revêtement de la chaussée. Le passage de la masse métallique d'un véhicule au-dessus de la boucle provoque une variation du champ électromagnétique. Cette variation se traduit par un créneau de tension dont la longueur est liée à celle du véhicule et à son temps de passage.

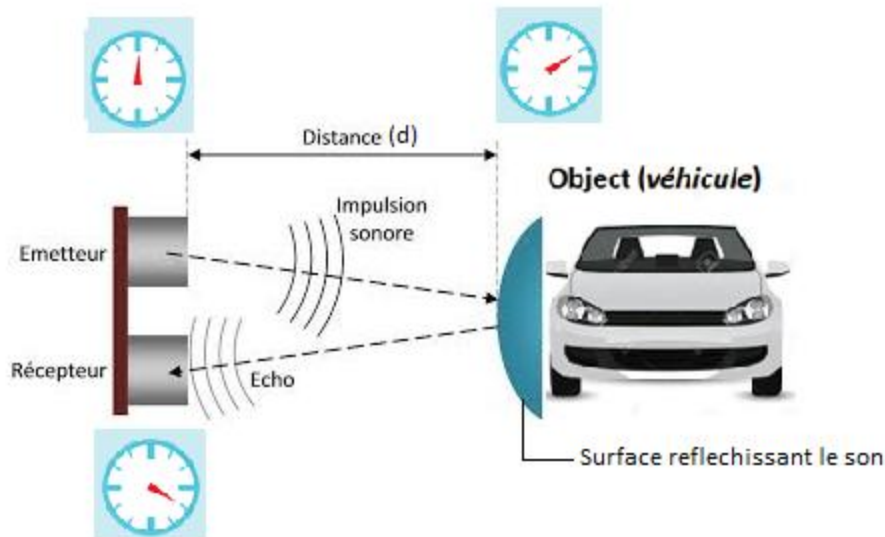
Avec une seule boucle par voie, on mesure bien sûr le débit mais aussi le taux d'occupation  $t$ , défini par :

$$t = \frac{100 \sum_{i=1}^n t_i}{T} \quad (\text{II.1})$$

Où  $t_i$  désigne le temps d'occupation de la boucle à la période de mesure  $i$ , et  $T$  le temps total de mesure.

#### **II.4.3. Principe de fonctionnement d'un capteur à ultrason HC-SR04 :**

Le principe de fonctionnement d'un capteur à ultrason HC-SR04 repose comme son nom l'indique sur l'utilisation des ultrasons. Ce sont des ondes acoustiques dont la fréquence est trop élevée pour être audible par l'être humain. Dans ce travail, on mesurera la présence d'un véhicule grâce au capteur ultrasonique HC-SR04 et son principe de fonctionnement est illustré à la figure II.3:



**Figure II.3: Principe de fonctionnement d'un capteur à ultrasons HC-SR04**

Un capteur à ultrasons HC-SR04 émet à intervalles réguliers de courtes impulsions sonores à haute fréquence. Ces impulsions se propagent dans l'air à la vitesse du son. Lorsqu'elles rencontrent un objet (véhicule), elles se réfléchissent et reviennent sous forme d'écho au capteur. Celui-ci calcule alors la distance le séparant de la cible sur la base du temps écoulé entre l'émission du signal et la réception de l'écho; la distance étant déterminée par le temps de propagation des ultrasons et non par leur intensité. Ainsi, pour déterminer la distance  $d$ , on utilise l'équation :  $v = \frac{d}{\frac{t}{2}} \Rightarrow d = \frac{v \times t}{2}$  (II.2)

Où :

- $t$  étant le temps qui s'écoule entre le début de l'émission et le début de la réception
- et  $v$  étant la vitesse du son qui est de 340 m.s<sup>-1</sup>.

Caractéristiques technique d'un capteur à ultrason HC-SR04:

- Mesure de distance : entre 2 centimètres et 3,3 mètres,
- Erreur moyenne inférieure à 0,5 centimètres,
- Fréquence d'émission : 40 KHz,
- Tension d'alimentation : 5 V en DC
- Curent d'alimentation : 15 mA

#### II.4.4. Le capteur vidéo

Ce domaine fait aujourd'hui l'objet de nombreuses recherches, particulièrement dans le cadre européen. Le principe de ce capteur consiste à utiliser une caméra vidéo et à traiter de manière automatique les images fournies par cette caméra afin d'en déduire les paramètres du trafic. Dans la suite de ce présent travail, nous allons nous focaliser sur ce capteur vidéo.

#### II.5. MOTIVATION ET JUSTIFICATION D'UTILISATION D'UN CAPTEUR VIDEO

Le choix du (ou des) capteur(s) qu'il convient d'utiliser pour percevoir l'environnement est une étape cruciale dans la réalisation d'un système de contrôle et surveillance du trafic routier. La sélection doit prendre en compte le domaine de fonctionnement de chaque capteur et ses performances.

On distingue généralement deux catégories de capteurs: les *capteurs intrusifs* (qui sont installés dans la chaussée) et les *capteurs non-intrusifs* (installés aux abords des routes, en portique ou borne centrale ou encore embarqué dans des véhicules).

Les capteurs intrusifs sont la plupart du temps des boucles inductives, placées dans chaque voie de circulation, qui vont réagir au passage des véhicules.

Cependant, l'inconvénient majeur des boucles inductives est la nécessité d'intervenir dans la route (et donc de perturber le trafic) pour l'installation et la maintenance, nécessitent une durée d'installation beaucoup plus longue comparativement à un *capteur non-intrusif* (camera).

De plus, les boucles inductives sont limitées à des fonctions de comptage, de mesure et de classification individuelle par bande de circulation. Les capteurs non-intrusifs sont des senseurs distants qui utilisent diverses technologies: laser, ultrasons ou vidéo.

Cependant, la caméra est devenue aujourd'hui un capteur fiable et très peu coûteux, elle fournit une représentation (images) directement exploitable par l'être humain, car elle fournit un signal proche de celui perçu par le cerveau et surtout l'œil, qui en est un de ces capteurs privilégiés, de plus les systèmes basés sur la vision par ordinateur ont démontré leur efficacité pour des applications d'aide à la conduite. Voilà pourquoi un système basé sur la

vision par ordinateur me semble logique si on envisage d'implémenter une technique à la fois rapide, flexible et moins chère.

## II.6. LE CAS PARTICULIER DES FEUX DE CIRCULATION

L'intérêt des feux de circulation en milieu urbain n'est plus à démontrer. En premier lieu, ils servent à la sécurité des utilisateurs, sur des carrefours (ou intersections dans la suite du document) subissant une charge suffisamment importante pour être considéré dangereux. Sans ces systèmes, si plusieurs flux de véhicules importants venaient à se croiser sur une intersection, le comportement individualiste des utilisateurs rendrait la traversée des carrefours périlleuse et augmenterait le temps perdu.

La mise en place des séquences temporisées permettant de régir le passage des véhicules

Le premier feu de circulation aurait ainsi été installé à Londres le 10 décembre 1868, sous la forme d'une lanterne à gaz pivotante aux couleurs rouge et verte. Pour fonctionner, il était nécessaire qu'un agent de police soit présent pour la manœuvrer. Aujourd'hui, et dans de nombreux pays, il est courant de voir des feux de circulation dont les ampoules sont remplacées par des diodes électroluminescentes et qui affichent le décompte des secondes restant avant un changement d'état [1].

L'application des nouvelles technologies de l'information et de la communication à ce type d'équipement représente une réelle opportunité. Les séquences habituellement utilisées ne sont pas toujours en adéquation avec la nature temps-réel du trafic routier.

De plus, la présence de plusieurs feux de circulation successifs peut rapidement devenir inefficace si l'automobiliste doit s'y arrêter à chaque fois. Comme le montre [27], la gestion intelligente des feux de circulation est au cœur de nombreuses problématiques STI et est capable de fluidifier le trafic routier, en plus de servir indirectement l'environnement.

## II.7. SYSTEME DE CONTROLE OU DE REGULATION DU TRAFIC EXISTANT

Dans cette partie, nous présentons différents système existant utilisé pour contrôler le trafic et leurs inconvénients sont discutés ci-dessous:

### II.7.1. Système de contrôle Manuel [28]

Le contrôle manuel, comme son nom l'indique, nécessite une puissance humaine pour contrôler le trafic. Les politiques de circulation sont attribuées en fonction des pays et des états pour contrôler une zone requise ou un trafic urbain. La police de circulation utilise différents outils tels que: le panneau de signalisation, le signe lumineux et de sifflet pour contrôler le trafic.

Ils sont donc invités à porter des uniformes spécifiques afin de contrôler le trafic.

Dans le système de commande manuelle, plus d'énergie est nécessaire et parfois il n'est pas possible de contrôler le trafic manuellement dans toutes les zones d'une ville.

Parmi les problèmes rencontrés dans le cas du contrôle Manuel par l'humain, nous pouvons citer:

- Seuls les opérateurs qualifiés peuvent faire des jugements et des décisions appropriés, car la situation est très compliquée et de nombreux facteurs doivent être pris en compte dans le contrôle;
- La charge de travail des opérateurs qualifiés est très élevée, car ils prennent toujours des décisions selon les conditions de circulation à des intervalles de temps très courts;
- Il est très difficile d'améliorer le processus de contrôle de la circulation, car le processus réel du jugement des opérateurs n'est pas clairement décrit.

### II.7.2. Système de contrôle à temps fixe (Système de contrôle traditionnel)

En fonction de l'heure et parfois du jour, le système va utiliser un plan de feux prédéfini. Par exemple : une configuration stricte est appliquée de 12h30 à 14h, tandis qu'une plus souple et équitable pour l'ensemble des voies d'un carrefour est appliquée à 3h du matin. Classiquement, trois configurations existent : matin, après-midi, et le reste (hors pointes) [29]. Nous pouvons imaginer une formulation plus complète prenant en compte des plans spécifiques aux heures de pointe.

Le principal inconvénient est que l'algorithme de contrôle ne vise que les véhicules sur le feu vert sans tenir compte du nombre de véhicules en attente au niveau des feux rouges.

### **II.7.3. Système de contrôle à temps dynamique**

Le feu de circulation automatique est contrôlé par des minuteries et des capteurs électriques.

Dans un système de feux de circulation traditionnel, chaque phase a une valeur numérique constante chargée dans la minuterie. Les lumières s'allument automatiquement en fonction de la valeur de la minuterie.

Étant donné que le système de feux de circulation traditionnel est très propice à la congestion du trafic, un système de feux de circulation intelligent est nécessaire, c'est-à-dire au lieu d'une minuterie définie, la densité des véhicules est calculée, puis la minuterie est réglée.

Le mécanisme le plus commun est l'utilisation de capteurs électriques intégrés dans le pavé (la voie de circulation). Le système de contrôle de feux de circulation détecte les véhicules et produit un signal de la minuterie en fonction de la densité des véhicules.

*L'utilisation des capteurs électroniques n'est pas si efficace car elle a une forte tendance à être affectée par les bruits et coûte cher. Tous ces inconvénients sont censés être éliminés par l'utilisation d'un capteur Vidéo.*

### **II.7.4. Système de signalisation utilisant le traitement d'image**

Dans un Système de contrôle du trafic par traitement d'image, une caméra est installée à côté de feu de signalisation, capture les séquences d'image, et ensuite analyse ces images. Le système détecte les véhicules à travers des images au lieu d'utiliser des capteurs électroniques intégrés dans le trottoir.

*Le traitement d'image est une meilleure technique pour contrôler le changement d'état de feu de circulation ainsi que du trafic. Cela montre qu'il peut réduire la congestion du trafic et éviter que le temps ne soit gaspillé par un feu vert sur une route vide. Il est également plus cohérent dans la détection de la présence du véhicule car il utilise des images de trafic réelles. Il visualise la réalité afin qu'elle fonctionne bien mieux que les systèmes qui dépendent de la détection du contenu métallique des véhicules.*

## II.8. GESTION DU TEMPS DE FEUX VERT [1]

Un cycle se décompose en plusieurs phases, qui ont chacune une durée de feu vert qu'il est primordial de bien régler. En effet, une phase trop courte peut ne pas laisser suffisamment de temps à une file de se décharger. À l'inverse, une phase trop longue augmente le temps d'attente des automobilistes présents sur les autres phases, sans être utile.

Le temps de feu vert  $T_i^g$  pour une phase  $i$  est généralement calculé proportionnellement au temps de cycle, en fonction de différentes contraintes (p. ex., statistiques connues de débit ou d'attente).

Dans certains systèmes, ce temps est calculé dynamiquement et est borné entre un temps minimal  $T_i^{g^{min}}$  et un temps maximal  $T_i^{g^{max}}$  [30]. Une fois le temps de feu vert minimal écoulé, un temps additionnel  $T_i^{g^{add}}$  peut rallonger la phase, jusqu'à éventuellement atteindre  $T_i^{g^{max}}$ . À chaque nouveau véhicule qui arrive assez proche du précédent,  $T_i^g$  est incrémenté de quelques secondes (paramètre généralement proche de  $T^{LT}$  secondes), laissant ainsi le temps au nouveau véhicule de passer. Ceci est illustré au travers de la figure 3.

La somme des durées de chaque phase, en tenant compte des temps intermédiaires, doit être égale au temps de cycle. Ceci est vérifié par la relation suivante :

$$T^c = \sum_i (T_i^{g^{min}} + T_i^{g^{add}} + T^{LT}) \quad (II.2)$$

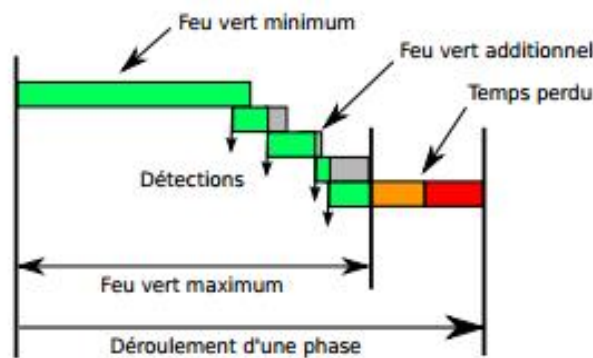


Figure II. 3: Illustration des différents temps de feu vert pour une phase [1].

Notons que le temps de feu vert maximal a fait l'objet de nombreuses études. Les auteurs de [31] observent qu'il doit se situer entre 30 et 60 secondes. Dans [32], il est suggéré que ce temps maximal doit être assez long pour laisser passer 1,3 fois la longueur moyenne de la file concernée. Néanmoins un feu vert maximal élevé n'a que peu d'impact sur un système adaptatif, si le trafic est peu important. Ce qui est tout à fait le cas contraire pour un système à temps fixe (carrefour à temps fixe).

## II.9. CONCLUSION PARTIELLE

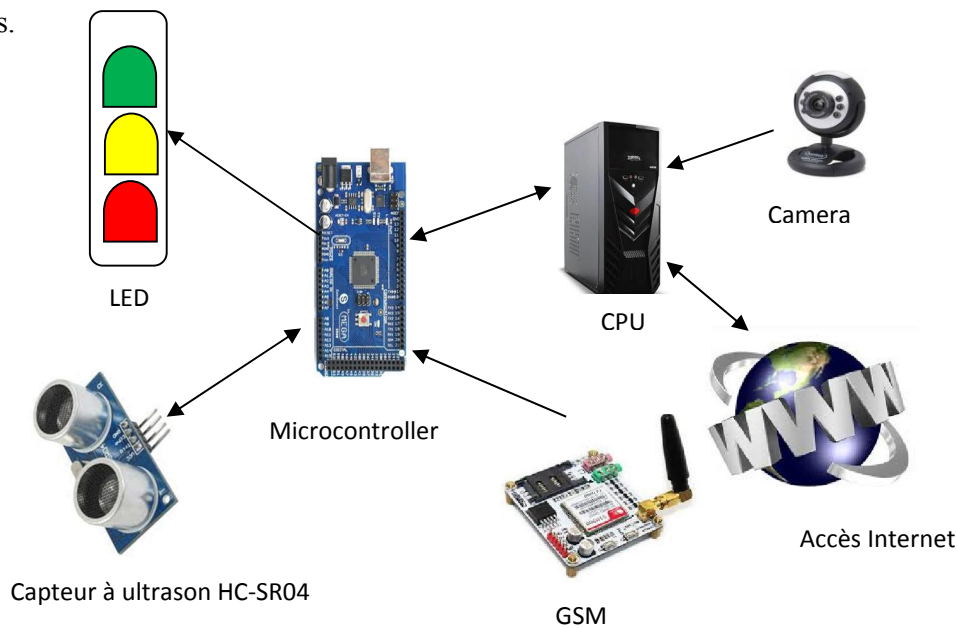
Dans ce chapitre, nous avons présenté les éléments de base composant un trafic routier urbain. Ensuite, nous avons introduit quelques méthodes émergentes et ainsi que quelques notions importantes utilisées pour la régulation du trafic au niveau des carrefours.

# CHAPITRE III. CONCEPTION DU SYSTEME

Dans ce chapitre, nous présentons l'architecture principale de notre système. Ensuite, nous présentons différentes approches et mécanismes que nous avons utilisés pour assurer une bonne gestion de contrôle et de surveillance du trafic routier urbain.

## III.1. ARCHITECTURE DU SYSTEME

Pour parvenir à la réalisation de notre système, nous avons implémenté l'architecture matérielle ci-dessous.



*Figure III.1: Architecture du Système*

Le CPU et le camera constituent l'organe d'acquisition et de traitement d'images. Les caméras sont considérées comme l'œil du système; elles ont comme fonction principale de fournir au microcontrôleur les informations en temps réel sur l'état du trafic.

Le microcontrôleur est le cœur du système, il est celui qui se charge d'ajuster le feu de circulation (LED) en fonction des informations fournies par le CPU.

Le GSM joue le rôle d'un écouteur du microcontrôleur, pour détecter une présence éventuelle d'un véhicule d'urgence, ambulance, VIP, cortège, etc.

Le capteurs à ultrasons HC-SR04 a pour rôle principale de pouvoir détecter la présence d'un véhicule (conducteur) qui viole la loi de circulation, par exemple lorsque qu'un véhicule essaye de passer pendant que le feu de circulation est rouge.

## III.2. REGULATION DES FEUX DE CIRCULATION

La régulation adaptative permet le changement de durée de phase basée sur la demande du trafic. Avec cette méthode, un temps vert minimum et maximum est défini pour chaque voie. Pendant l'écoulement de la circulation, le temps du vert d'une voie peut être prolongé par une unité d'extension si un nouveau véhicule est détecté pour cette phase. Cela donne aux feux de signalisation l'opportunité de s'adapter à la demande de circulation.

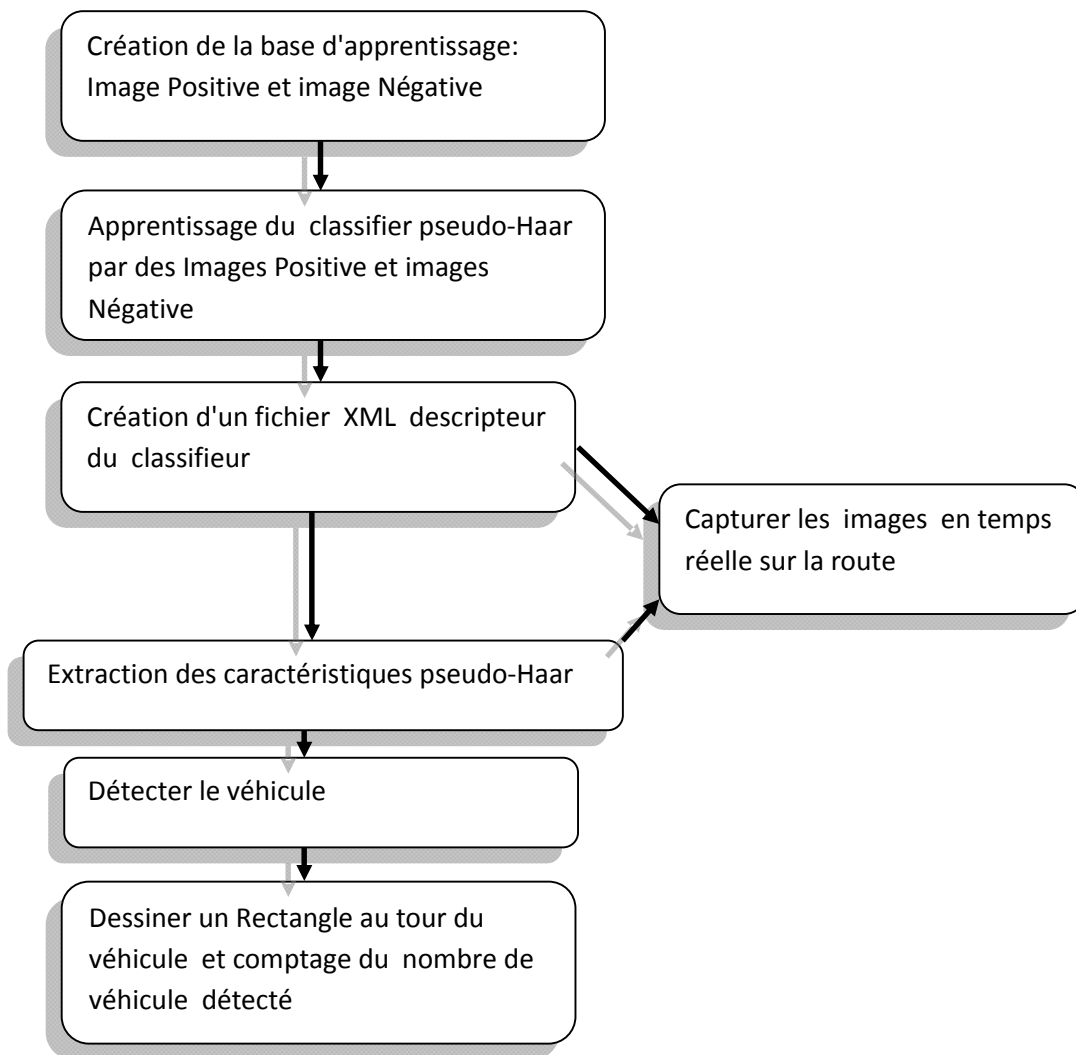
Cependant, l'étape de la détection du véhicule reste la partie la plus primordiale; nous allons essayer d'élucider la manière dont on parvient à détecter le véhicule dans le sous point suivant.

### III.2.1. Détection du véhicule

Dans le cadre de la détection d'objet (véhicule), les approches basées sur l'apprentissage s'appuient sur des images d'apprentissage pour construire un modèle permettant de discriminer des instances de la classe objet par rapport à toutes les instances de la classe non-objet. Les performances de ces méthodes sont entre autres conditionnées par la qualité de la base d'apprentissage qui se doit d'être la plus représentative possible. Autrement dit, cette base doit permettre de capter la variabilité d'apparence présente au sein de la classe objet et non-objet. Dans le cas de cette étude, la classe objet est représentée par des images des véhicules (images positives) alors que la classe non-objet correspond à toutes les images qui ne sont pas des véhicules (image Négative).

La constitution d'une base d'apprentissage est un point important pour la réussite de la réalisation du présent travail. Pour cela, après reconstitution de la base d'apprentissage, nous devons utiliser cette base d'apprentissage pour entraîner un classifieur pseudo-Haar (**Méthode de Viola et Jones**, le principe de fonctionnement de cette méthode sera présentée dans la partie annexe de ce travail). Une fois son apprentissage terminé, nous devons ensuite générer un descripteur pseudo-Haar de ce classifieur permettant de détecter la présence éventuelle de l'objet dans une image en parcourant celle-ci de manière exhaustive. Une fois que l'objet est détecté un contour lui est associé et alors un comptage d'objet détecté est déclenché.

La chaîne de traitement des données est constituée de nombreuses étapes. Nous pouvons synthétiser le processus de détection du véhicule sur la figure suivante:



*Figure III.2: Schéma global du processus de détection du système proposé*

*Les étapes 1 à 3 sont exécutées une seule fois, et les étapes 4 à 7 sont exécutées d'une manière répétitive et simultanée en temps réel.*

### **III.2.2. Processus décisionnel pour le contrôle des feux de signalisation**

Dans cette partie de régulation des feux de circulation, après la phase de détection et de comptage de véhicules par le système, le résultat est analysé puis envoyé au microcontrôleur. Le microcontrôleur prendra la décision en fonction des données qu'il a reçues, puis il enverra une commande aux postes de feux de signalisation et les feux de signalisation seront contrôlés en fonction de cela.

Dans ce système proposé, nous donnons l'illustration d'un processus décisionnel pour le contrôle des feux de signalisation qui est décrit ci-dessous (*Tableau III.1*).

La sortie de la caméra sera balayée deux fois pour vérifier le l'état de la circulation ; ce balayage est décrit dans le *Tableau III.1* comme **cam\_check\_1** et **cam\_check\_2**, où N est le nombre de véhicules détecté par le système.

Tout en nous référant au diagramme fondamental du trafic ainsi qu'à la théorie des trois phases du trafic (*figure II.2*), nous avons considéré trois conditions de circulation différentes, notées comme *situation\_1*, *situation\_2*, et *situation\_3*.

Nous signalons que la valeur seuil du nombre de voitures peut varier en fonction de la taille de la route, ainsi que de la hauteur, de l'angle de positionnement de la caméra, dans notre cas la valeur seuil est estimer à 33 voitures:

**Tableau III.1: Processus décisionnel pour le contrôle le feu de signalisation**

Situation_1	Yellow		Green		Red		Cam_check_1		Cam_check_2	
	On/Off	Time( Sec)	On/Off	Time( Sec)	On/Off	Time( Sec)	N >33	N <33	N >33	N <33
Embouteillage	1	5s	0	0s	0	0s	1	0		
	0	0s	1	60s	0	0s				
	0	0s	1	30s	0	0s			1	0
	0	0s	0	0s	0	0s				
Situation_2	Yellow		Green		Red		Cam_check_1		Cam_check_2	
Intermédiaire	On/Off	Time( Sec)	On/Off	Time( Sec)	On/Off	Time( Sec)	N >33	N <33	N >33	N <33
	1	5s	0	0			1	0		
	0	0s	1	60s						
	0	0s	0	0s	0	0s			0	1
Situation_3	Yellow		Green		Red		Cam_check_1		Cam_check_2	
Fluide	On/Off	Time( Sec)	On/Off	Time( Sec)	On/Off	Time( Sec)	N >33	N <33	N >33	N <33
	1	5s	0	0s			0	1		
	0	0s	1	30s						
	0	0s	0	0s					0	1

Comme nous travaillons avec une intersection à quatre voies (routes), pour l'instant, nous allons nous focaliser (simuler) sur une seule route. Pour chaque route, nous vérifierons deux fois la circulation.

Nous allons démontrer dans la suite que nous pouvons couvrir près de 33 voitures (32,66 voitures exactes) avec notre caméra pour une route spécifique.

Donc, nous allons considérer un seuil de **33 voitures**, cela signifie que si notre caméra vidéo pourrait détecter 33 voitures, alors le CPU enverra un caractère spécifique à la carte Arduino via un port de communication, le microcontrôleur prendra alors une discision en fonction du nombre de véhicules détecté.

Nous avons pour cela 3 types des situations:

- **Situation 1 (Embouteillage Majeur):** où le système détecte un nombre de véhicules supérieur ou égale à  $N$  pour la première fois; le microcontrôleur alloue une période de 60 secondes au feu vert, et si encore pour la deuxième vérification après 56 secondes le système détecte toujours un nombre supérieur ou égale à  $N$  dans ce cas le microcontrôleur ajoutera encore 30 secondes de surplus pour le feu vert. *Dans cette situation, le temps Maximale pour le feu vert est de 90 secondes.*
- **Situation 2 (Intermédiaire):** où le système détecte un nombre de véhicules supérieur ou égale à  $N$  pour la première fois, le microcontrôleur alloue une période de 60 secondes au feu vert et si pour la deuxième vérification après 56 secondes le système détecte un nombre inférieur à  $N$ , dans ce cas le microcontrôleur n'ajoutera pas des secondes supplémentaire au feu vert. *Dans cette situation, le temps Maximale pour le feu vert est de 60 secondes.*
- **Situation 3 (Fluide):** où le système détecte un nombre de véhicules inférieur à  $N$  pour la première fois, le microcontrôleur alloue une période de 30 secondes au feu vert et si pour la deuxième vérification après 26 secondes le système détecte toujours un nombre inférieur à  $N$ , dans ce cas le microcontrôleur n'ajoutera pas de seconde supplémentaire au le feu vert. *Dans cette situation, le temps Maximale pour le feu vert est de 30 secondes.*

### III.3. GESTION DES VEHICULES DE PRIORITE

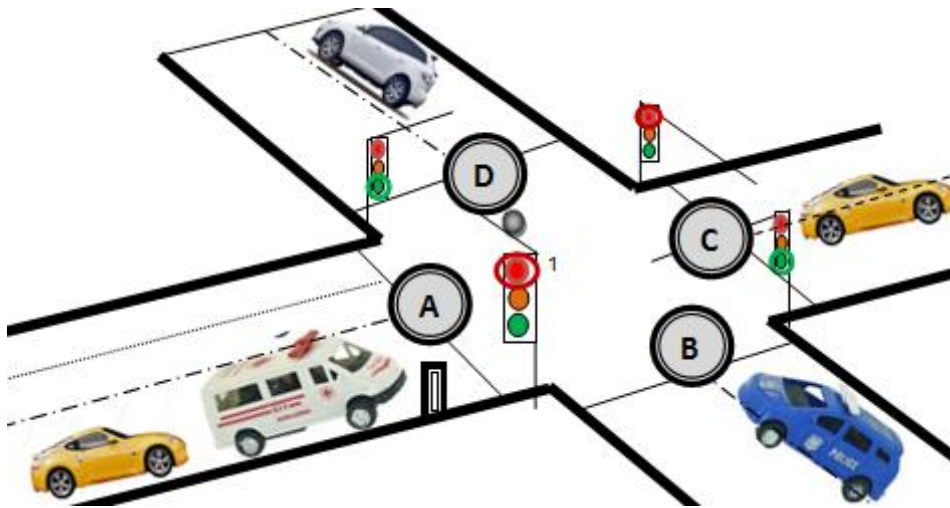
La gestion de véhicule des priorités (ambulance, véhicule VIP,...) est possible grâce à un shield GSM, qui devra recevoir un SMS en provenance d'un véhicule de priorité; le shield GSM envoie à son tour l'information au microcontrôleur, ce dernier analyse l'information reçue, prend une décision et ajuste le feu de circulation en fonction de l'information reçu; face à cette situation exceptionnelle, le microcontrôleur alloue un temps de feu vert de 90 Secondes.

Nous avons choisi d'estimer un temps de feu vert de 90 secondes pour différentes raisons:

- ✓ Nous ne connaissons pas avec exactitude le nombre des véhicules qui sépare le véhicule d'urgence et le point de feu vert,
- ✓ il peut s'agir d'un cortège présidentiel, mariage, enterrement, .... d'où le nombre de véhicule qui forme le cortège nous est inconnu.

Le message qu'un utilisateur peut envoyer au système est de la forme: **source\_destination** ; avec ces deux informations le système est en mesure de pouvoir détecter la destination du véhicule et alors le microcontrôleur ajustera le feu de circulation en fonction de cela.

Par exemple comme nous avons une intersection à 4 entrées (routes), les routes sont numérotées de gauche vers la droite en utilisant les 4 lettres de l'alphabet: A, B, C, D.



*Figure III.3: Gestion de véhicule de Priorité.*

Dans cette illustration, le véhicule d'urgence doit se diriger de la route A vers la route C, alors le conducteur devra envoyer un SMS au système ayant une forme: A\_C, puis le microcontrôleur affichera un feu vert à l'entrée A et C, et un feu rouge à l'entrée B et D.

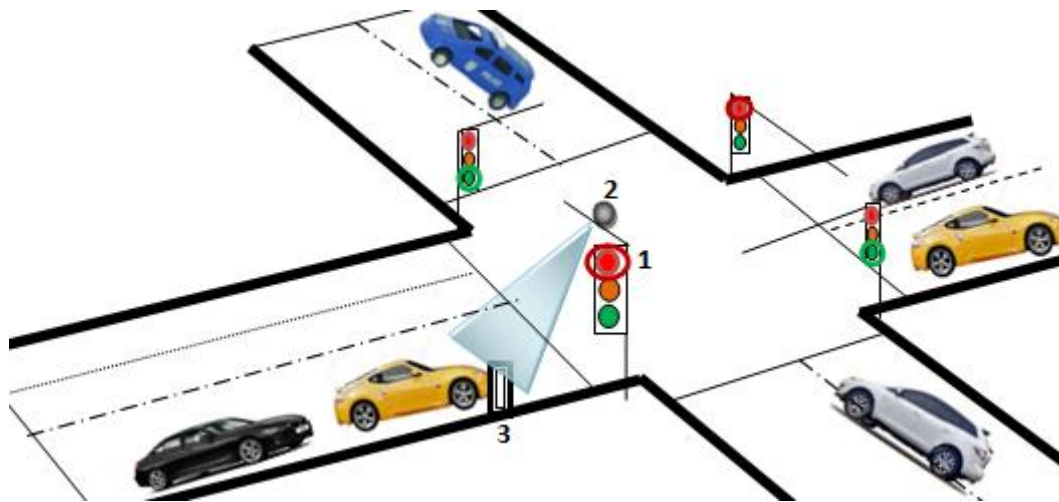
### III.3.1. Mesures de sécurité

Pour des raisons de sécurité, nous avons pris en compte différentes spécifications:

- 1) Tous les véhicules d'urgence doivent être enregistrés avec 2 numéros de téléphone (un numéro principale et un numéro secondaire) aux près de la police de circulation. Seul un numéro enregistré sera en mesure de pouvoir utiliser le module de gestion de priorité de véhicules d'urgence.
  
- 2) En cas de perte ou de vol du téléphone (numéro principale du véhicule d'urgence), le propriétaire du véhicule peut notifier la police de circulation enfin que le module de gestion de priorité soit retiré du numéro de téléphone volé. Cela n'est possible que si le propriétaire utilise le numéro de téléphone secondaire qui avait été enregistré. Il a aussi la possibilité d'envoyer un sms sous cette forme: **LOCK\_XXXXX** (ici XXXXX est le numéro principale du véhicule d'urgence volé).
  
- 3) Un numéro de Téléphone ne peut pas utiliser plus d'une seul fois le module de gestion de priorité dans un intervalle inférieur à 5 minutes.

### III.4. GESTION D'INFRACTIONS AU NIVEAU DES FEUX DE CIRCULATION

Le système est conçu de manière à pouvoir détecter et identifier tout véhicule qui viole la loi de circulation et spécialement au niveau de feux de circulation. Pour mettre en œuvre cette fonctionnalité deux éléments doivent intervenir notamment: une caméra et un capteur à ultrason HC-SR04. Comme nous avons au total une intersection à 4 routes, ces deux éléments sont multipliés par 4 (d'où on aura 4 caméras et 4 capteurs à ultrason HC-SR04). Un exemple du fonctionnement du module est illustré à la figure ci-dessous:



*Figure III.4: Gestion d'Infraction au niveau de feux de Circulation.*

1. le Feux de Circulation est Rouge; 2. une caméra pour capturer l'image du véhicule en infraction, 3. un capteur à ultrason HC-SR04 pour détecter la présence du véhicule.

Une fois que le feu de circulation est rouge, le microcontrôleur active un capteur à ultrason HC-SR04, celui-ci devra détecter tout passage d'un véhicule. Si une fois un véhicule est détecté alors le microcontrôleur recevra le signal et activera à son tour une caméra qui devra capturer l'image du véhicule et l'enregistrer dans un répertoire. Pour éviter de trop surcharger l'unité de contrôle de feux de circulation, toutes les images (véhicules) capturées pendant la journée doivent être envoyées automatiquement vers minuit (par exemple vers 1 heure du matin) à un serveur de données de la police de circulation routière.

Une fois que les données sont arrivées, reçues et archivées au niveau du serveur de la police de circulation, la police de circulation décide et génère une amende d'infraction en fonction de la loi en vigueur du pays.

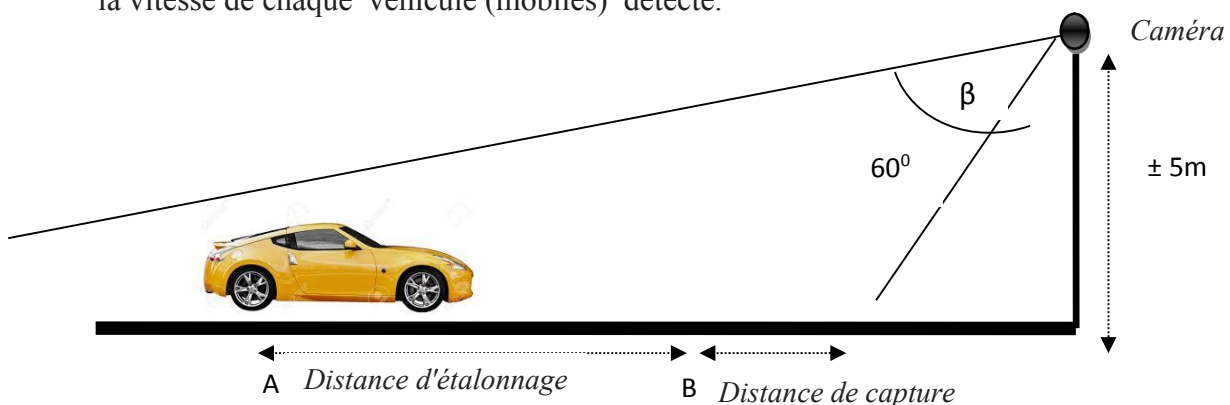
### III.5. GESTION D'INFRACTIONS AU NIVEAU DE LA VOIE DE CIRCULATION

La conception d'un système de détection de la vitesse limite d'un mobile à l'aide de la technologie de vision par ordinateur se fait en deux phases:

- La conception du matériel: Elle est utilisée pour modéliser la capture de données sur le l'état du trafic,
- La conception du programme: Elle comprend la capacité à détecter le véhicule ainsi que de calculer et de mesurer la vitesse des mobiles qui passent.

#### III.5.1. Conception du Matériel

Dans un système de mesure de la vitesse d'un mobile, le système nécessite d'acquérir l'information en temps réelle sur l'état du trafic. La sortie résultante du système est la vitesse de chaque véhicule (mobiles) détecté.



*Figure III.5: Schéma du système d'acquisition de données*

Dans le processus de récupération des données, le camera est placée sur un support métallique ou en béton avec une hauteur de  $\pm 5$  mètres. Cette hauteur est la hauteur standard d'un pont routier qui nous permettra aussi de bien visualiser la plaque d'immatriculation du mobile. La détermination de l'angle  $\beta$  de  $60^\circ$  est basée sur la recherche qui a été menée par Li, et al. [33]. Selon Li, à  $\beta=60^\circ$  d'angle, le résultat de la mesure est plus précis et optimal (où  $\beta$  est l'angle d'inclinaison de la camera face à la voie de circulation).

### **III.5.2. Conception du programme**

La conception du logiciel utilisé dans ce système fait partie du système de traitement d'image. En entrée, le système reçoit une vidéo (état du trafic) prise avec une caméra placée au-dessus de l'autoroute. Dans cette recherche, la conception du logiciel est divisée en deux parties, à savoir:

- ✓ la distance d'étalonnage,
- ✓ la détection véhicule (objet) en mouvement ainsi que la mesure de la vitesse du véhicule.

#### **A. Distance d'étalonnage**

La distance d'étalonnage est une partie importante dans le processus de mesure de la vitesse afin de pouvoir obtenir un meilleur résultat.

Le processus de calcul de la distance d'étalonnage est effectué en conformité avec les conditions réelles sur la voie de circulation, il consiste donc à pouvoir déterminer une distance (exprimée en mètre) entre deux lignes A et B. Cette distance est alors délimitée par le ROI (Région d'Intérêt) en tant que zone régionale de mesure de la vitesse du véhicule (objet).

#### **B. Détection d'un Mobile**

Le système utilise la méthode de détection de blob et la soustraction d'arrière-plan. L'utilisation de ces méthodes est motivée par les résultats d'études de recherche antérieures. En outre, la méthode de détection de blob est utilisée car la méthode facilite la détection d'objets dans une image (séquence vidéo). La méthode de soustraction d'arrière-plan est utilisée parce qu'elle est une méthode qui facilite la séparation entre un objet en mouvement (foreground) et un arrière-plan (background) d'une manière adaptative par rapport à une séquence vidéo.

Ainsi, nous pouvons synthétiser le processus de détection de la vitesse du véhicule sur la figure suivante (figure III.6);

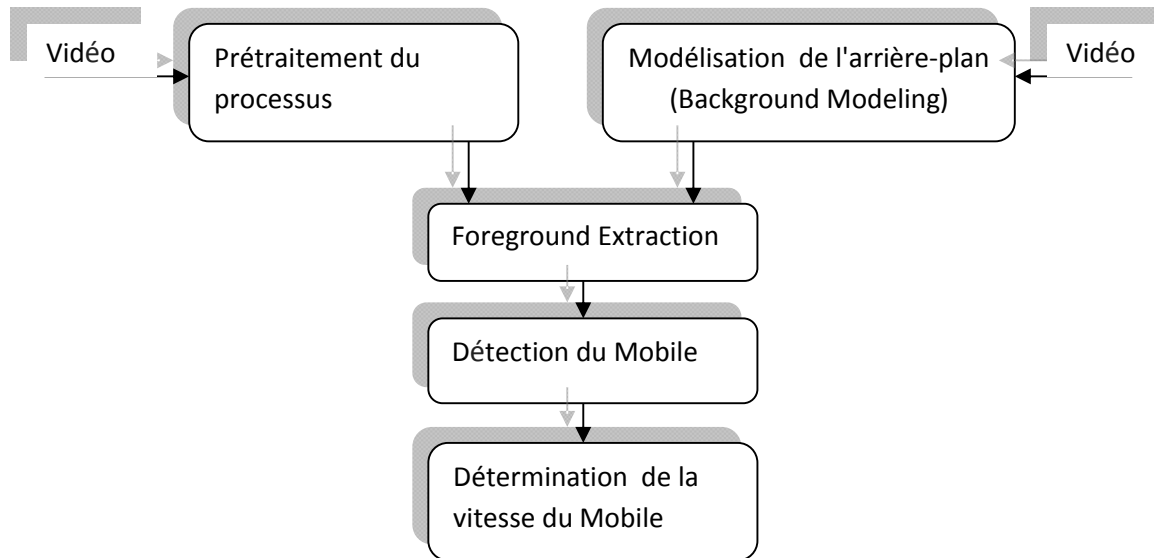


Figure III. 6: Schéma du processus de détection de la vitesse d'un mobile

1. **Prétraitement du processus:** cette phase consiste en la détermination de la distance d'étalonnage (distance se trouvant entre la ligne A et la ligne B):

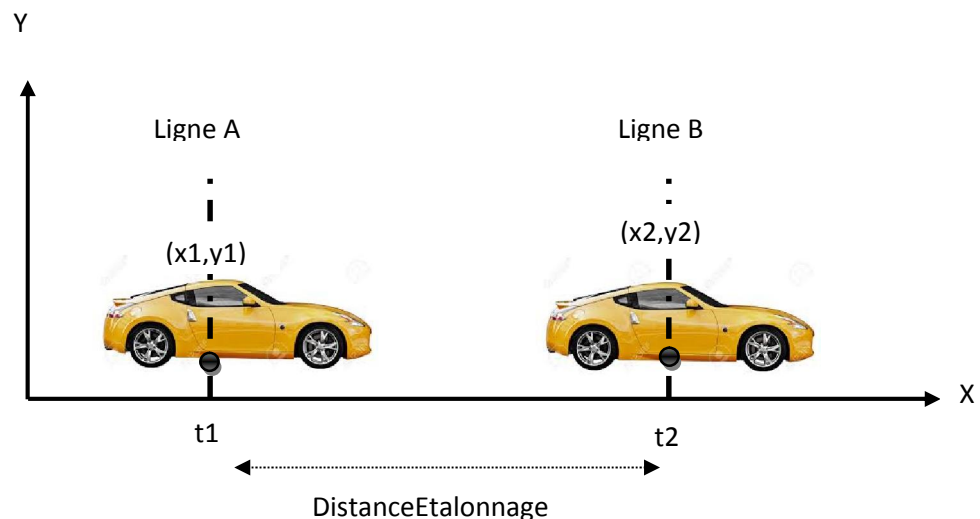


Figure III.7: Schéma de configuration pour la déduction de la vitesse du Mobile

$$\text{distanceEtalonnage} = \sqrt{(x2 - x1)^2 + (y2 - y1)^2} \quad (\text{III.2})$$

2. **Modélisation de l'arrière-plan:** consiste en la définition de l'arrière-plan (tout objet statique ou dynamique non utile se retrouvant dans la scène filmée)
3. **Foreground Extraction:** consiste à pouvoir détecter l'objet recherché (dans notre cas nous recherchons un mobile)
4. **Détection du Mobile:** Elle est le résultat du processus de Forgeround Extraction. Une fois que le mobile est détecté, un processus de calcul du centre de gravité du mobile en mouvement est déclenché.

**5. Détermination de la vitesse du Mobile:** la phase de détection de la vitesse du mobile intervient après celle de détection du mobile, la vitesse du mobile s'évalue comme suit :

$$\text{vitesse} = \frac{\text{distanceEtalonnage}}{t_2 - t_1} \quad (\text{III.3})$$

Où :  $t_1$  est le temps où le mobile (centre de gravité du mobile) franchis la ligne A, et  $t_2$  est le temps où le mobile (centre de gravité du mobile) franchis la ligne B; cette vitesse est alors comparée à la vitesse limite. Si la vitesse calculée est supérieure ou égale à la vitesse limite, un processus de capture et d'archivage de l'image du mobile courant est alors déclenché automatiquement par le système, puis l'image envoyée automatiquement vers le serveur de la police de circulation.

### III.6. DETECTION D'ACCIDENT AU NIVEAU DE LA VOIE DE CIRCULATION

Dans le cadre de la détection des accidents au niveau de la voie de circulation et au niveau de feu de circulation, nous avons utilisé la même approche que celle de la détection du véhicule sauf qu'ici la seule différence se retrouve au niveau du choix de la classe objet (représenter par différents images positive où il y a présence d'un accident) et de la classe non objet (image négative où il n'y a pas d'accidents).

Pour plus de détails référez-vous à la figure III.2 et au point III.2.1 du présent travail.

Les images constituant notre base d'apprentissage ont été extrait de deux vidéos disponibles sur YouTube [34], [35].

### III.8. CONCLUSION PARTIELLE

Dans ce chapitre, nous avons présenté l'architecture principale de notre système. Ensuite, nous avons présenté différentes approches et mécanismes que nous allons utiliser pour assurer une bonne gestion du contrôle et de la surveillance du trafic routier urbain.

# CHAPITRE IV: ANALYSE FONCTIONNELLE ET MODELISATION DU SYSTEME


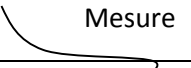
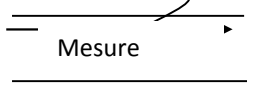
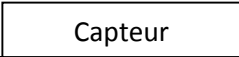
## IV.1. ANALYSE FONCTIONNELLE DU SYSTEME

Dans cette partie, nous allons nous focaliser sur l'analyse fonctionnelle du système. Pour cela nous allons faire intervenir la méthode d'analyse fonctionnelle selon SA-RT (Structured Analysis-Real Time). Pour compléter cette phase d'analyse fonctionnelle nous allons aussi faire recours au diagramme de cas d'utilisation du système.

La méthode SA-RT est une méthode d'analyse fonctionnelle et opérationnelle des applications de contrôle-commande. Cette méthode permet de réaliser une description graphique et textuelle de l'application en termes de besoins, c'est-à-dire de « ce que l'on a à faire » ou le « quoi » (What ?) [36].

### IV.1.1. Les différents éléments graphiques de la méthode SA--RT [36]

**Tableau IV.1: Les différents éléments graphiques de la méthode SA-RT**

Fonction	Signalisation	Représentation graphique	
Traitement ou process	Unité de travail qui Réalise la transformation des données d'entrée en données de sortie	– Cercle ou bulle – Action décrite par : verbe + nom	
Flot de données	Vecteur nommé reliant deux process, sur lequel circule un ensemble de données de même nature	– Flèche en trait plein – Donnée nommée	
Unité de stockage ou réservoir	Entité ou zone de rangement de données	– Deux traits parallèles – Entité nommée	
Entité externe ou terminateur	Provenance, source ou destination des données	– Rectangle – Entité nommée	

### IV.1.2. Diagramme de contexte

Le diagramme de contexte est une première étape extrêmement importante puisqu'elle va définir le contexte et l'environnement extérieur du système piloté.

Nous avons un et un seul processus fonctionnel "SCSTR (*Système de Contrôle et Surveillance du trafic Routier*)", numéroté « 0 », qui traduit l'application à réaliser. Autour de ce processus fonctionnel, nous avons 4 bords de modèles qui fournissent ou consomment les données ou événements de cette application.

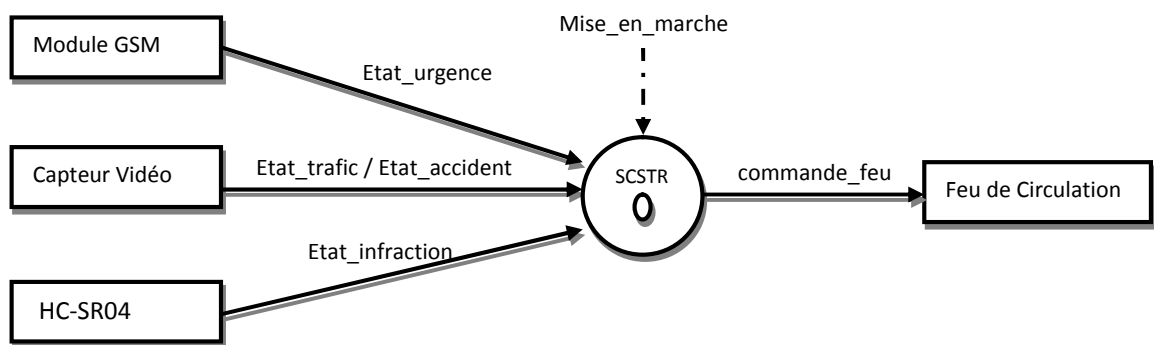
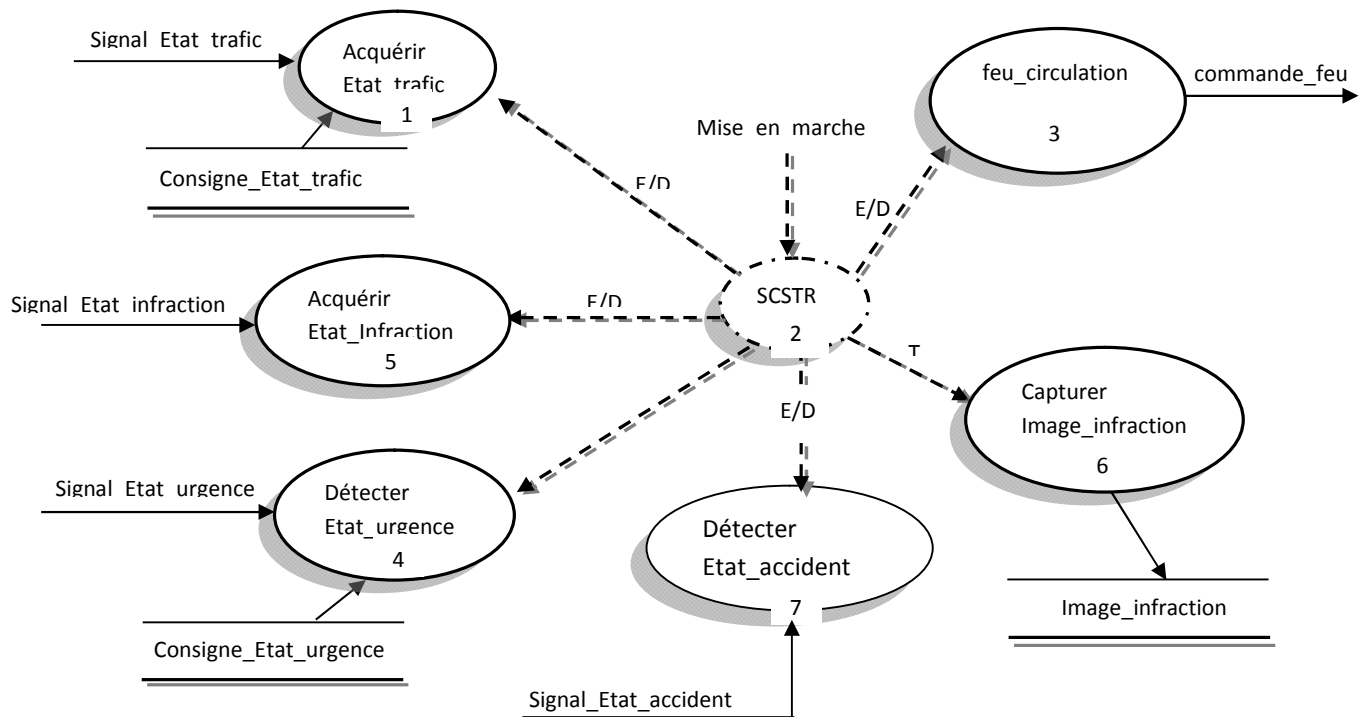


Figure IV.1: Diagramme de contexte générique du système.

### IV.1.3. Diagramme préliminaire

Le premier niveau d'analyse est représenté par le *diagramme préliminaire*. Ce diagramme préliminaire est la première décomposition du processus à réaliser présenté dans le diagramme de contexte. À ce niveau, le diagramme représente la liste « graphique » des processus fonctionnels nécessaires à l'application sans se soucier de l'enchaînement (séquence d'exécution) [36].



*Figure IV. 2: Diagramme préliminaire du système par la méthode SA-RT.*

Rappelons que l'événement couplé «E/D» est utilisé pour piloter un processus fonctionnel de type «boucle sans fin» et l'événement «T» est utilisé pour activer un processus fonctionnel de type «début-fin»

#### IV.1.4. Diagramme de cas d'utilisation

Un diagramme de cas d'utilisation permet de structurer les besoins des utilisateurs et les objectifs correspondant d'un système. Il permet aussi d'identifier les possibilités d'interactions entre le système et les acteurs (intervenants extérieurs au système).

Un **acteur** représente l'abstraction d'un rôle joué par des entités externes (utilisateur, dispositif matériel ou autre système) qui interagissent directement avec le système étudié.

Un **cas d'utilisation** modélise un service rendu par le système et en plus elle représente un ensemble de séquences d'actions qui sont réalisées par le système et qui produisent un résultat observable intéressant pour un acteur particulier.

La mise en marche du système nécessite essentiellement un acteur principale et 4 cas d'utilisation:

- Détecter Infraction,
- Réguler Feux Circulation,
- Détecter véhicule de Priorité,
- Détecter accident,

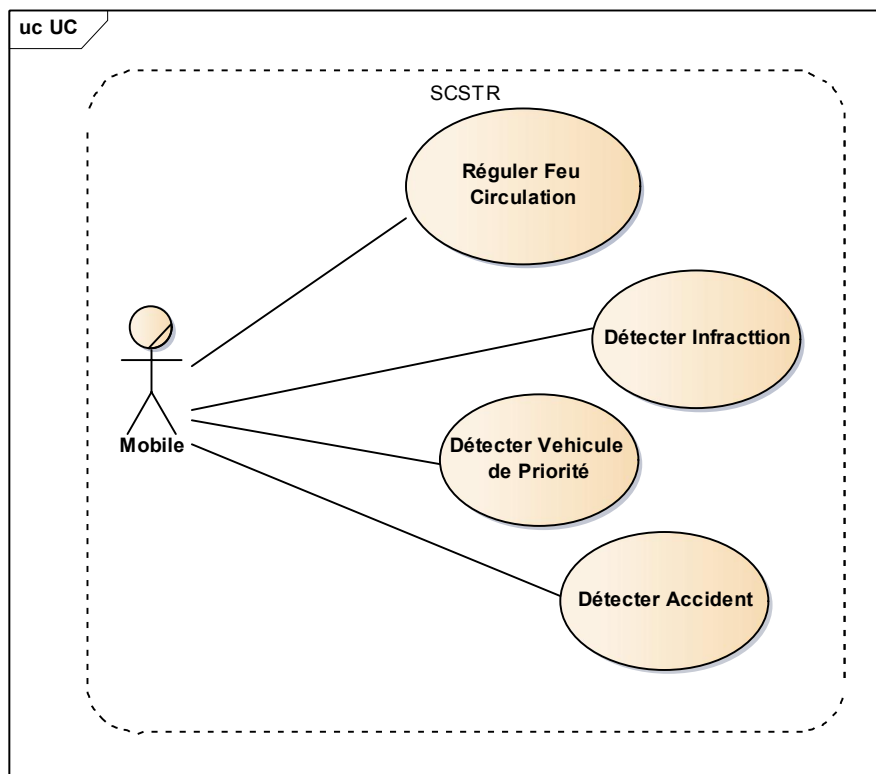


Figure IV. 3: Diagramme de cas d'utilisation du système

#### IV.1.5. Documentation des UCs (User Cases):

Tableau IV.2. : Documentation du diagramme de cas d'utilisation « Réguler Feux Circulation »

UC: Réguler Feux Circulation
ID: 1
Description: Réguler le feu circulation après analyse en temps réelle de l'état du trafic
Acteurs Primaire: Mobile
Acteurs Secondaire: -
Près condition: Aucun
Enchaînement principale: <ul style="list-style-type: none"> <li>1. le système capture et analyse les images du trafic en temps réelle,</li> <li>2. le système détermine le nombre de mobiles détectés,</li> <li>4. Le système ajuste le feu de circulation en fonction du nombre de mobile détectés</li> </ul>
Post Condition: Aucun

**Tableau IV. 3: Documentation du diagramme de cas d'utilisation «Détecter Infraction»**

<b>UC: Détecter Infraction</b>
ID: 2
Description: <i>Détecter une Infraction lorsque le feu de circulation est Rouge</i>
Acteurs Primaire: Mobile
Acteurs Secondaire: -
Près condition: le feu de circulation est rouge ou le module détection vitesse est déclenché
Enchaînement principale: <ul style="list-style-type: none"> <li>1. le UC démarre lorsque le feu de circulation est Rouge ou Lorsqu'un le module détection vitesse est déclenché; <ul style="list-style-type: none"> <li>1.1. Si le feu de circulation est rouge; <ul style="list-style-type: none"> <li>1.1.1. le système active un capteur pour détecter le passage d'un mobile;</li> <li>1.1.2. Si un mobile est détecté; <ul style="list-style-type: none"> <li>1.1.2.1. Le système capture et sauvegarde l'image du mobile;</li> <li>1.1.2.2. Le système sauvegarde l'image capturée dans un répertoire;</li> <li>1.1.2.3. Le système envoie l'image Capturée vers le serveur de la police de circulation;</li> </ul> </li> </ul> </li> <li>1.2. Si le module détection vitesse est déclenché; <ul style="list-style-type: none"> <li>1.2.1. le système déclenche le processus de détection du Mobile;</li> <li>1.2.2. le système calcule la vitesse du mobile;</li> <li>1.2.3. le système compare la vitesse calculée par rapport à la vitesse limite;</li> <li>1.2.4. Si vitesse calculée est supérieure ou égale à vitesse limite ; <ul style="list-style-type: none"> <li>1.2.4.1. Le système capture et sauvegarde les images du mobile;</li> <li>1.2.4.2. Le système envoie l'image capturée vers le serveur de la police de circulation routier;</li> </ul> </li> </ul> </li> </ul> </li> </ul>
Post Condition: Augmentation du nombre de mobile en infraction

**Tableau IV.4: Documentation du diagramme de cas d'utilisation «Détecter Véhicule de Priorité»**

<b>UC: Détecter Véhicule de Priorité</b>
ID: 3
Description: <i>Détecter tout véhicule de priorité, lorsque le feu de circulation est rouge</i>
Acteurs Primaire: Mobile
Acteurs Secondaire: -
Près condition: le feu de circulation est rouge
Enchaînement principale: <ul style="list-style-type: none"> <li>1. le UC démarre lorsque le système reçoit un message d'un mobile;</li> <li>2. le système vérifie l'authenticité du message, <ul style="list-style-type: none"> <li>2.1 Si le message est authentique <ul style="list-style-type: none"> <li>2.1.1 le système change le feu circulation en VERT</li> </ul> </li> </ul> </li> </ul>
Post Condition: Aucun

**Tableau IV.5: Documentation du diagramme de cas d'utilisation «Détecter Accident »**

<b>UC: Détecter Accident</b>
ID: 4
Description: <i>Détecter différentes accidents de circulation observée sur la voie de circulation</i>
Acteurs Primaire: Mobile
Acteurs Secondaire: -
Près condition: -
Enchaînement principale: <ol style="list-style-type: none"> <li>1. le UC démarre lorsque le système détecte un accident sur la voie de circulation;</li> <li>2. le système vérifie pour la deuxième fois le niveau de détection de l'accident,                     <ol style="list-style-type: none"> <li>2.1 Si niveau détection est valide,                             <ol style="list-style-type: none"> <li>2.1.1 le système archive les données en rapport avec l'accident détecté,</li> </ol> </li> <li>2.1. le système envoie un message d'alerte au poste de la police de circulation</li> </ol> </li> </ol>
Post Condition: Augmentation du nombre d'accident

## IV.2. MODELISATION DU SYSTEME

Dans la partie précédente, nous avons présenté les différents diagrammes de cas d'utilisation, le diagramme de contexte, ainsi que le diagramme préliminaire qui nous décrivent un comportement fonctionnel du système. Cependant, dans cette partie du chapitre, nous allons présenter en détails le comportement dynamique du système en faisant appel à la méthode de conception DARTS ainsi qu'au diagramme d'activité d'UML.

Le diagramme d'activité est un diagramme comportemental d'UML, permettant de représenter le déclenchement d'événements en fonction des états du système et de modéliser des comportements pouvant être parallèles. Le diagramme d'activité est également utilisé pour décrire un flux de travail.

La méthode de conception DARTS constitue le lien entre la méthode d'analyse SA-RT et l'implémentation de l'application.

Pour s'harmoniser de façon complète avec la méthode d'analyse SA-RT, la méthode de conception DARTS est de type flots de données. Ainsi, les diagrammes flots de données de la méthode SA-RT (diagrammes préliminaires ou diagrammes de décomposition) sont traduits en diagramme flots de données DARTS représentant l'architecture multitâche de l'application.

## IV.2.1. Les syntaxes graphiques de la méthode DARTS [36]

### A. La modélisation des tâches

En premier lieu, nous trouvons la tâche qui représente l'entité de base de l'architecture multitâche. Nous pouvons avoir un ou plusieurs flots de données en entrées et un ou plusieurs flots de données en sortie. Les tâches sont modélisées par un parallélogramme qui comporte une étiquette ou label explicite comme illustré par la figure IV.4.

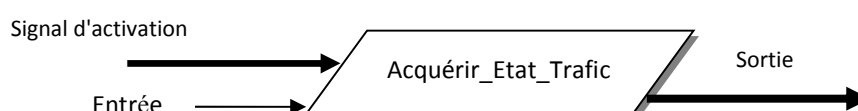


Figure IV.4: Représentation des tâches

### B. La modélisation des synchronisations et communications entre tâches

En ce qui concerne les synchronisations, nous avons deux types de synchronisation; c'est-à-dire la *synchronisation de type asynchrone* et la *synchronisation de type synchrone*. Le modèle «*synchronisation de type asynchrone*» est généralement le plus utilisé car il permet une validation plus aisée de l'application:

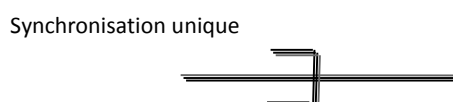


Figure IV.5 : Communication entre tâches

Il existe deux types de tâches :

- *tâches matérielles*
- *tâches logicielles.*

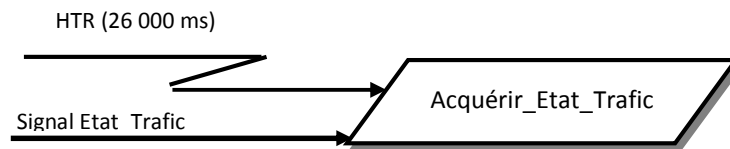
Dans les deux cas, l'activation est très différente.

Ainsi, pour les tâches matérielles qui sont déclenchées par des événements ou signaux externes, nous distinguons trois types de signaux :

- ✓ Signal « **Horloge temps réel – HTR** ». Ce signal, qui provient d'une horloge matérielle interne à l'ordinateur, correspond à un signal rigoureusement périodique.
- ✓ Signal « **Interruption – IT** ». Ce signal qui provient du procédé externe doit toujours être considéré comme aperiodique du fait de l'asynchronisme du monde extérieur par rapport au cadencement de l'ordinateur.

✓ Signal «**Chien de garde – CG** ». Ce signal provient d'une horloge interne utilisée comme un réveil. En termes de signal, il est identique à l'horloge temps réel (signal interne) ; mais il se produit de façon aperiodique.

Les activations sont donc représentées par un symbole orienté (ligne brisée) avec une étiquette ou label explicite comme illustre ci-dessous :

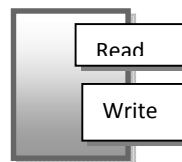


*Figure IV. 6: Activation des tâches.*

### C. La modélisation des stockages de données

Le dernier élément à modéliser est le **module de données** qui permet une protection des accès à une unité de gestion de données en **exclusion mutuelle** par deux ou plusieurs tâches.

Les modules de données sont représentés par un rectangle associé à des entrées permettant de réaliser une action sur les données : READ, WRITE, etc. Ce symbole du module de données est représenté avec une étiquette ou label explicite formé de :



**Figure IV. 7: Module de donnée.**

## IV.2.2. Description du diagramme d'activité du cas d'utilisation « Réguler Feu Circulation »

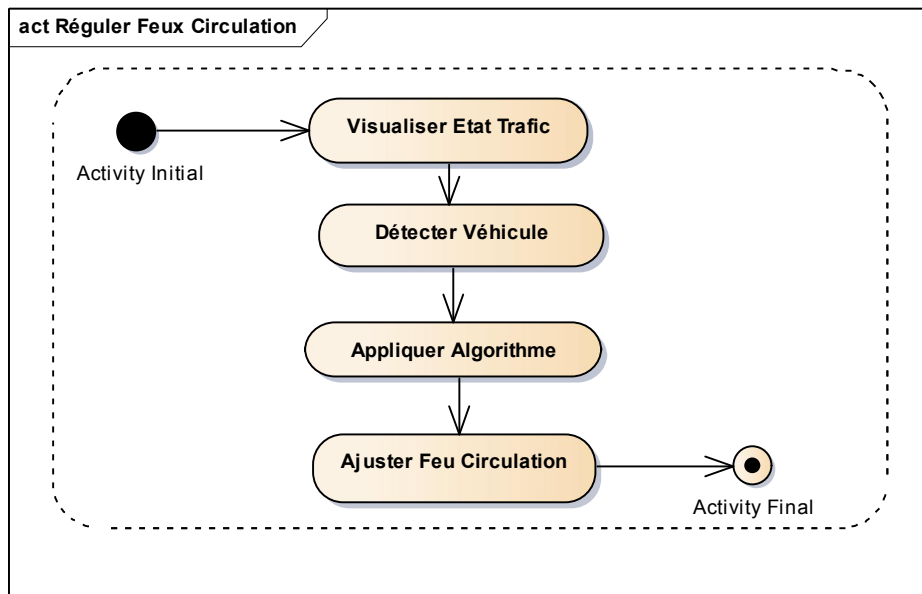


Figure IV. 8: Description du diagramme d'activité du cas d'utilisation « Réguler Feu Circulation »

Sur la figure ci-dessus, nous avons fait une représentation graphique du diagramme d'activité permettant de représenter le déclenchement d'événements en fonction de l'état du système pour le cas du diagramme de cas d'utilisation « Réguler Feu Circulation ».

### IV.2.3. Description du diagramme d'activité du cas d'utilisation « Détecter Véhicule de Priorité »

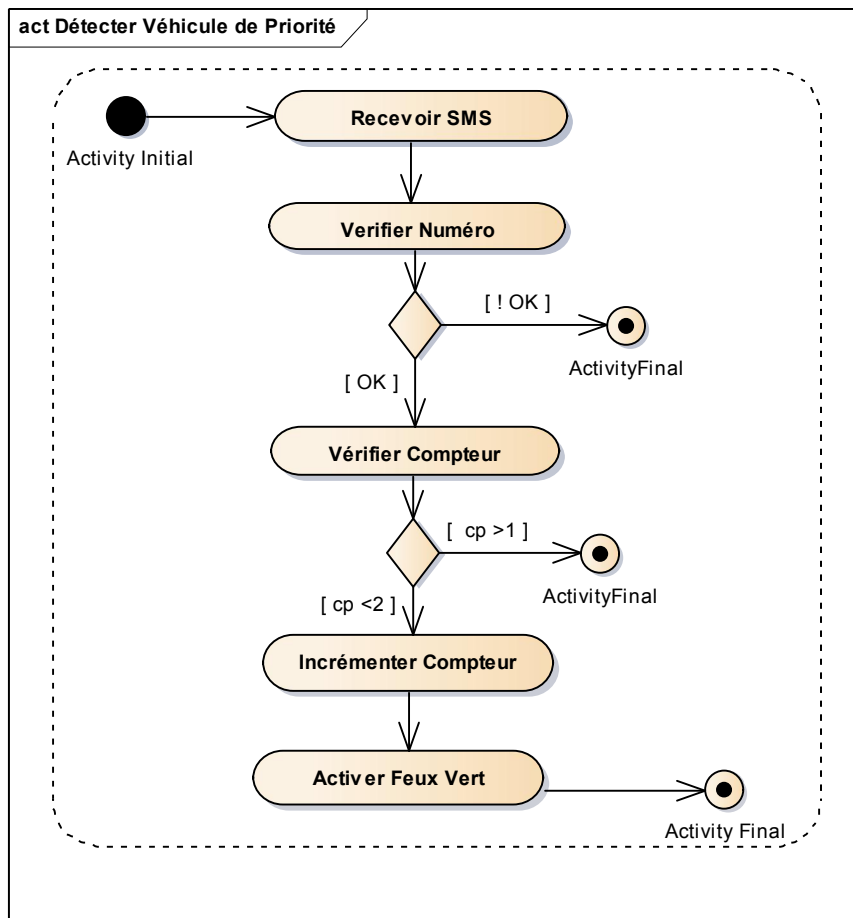


Figure IV. 9: Description du diagramme d'activité du cas d'utilisation « Détecter Véhicule de Priorité »

Sur la figure ci-dessus, nous avons fait une représentation graphique du diagramme d'activité permettant de représenter le déclenchement d'événements en fonction de l'état du système pour le cas d'utilisation « Détecter Véhicule de Priorité ».

#### IV.2.4. Description du diagramme d'activité du cas d'utilisation « Détecter Infraction »

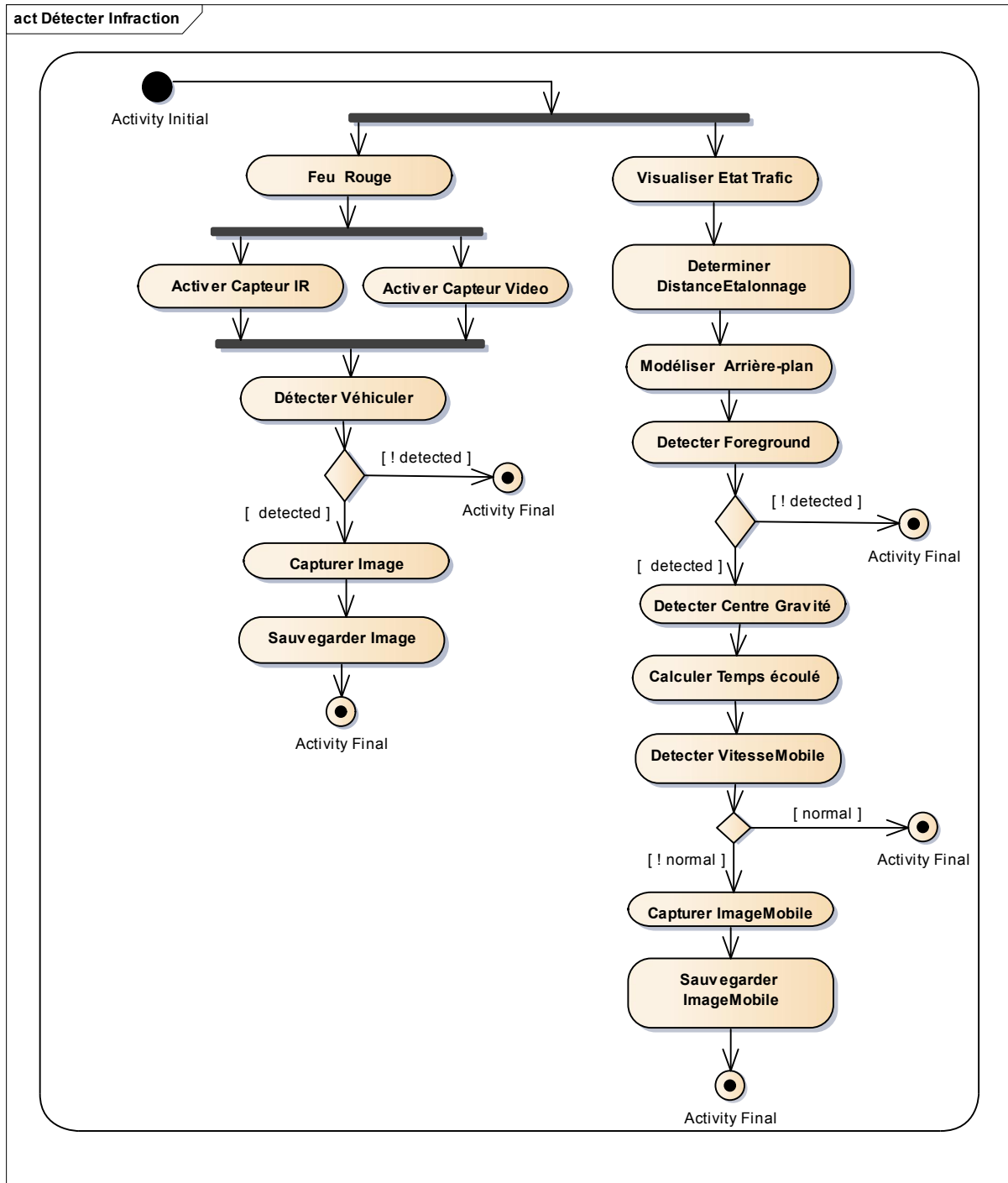


Figure IV. 10: Description du diagramme d'activité du cas d'utilisation « Détecter Infraction »

Sur la figure ci-dessus, nous avons fait une représentation graphique du diagramme d'activité et une modélisation des comportements pouvant être en parallèles permettant de représenter le déclenchement d'événements en fonction de l'état du système pour le cas d'utilisation « Détecter Infraction ».

#### IV.2.5. Description du diagramme d'activité du cas d'utilisation « Détecter Accident»

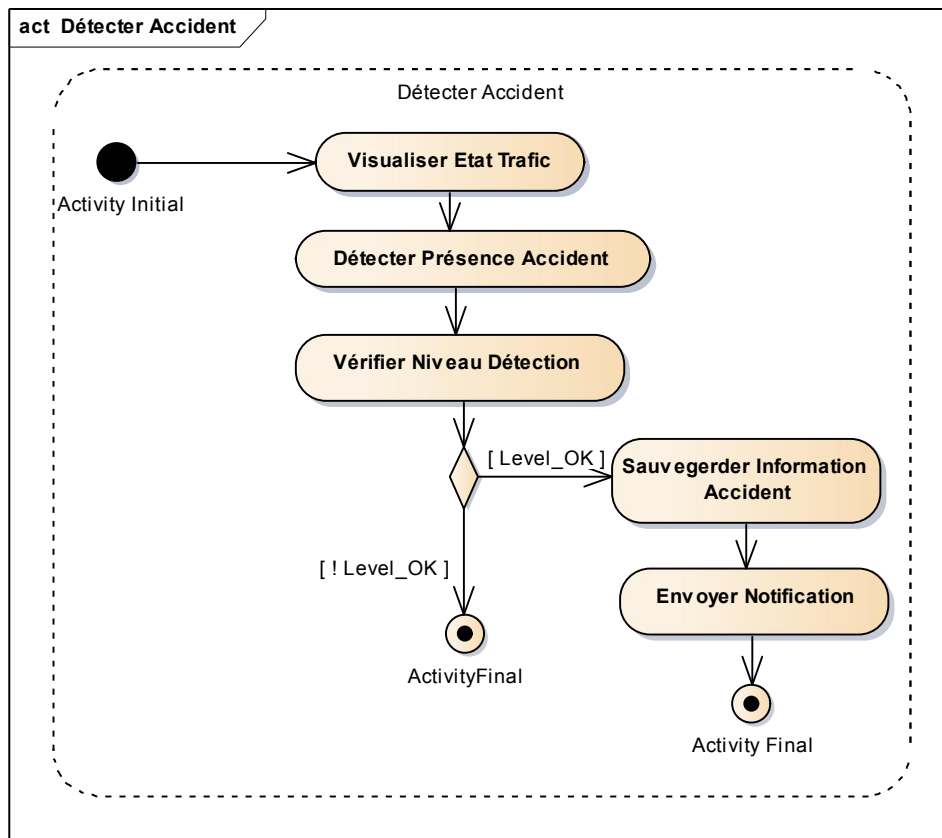
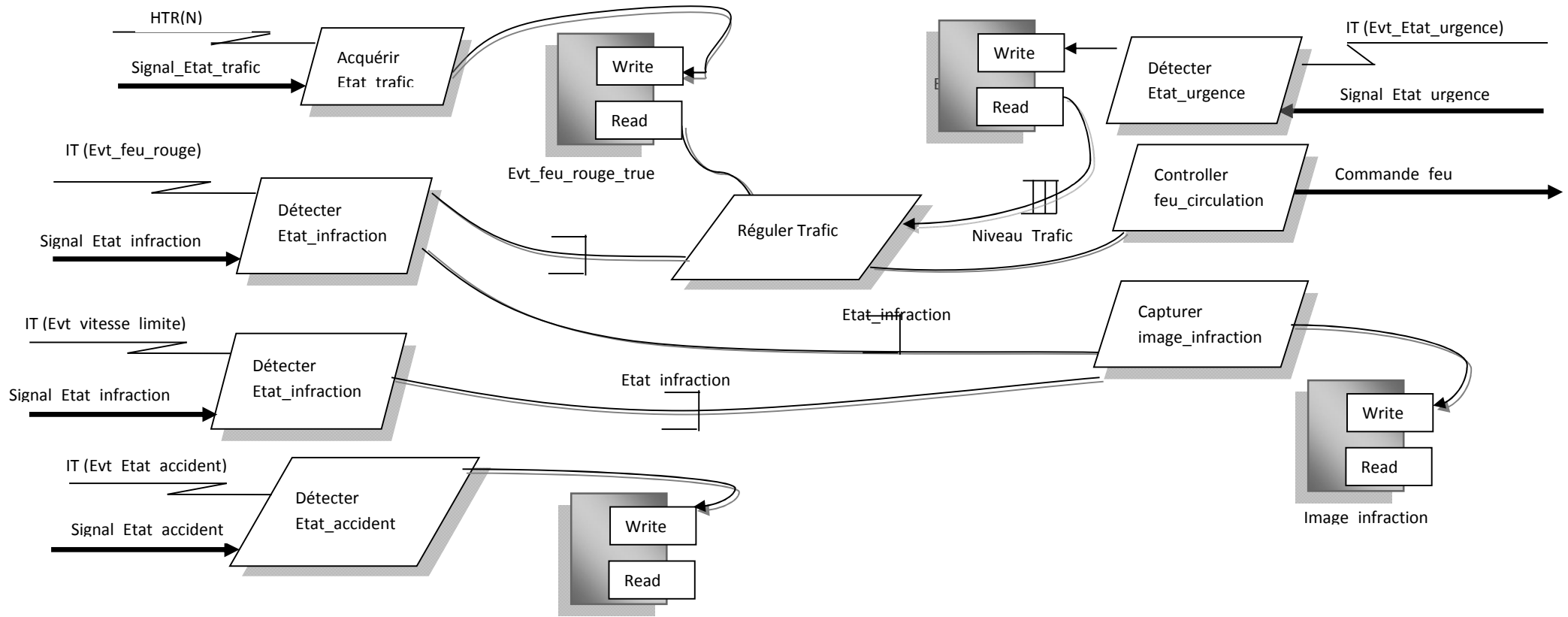


Figure IV.11: Description du diagramme d'activité du cas d'utilisation « Détecter Accident»

Sur la figure ci-dessus, nous avons fait une représentation graphique du diagramme d'activité permettant de représenter le déclenchement d'événements en fonction de l'état du système pour le cas d'utilisation « Détecter Véhicule de Priorité».

### IV.2.6. Diagramme Multitâche DART du système



Donnée accident **Diagramme multitâche DARTS du système.**

où N= 26 000 ms, 56 000 ms

Sur la figure ci-dessus, nous avons fait une représentation graphique du comportement dynamique multitâche du système en faisant appel à la méthode de conception DARTS.

### IV.3. CONCLUSION PARTIELLE

Dans ce chapitre, nous nous sommes focalisé sur l'analyse fonctionnelle du système. Pour cela nous avons fait intervenir la méthode d'analyse fonctionnelle selon SA-RT (Structured Analysis-Real Time) et au diagramme de cas d'utilisation du système. Ensuite nous avons présenté la modélisation de la solution proposée en faisant appel à la méthode de conception DARTS ainsi qu'au diagramme d'activité d'UML en vue de bien capter les détails nécessaires pour la phase d'expérimentation de notre système.

# CHAPITRE V. EXPERIMENTATION ET ESTIMATION DU COUT DU PROJET

Ce dernier chapitre est consacré à l'expérimentation et l'estimation du coût du système qui permettra d'assurer le contrôle et la surveillance du trafic routier dans un milieu urbain. Il est question ici de présenter différents tests effectués sur des séquences de mise en situation ainsi que d'estimer le coût du projet par utilisation de la méthode de COCOMO.

## V.1. EXPERIMENTATION

Le système est constitué de deux parties principales; chacune de ces parties sera codée dans un langage de programmation spécifique:

- le CPU sera codé en C++ et utilisera les bibliothèques OpenCV pour pouvoir prendre des photos (ou image vidéo en temps réel) à l'aide d'une caméra. Cette dernière est considérée comme l'œil du système et a comme fonction principale de fournir au microcontrôleur les informations sur l'état du trafic.
- le microcontrôleur sera codé en langage Arduino, son rôle sera d'ajuster le feu de circulation en fonction de l'état du trafic tout en implémentant la logique que nous avons élaborée (*Tableau III.1*).

### V.1.1. Préparation du CPU:

#### A. Environnement du travail:

Afin de mener à bien ce projet, nous avons utilisé :

- ❖ un ordinateur dont les principales caractéristiques sont les suivantes :
  - Processeur : Intel(R) Core(TM)2 Duo CPU P9400 @ 2.1GHz (2 CPUs), ~2.1GHz
  - RAM : 3.00 GO
  - OS : Windows 7 Édition Intégrale 64-bit
- ❖ une webcam dont les principales caractéristiques sont les suivantes :
  - Type : USB Webcam
  - Model : Hp,
  - Résolution Vidéo : 50 MP, 1024 x 768
  - Fréquence d'image : 30 trames / s

## B. Environnement Logiciel:

Lors du développement de notre projet, nous avons utilisé les outils suivants :

### 1. Qt Creator (Community, Version 3.3.1, Basé sur Qt 5.4.1 (MSVC 2010,32 bit)):

Cette IDE nous a servi comme éditeur du code en langage C++, sous le compilateur CGUI

2. l'IDE Arduino (Version 1.6.2): Vu que notre microcontrôleur (ATMega2560) sera code en langage arduino et que ce dernier repose sur le matériel et logiciel. Cet IDE nous a servi comme éditeur du code en langage Arduino.

3. Proteus ISIS Professional (version 7.10): Vu que notre système est basé sur quelque capteurs et composants électroniques, le logiciel Proteus ISIS nous a bien servi pour la modélisation et la simulation de schéma électroniques.

## V.1.2. DataSet (Base d'apprentissage)

Pour parvenir à mettre en marche notre système, nous avons constitué une base d'apprentissage (un data set) contenant 1000 images; ces images nous ont servi dans la phase d'apprentissage de notre classifieur. Les images ont été extraites d'une vidéo enregistrée localement à partir d'une hauteur d'environ 7 m avec une caméra de 50MegaPixel. Les 1000 images sont subdivisées en deux catégories notamment 594 images positives et de 406 Images négatives. Exemple d'illustration de deux catégories d'images:

### Images Positives:



*Figure V.1: quelques images positives*

### Images négatives:



*Figure V.2: quelques images négatives*

### V.1.3. Fixation de la caméra (positionnement de la caméra)

Le fonctionnement du système repose sur la détection des véhicules. La qualité des images vidéo est un facteur important pour la robustesse et la précision du système. Pour détecter une voiture ou tout autre véhicule, le positionnement de la caméra est une phase très importante pour une détection précise de véhicule.

La hauteur et l'angle parfaits pour la caméra permettent une grande précision pour la détection de voitures, en assurant la plus grande couverture de la région. Plus la couverture de la caméra est élevée, meilleure est la décision que prendra le système.

Dans le cadre du présent travail, en fonction des données enregistrées, nous estimons que la caméra doit être positionnée dans une certaine gamme de hauteur afin que le système puisse détecter autant de voitures que possible sur une route.

Après avoir pris des échantillons à différentes hauteurs, nous avons constaté que lorsque la hauteur de la caméra est au-delà de 7,62 m elle nous donne le meilleur résultat.

Sur la figure V.3, le positionnement de la caméra est illustré. Ainsi, la hauteur de la caméra ne peut pas être inférieure à 7,62 m et en plus la caméra doit être placée perpendiculairement face à la route sinon le système aura des difficultés à pouvoir détecter les véhicules.

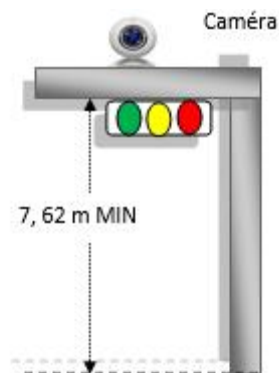
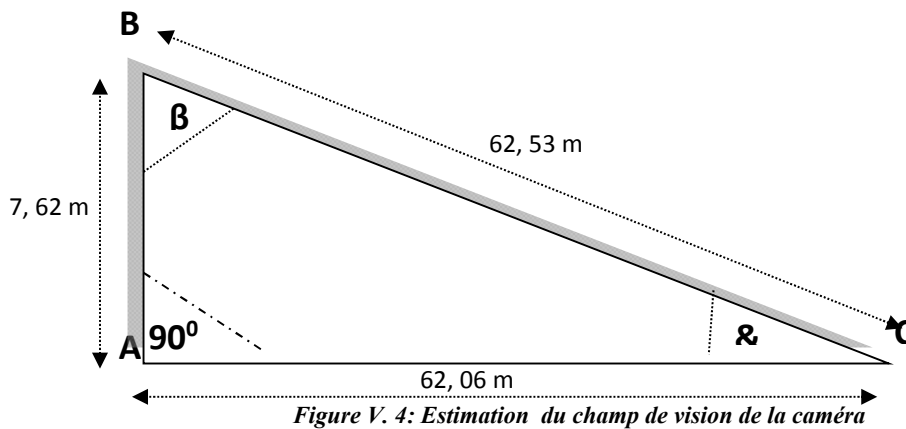


Figure V. 3: Fixation de la caméra

L'estimation du champ de vision de la caméra est illustrée par la figure ci-dessous :



Dans cette expérimentation, nous utilisons:

Une caméra placée à une hauteur de 7,62 mètres.

$\alpha = 7^\circ$  et  $AB = 7,62$  m

$\beta = (90^\circ - 7^\circ) = 83^\circ$

Notre objectif est de pouvoir obtenir (détecter) plus de 30 voitures dans une voie. Par cet arrangement de la figure V.4, nous pouvons obtenir une couverture de 62,06 mètres de longueur.

Maintenant, nous devons calculer combien de voitures possibles pourraient être détecté dans cette voie.

La longueur moyenne d'une voiture RAV4 est d'environ 4,6 mètres; considérant une distance de sécurité (espacement entre véhicule) de 3 m ; Alors:  $\frac{62,06}{4,6+3} = 8,1658 \cong 8$  Vehicules.

Supposons que nous avons une route à 4 colonnes des voitures, ainsi nous nous retrouverons avec  $8,1658 \times 4 = 32,66$  voitures (33 voitures approximativement).

Nous signalons que l'objectif principale est de pouvoir couvrir le plus d'espace que possible (accroître les distances AC et AB) en vue de pouvoir détecter plus des véhicules que possible.

## V.2. SCHEMA ELECTRONIQUE DU PROJET

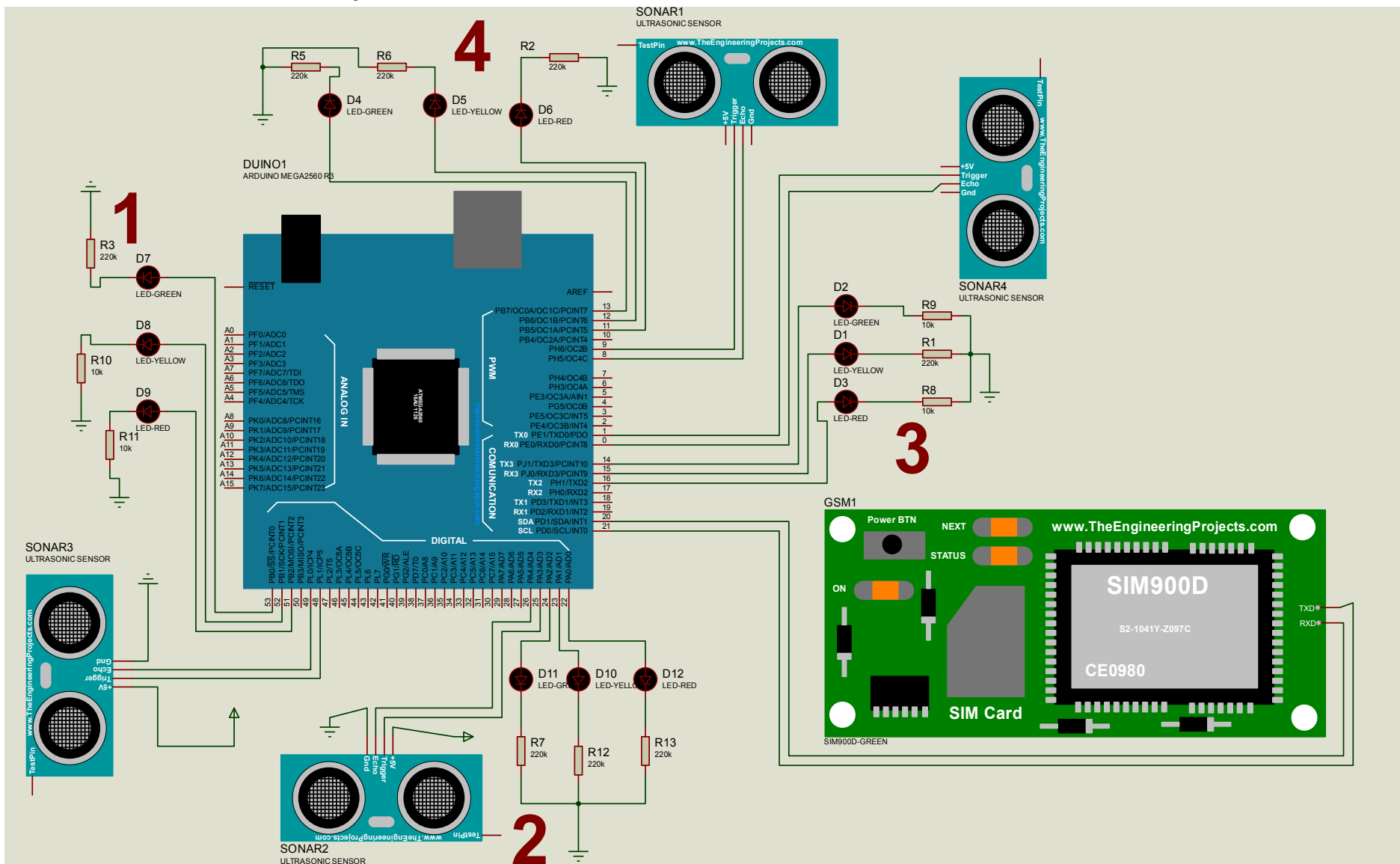


Figure V. 5: schéma électronique du projet

Sur la figure V.5, est illustré le schéma électronique du projet commandé par un microcontrôleur ATmega2560, hormis la partie CPU qui est liée directement aux différentes caméras. Nous avons utilisé la carte arduino méga, vu que cette dernière a plusieurs broches (53 digitales et 16 analogiques) et que nous avons plusieurs composants à y brancher. Elle est alimentée avec source de tension de 9 Volts via le port Universal Serial Bus (USB) de l'ordinateur.

### V.3. QUELQUES RESULTATS:

#### A. MODULE DE COMPTAGE DE VEHICULE

##### A.1. Image temps réel du trafic avant détection:



Figure V.6: Image temps réel du trafic avant détection

Sur la figure ci-dessus (figure V.6) est illustré l'état du trafic routier avant la détection (comptage du nombre des véhicules) par le système.

##### A.2. Image temps réel du trafic avant après détection:

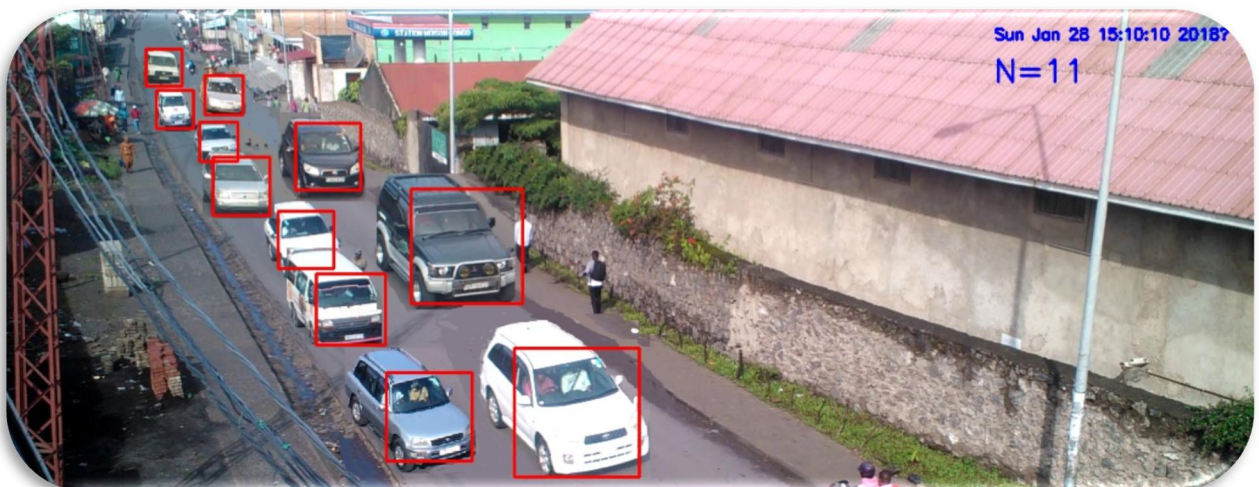
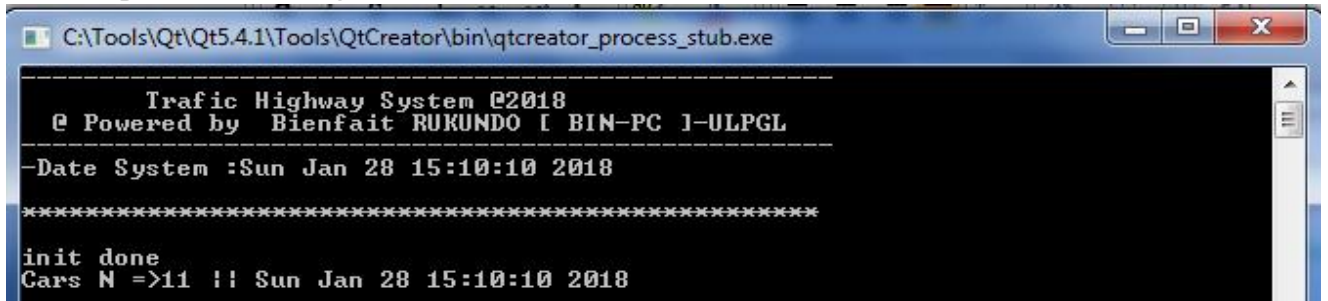


Figure V.7: Image temps réel du trafic avant après détection

Sur la figure ci-dessus (figure V.7) est illustré l'état du trafic routier après détection (comptage du nombre des véhicules) par le système. Cette détection indique qu'il y a 11 véhicules comptés sur la route.

### A.3. Output console du système :



```
C:\Tools\Qt\Qt5.4.1\Tools\QtCreator\bin\qtcreator_process_stub.exe
-----
Traffic Highway System @2018
@ Powered by Bienfait RUKUNDO [ BIN-PC 1-ULPGL
-----
-Date System :Sun Jan 28 15:10:10 2018
*****
init done
Cars N =>11 !! Sun Jan 28 15:10:10 2018
```

Figure V. 8: Output console du système de détection de véhicule

Sur la figure ci-dessus (figure V.8) est illustré l'output console du système indiquant l'heure, la date et le nombre des véhicules détectés (11 véhicules détectés) par le système pour le cas du module de comptage de véhicule.

## B. MODULE DE DETECTION D'ACCIDENT

### B.1. Image du trafic après détection

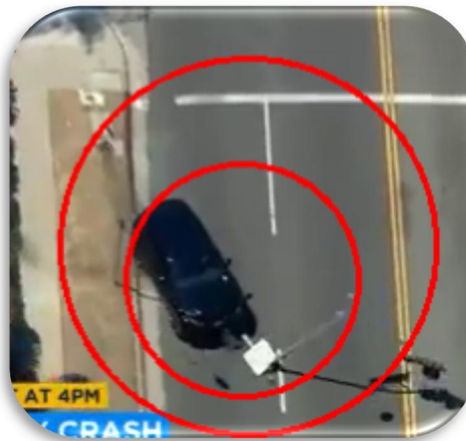
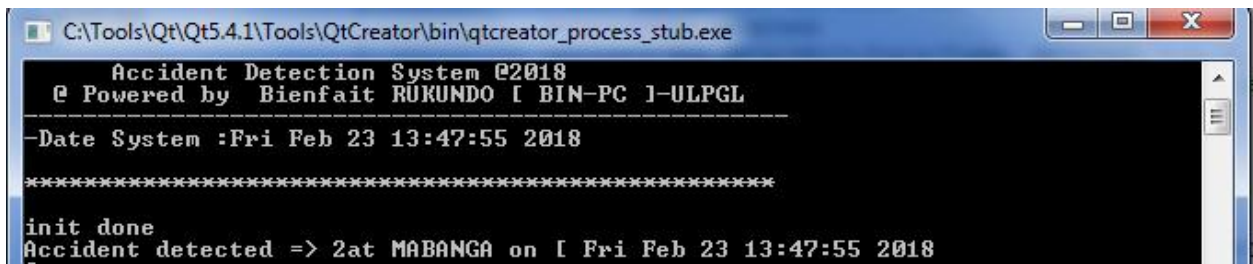


Figure V.9: Image du trafic après détection d'un accident

Sur la figure ci-dessus (figure V.9) est illustré l'état du trafic routier après détection d'accident par le système.

## B.2. Output console du système :



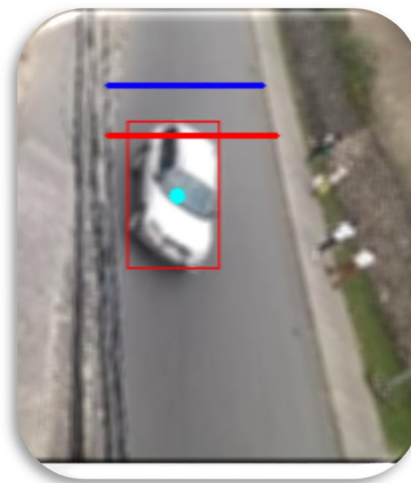
```
C:\Tools\Qt\Qt5.4.1\Tools\QtCreator\bin\qtcreator_process_stub.exe
Accident Detection System ©2018
© Powered by Bienfait RUKUNDO I BIN-PC I-ULPGL
-----
-Date System :Fri Feb 23 13:47:55 2018
*****
init done
Accident detected => 2at MABANGA on [ Fri Feb 23 13:47:55 2018
```

*Figure V. 10: Output console du système après détection d'un accident*

Sur la figure ci-dessus (figure V.10) est illustré l'output console du système du module de détection d'accident indiquant l'heure, la date, le lieu et l'accident détecté par le système.

## C. MODULE DE DETECTION DU DEPASSEMENT DE LA VITESSE

### C.1. Image du trafic pendant la détection de la vitesse



*Figure V. 11: image du trafic pendant la détection de la vitesse*

Sur la figure ci-dessus (figure V.11) est illustré un processus de détection de la vitesse du véhicule par le système pour le cas du module de détection du dépassement de la vitesse limite.

### C.1. Output console du système :

```

C:\Tools\Qt\Qt5.4.1\Tools\QtCreator\bin\qtcreator_process_stub.exe
-----
Trafic Highway System ©2018
© Powered by Bienfait RUKUNDO [ BIN-PC ]-ULPGL
-----
-SPEED CHECK Module:
-Date System :Fri Feb 23 13:54:10 2018
-----
*****
init done
-----
W->[11]-> 10.9581 KM/H on Fri Feb 23 13:54:14 2018
-----
*****

```

*Figure V. 12: Output console du système pendant la détection de la vitesse*

Sur la figure ci-dessus (figure V.12) est illustré l’output console du système pour le cas du module de détection du dépassement de la vitesse limite, indiquant l’heure, la date, et la vitesse du mobile détecté par le système.

## V.4. ESTIMATION DU COUT DU PROJET

### A. COUT MATERIEL:

**Tableau V.1: Liste des matériels**

N°		Prix Unitaire [USD]	Quantité	Coût Total [USD]
1	Ordinateur (CPU)	500	1	400
2	Capteur Vidéo	20	6	120
3	Arduino MEGA	15	1	15
4	capteurs à ultrasons HC-SR04	2	4	8
5	LED	1	22	22
6	GSM Shield	30	1	30
7	Pilonne	50	4	200
	<b>Coût matériel en USD</b>			<b>795</b>
	<b>Main d’œuvre (30% du prix matériel)</b>			<b>238</b>
	<b>TOTAL GENERAL</b>			<b>1 033 USD</b>

Le tableau ci-dessus (Tableau V.1), nous donne une illustration du coût et des différents matériels nécessaire pour la mise en place de notre système. Pour cela le coût matériel est estimé à **1 033 USD**.

## B. Cout Logiciel avec COCOMO:

Constructive Cost Model« COCOMO » c'est un modèle qui permet d'estimer le coût, l'effort et le temps nécessaire au développement d'un logiciel. La méthode COCOMO se base sur une approche algorithmique pour déterminer « l'effort » et le « temps de développement » d'une application. Son principe est basé sur le nombre de lignes de code en Kilo (KLOC) [37].

### B.1. Module du programme de control et surveillance du trafic routier urbain

Tableau V.2: Découpage en Module

Module	Fonctionnalité
Gestion Feu_circulation	<ul style="list-style-type: none"><li>- Détecter véhicule,</li><li>- Regulerfeu_circulation,</li><li>- Détecter véhicule_prioritaire.</li></ul>
Détection infractions	<ul style="list-style-type: none"><li>- Détecter vitesse limite</li><li>- Détecter passage interdit</li></ul>
Détection accident	<ul style="list-style-type: none"><li>- Détecter accident</li></ul>

Le tableau ci-dessous (Tableau V.2) donne les différents modules principaux de notre système.

### B.2. Comptage des points de fonctionnement non ajusté :

Les points de fonction non ajustés sont :

#### 1) Dépôts internes aussi appelés fichiers logiques internes

Tableau V.3: Dépôts internes

Fonctionnalité	Complexité	Valeur
Détecter véhicule	Élevée	10
Regulerfeu_circulation	Élevée	7
Détecter véhicule_prioritaire	Élevée	8
Détecter vitesse limite	Élevée	11
Détecter passage interdit	Élevée	10
Détecter accident	Élevée	10
<b>TOTAL</b>		<b>56</b>

Le tableau ci-dessus (Tableau V.3) représente différentes fonctionnalités du système en fonction de leur valeur de complexité.

## 2) Intrants aussi appelés entrés externes

*Tableau V.4: entrés externes*

Attribut	Donnes élémentaire	Complexité	Valeur
Sauvegarder	typeEtatTrafic, ImageConcerné, date, localisation	Moyenne	3
<b>TOTAL</b>			<b>3</b>

Le tableau ci-dessus (Tableau V.4) représente la somme de tous les intrants du système estimé à 5 intrants.

Ainsi, les points de fonction non ajustés sont calculés avec la grille suivante :

*Tableau V.5: Comptage des points de fonction non ajusté*

Composantes				TOTAL
	Simple	Moyenne	Complexe	
Dépôt internes	0	0	6	56
Intrants	0	1	0	3
<b>Nombre de point de fonction non ajustée</b>				<b>59</b>

Le tableau ci-dessus (Tableau V.5) représente la somme de tous les points de fonction non ajustée qui est estimée à 83 points de fonction non ajustée.

## 3) Evaluation des fonctions d'ajustement

Les facteurs d'ajustement à prendre en compte pour calculer les points de fonctions sont :

*Tableau V.6: fonctions d'ajustement*

Facteurs	Degrés d'influence
Traitement distribué	5
Performance	6
Charge de l'équipement	4
Saisie des données en direct	1
Mise à jour en direct	2
Complexité de traitement	6
Réutilisabilité	4
Sécurité	5
Degré Total d'influence (DTI)	33

Le tableau ci-dessus (Tableau V.6) illustre la répartition du degré d'influence en fonction des différents facteurs qui est estimée à 33 degrés d'influence.

#### 4) Calcul du nombre de point de fonctions

Finalement le nombre de point de fonction est :

$VAF = DTI * 0.01 + 0.65$  où VAF est le facteur d'ajustement

$VAF = 33 * 0.01 + 0.65 = 0,98$

$PFA = PF * VAF$  où PAF est le nombre de point de fonction non ajusté

$PFA = 59 * 0,98 = 57,82$

#### 5) Nombre de ligne de code

*Tableau V.7: Conversion de points de fonction aux lignes de code*

Langage	SLOC par point de fonction	Langage	SLOC par point de fonction
C	128	APL	32
C++	52	2GL	107
Java	46	4GL	20
Cobol	107	Visual Basic	24
ADA	41	Delphi	18
Lisp	64	HTML4	14
Assembleur	320	SQL	13
Basic	107		

Notre projet a été développé en deux langages différents, notamment le C++ et le langage arduino, comme Arduino ne figure pas sur la liste du fait qu'arduino est un langage récent et cette liste avait été élaborée avant l'existence arduino, pour cela nous allons utiliser la même donnée que celle du langage Java.

Nous allons subdiviser notre point de fonction en deux en fonction de ces deux langages de programmation (C++ et arduino):

- Le nombre de points de fonction non ajusté en C++ vaut : 44
- Le nombre de points de fonction non ajusté en Arduino vaut : 34

D'où le PFA pour le C++ =  $34 * 0,98 = 33,32$  et le PFA pour Arduino =  $25 * 0,98 = 25,5$

Pour réaliser notre application il faut :

- ✓ 2242, 24 lignes de code pour le C++ soit  $52 * 33,32$  ;
- ✓ 1532, 72 lignes de code pour le Arduino soit  $46 * 25,5$ ;

En faisant la somme des lignes de code nous obtenons : **2 906, lignes de code** soit  $1\ 732,64 + 1\ 173$ .

- 6) Calcul de l'effort total et la durée de développement **avec le modèle** d'estimation COCOMO basique avec 12KDSI

Nous savons que :

**Tableau V.8: Estimation COCOMO de base**

Type de projet	Effort Homme Mois	Temps de développement
Organique	$2,4 \text{ (Kloc)}^{1,05}$	$2,5 \text{ (HM)}^{0,38}$
Médian	$2,4 \text{ (Kloc)}^{1,12}$	$2,5 \text{ (HM)}^{0,35}$
Imbriqué	$2,4 \text{ (Kloc)}^{1,20}$	$2,5 \text{ (HM)}^{0,32}$

Le modèle COCOMO s'applique à trois types de projets informatiques:

–**Organiques** : Application simple, de routine, réalisée par une équipe expérimentée ayant l'habitude de travailler ensemble, maîtrisant le langage et l'environnement de développement.

–**Semi-détachés** Niveau intermédiaire, le projet n'est ni trop simple ni trop compliqué, l'équipe de développement a déjà réalisé quelques projets ensemble mais n'est pas totalement rodée. Les technologies et le domaine d'application sont un peu flous, mais pas de grosses difficultés.

–**Imbriqué** Techniques innovantes, organisation complexe, couplage fort avec beaucoup d'interactions. Technologie et domaine nouveau, équipe 'jeune'.

En considérant que notre projet est Médian. Les paramètres pour le calcul de l'effort et de la durée suivant COCOMO basique sont :

**HM COCOMO basique** =  $2.4 * (\text{KDSI})^{1,12}$  avec KDSI : taille en millier des codes sources livrées

**D'où HM COCOMO basique** =  $2.4 * (2,90)^{1,12} = 7,91 \text{ HM}$

**TDEV<sub>COCOMO basique</sub>** =  $2,5 \times (\text{HM})^{0,35} = 2,5 \times (7,91)^{0,35} = 5,16 \text{ Mois}$

$$Productivité = \frac{2\,906\,DSI}{7,91\,HM} = 524,55\,DSI/HM$$

$$FSP = \frac{7,91\,HM}{5,16\,Mois} = 1,53\,FSP$$

i) **Distribution de l'effort par phase en pourcentage**

*Tableau V.9: Distribution de l'effort par phase en pourcentage*

Type de Projet Médian	3,78 KLOC
Conception globale et plan d'intégration (RPD)	16
Conception Détaillée (DD)	26
Programmation et Tes Unitaire (CUT)	38
Intégration (IT)	20

Le tableau ci-dessus (tableau V.9) nous représente une répartition d'effort pour chaque phase de développement du logiciel en pourcentage.

ii) **Distribution de temps de développement par phase en pourcentage**

**Table V.10: Distribution de temps de développement par phase en pourcentage**

Type de Projet Médian	3,78 KLOC
Conception globale et plan d'intégration (RPD)	17
DD et CUT	63
Intégration (IT)	20

Conformément au tableau de Distribution de l'effort par phases en pourcentage on a :

- Phase de conception =  $0,16 \times 7,91 = 1,27\,HM$
- Phase de codage =  $0,64 \times 7,91 = 5,06\,HM$
- Phase d'intégration =  $0,20 \times 7,91 = 1,58\,HM$

Conformément au tableau de Distribution du temps de développement par phases en pourcentage on a :

- Phase de conception =  $0,17 \times 5,16 = 0,88\,mois$
- Phase de codage =  $0,63 \times 5,16 = 3,25\,mois$
- Phase d'intégration =  $0,20 \times 5,16 = 1,03\,mois$

**Tableau V.11: Synthèse du cout Logicielle**

Phase	HM en Homme	TDEV en mois	Salaire Individu en USD	Total
Conception	1	1	200	200
Codage	5	3	200	3 000
Intégration	2	1	200	400
<b>Total en USD</b>				<b>2 000</b>

Le tableau ci-dessus (Tableau V.11) donne le coût total du logiciel qui est estimé à **3 600 USD**

## V.5. CONCLUSION PARTIELLE

Dans ce chapitre, nous avons présenté différentes techniques, outils matériels et logiciels nécessaire pour la mise en place de notre système; ensuite nous avons estimé le coût matériel et logicielle de notre système qui est estimé à **4 633 USD** (dont **3 600 USD** pour le coût du logicielle et **1 033 USD** pour le coût matériel).

## CHAP VI. CONCLUSION GENERALE ET PERSPECTIVES

Nous voici au terme de notre travail qui a porté sur le « *contrôle et surveillance du trafic routier urbain* ».

Au fil de ce document, nous avons vu plusieurs aspects de la gestion du trafic routier urbain et par la suite nous avons présenté et implémenté un système qui réalise automatiquement et en temps réel le contrôle et la surveillance du trafic routier urbain. En fusionnant les résultats de plusieurs approches complémentaires, ce système implémenté donne un ensemble des pistes des solutions face aux problèmes rencontrés dans la gestion du trafic routier urbain (congestion du trafic, feux de circulation, tracasseries, accidents, véhicule d'urgence,...). Ainsi nous pouvons qualifier ce système d'un :

- ✓ système qui utilise la technique de vision par ordinateur pour réguler le feu de circulation d'une manière intelligente, adaptative et efficace en prenant des images en temps réelle de l'état du trafic à une jonction;
- ✓ système de contrôle de feu de circulation intégrant la gestion des véhicules d'urgence ou prioritaire par utilisation de la technologie GSM;
- ✓ système de détection du dépassement de la vitesse limite ainsi que celui de détection des accidents observé au niveau de la voie de circulation ;
- ✓ système de détection d'infraction au niveau du feu de circulation et sur la voie de circulation.

Certes ce système reste perfectible mais il pourrait constituer une base de réflexion pour la promotion de solutions informatiques.

Pour parvenir à implémenter notre système de contrôle et surveillance du trafic urbain, nous sommes passés par plusieurs phases :

La première phase consistait d'abord à présenter un état de l'art en rapport avec notre thème de recherche; dans cette phase nous avons d'abord analysé et montré les quelques travaux et techniques utilisées par d'autres auteurs dans le domaine du trafic routier, ensuite nous avons énoncé notre approche dont celui de l'utilisation de la vision par ordinateur, d'un module GSM et des quelques capteurs à ultrasons HC-SR04 pour assurer un bon contrôle et une bonne surveillance du trafic routier urbain contrairement à d'autres travaux qui utilisent des capteurs incorporés dans le véhicule et sur la voie de circulation.

La deuxième phase quand à elle consistait tout d'abord à introduire quelques notions importantes sur le trafic routier urbain et sur quelques systèmes régulation du trafic existant; en suite nous avons aussi parlé des différents capteurs de mesures usuelle du trafic. De cela nous avons montré que le capteur vidéo est un capteur fiable et très peu couteux. Nous avons opté pour l'utilisation d'un capteur vidéo qui est adapté aux exigences et objectifs de notre travail. Ainsi, nous avons conclu qu'un système basé sur la vision par ordinateur nous semble beaucoup plus logique si on envisage implémenter une technique à la fois rapide, flexible et moins chère.

Dans la troisième phase, nous avons commencé d'abord par une étude conceptuelle, dans cette partie nous avons présenté et décrit l'architecture matériel de notre système; ensuite nous avons présenté la logique de fonctionnement (logiciel, matériel, mesure de sécurité) du système. Par la suite nous avons entamé l'étape d'analyse fonctionnelle et modélisation du système; dans cette étape nous avons réalisé une description graphique et textuelle de l'application (du système) en faisant recours au diagramme de cas d'utilisation, d'activité et aussi aux méthodes d'analyse et conception d'application contrôle-comment de SA-RT et DARTS.

Dans la dernière phase, nous avons abordé l'étape d'expérimentation du système temps réelle. Pour cela, nous avons implémenté à l'aide du langage C++ et une Librairie OpenCv, un CPU qui est en charge du traitement d'images. Le reste des fonctionnalités du système ont été implémentée dans un microcontrôleur de la plateforme Arduino MEGA.

Suite aux éléments de nos hypothèses et vue les résultats des différentes phases énoncées ci-haut, nous confirmons nos hypothèses avec certitude que :

- ✓ La mise en place d'un système de régulation de feu de circulation d'une manière intelligente, adaptative et efficace par vision d'ordinateur permet donc d'assurer une meilleure gestion de feux de circulation, d'améliorer l'expérience des usagers et réduire les coûts globaux.
- ✓ La mise en place d'un système de gestion de véhicule d'urgence ou prioritaire ainsi que l'intégration d'un module de détection d'infraction, de détection d'accident et celui de la détection du dépassement de la vitesse limite observé sur la voie de circulation contribue donc à la surveillance et au contrôle du trafic routier.

Comme perspective, nous proposons de continuer par le développement et l'intégration d'un module de détection des plaques d'immatriculation ainsi que celui d'une recherche exhaustive d'un véhicule suspect ou en poursuite par le service de la police de circulation pour essayer de renforcer la fréquence de contrôle et de surveillance du trafic routier.

En fait, à la fin de la réalisation de ce travail de fin d'étude d'ingénieur, nous avons accumulé une masse importante de connaissances aussi bien sur le plan théorique que sur le plan pratique; et nous estimons qu'elles nous seront très utiles à l'avenir, dans nos expériences et études ultérieures.

# BIBLIOGRAPHIE

## A. REFERENCES :

- [1] F. Sébastien, «Contrôle et gestion du trafic routier urbain par un réseau de capteur sans fils,» Paris telecom, Paris, pp.2-5, 2014.
- [2] F. KSONTIN, *Modèle d'agent fondé sur les affordances : application à la simulation de trafic routier*, Université de VALENCIENNES ET DU HAINAUT-CAMBRESIS, pp. 6-19, 2013.
- [3] J.-C. Tai, S.-T. Tseng, C.-P. Lin et K.-T. Song, «Real-time image tracking for automatic traffic monitoring and enforcement applications,» *IJART*, vol. 3, n° 25, pp. 5-7, 2016.
- [4] B. Zhou, «Adaptive Traffic Light Control in Wireless Sensor Network-Based Intelligent Transportation System,» *Vehicular Technology Conference Fall*, vol. 2, n° 72, p. 3, 2010.
- [5] V. Patil, «Intelligent Traffic Control System,» *ICIET*, vol. 2, n° 144, p. 4, 2016.
- [6] S. Brand et S. Stalin, «Design of Intelligent Traffic Control System,» *IEEE*, vol. 1, n° 1345, pp. 2-4, 2016.
- [8] T. Cherif et L. Dimitri, «Des carrefours Intelligent pour une meilleur gestion du trafic urbain,» *VCRE*, vol. 2, n° 141, pp. 2-4, 2015.
- [9] AFNOR, «Guide de l'Association Française de Normalisation,» 1991, p. 12.
- [10] T. Yang et S. Li, «Real-time and accurate segmentation of moving objects in dynamic scene,» New York, 2004, pp. 5-10.
- [11] C. Wren et A. Azarbayejani, «Real-time tracking of the human body,» *ICJR*, p. 4, 1997.
- [12] N. Oliver et B. Rosario, «A Bayesian computer vision system for modeling human interactions,» *IEEE Trans*, pp. 3-5, 2000.
- [13] N. Verbeke, *Suivi d'objets en mouvement dans une séquence vidéo*, vol. Thèse en Informatique, Université Paris Descartes, pp.25-35, 2007.
- [14] A.-C. MAHEO, *Méthodes de suivi d'un objet en mouvement*, Institut Supérieur de l'électronique et du Numérique, pp. 35-44, 2015.
- [16] B. FARAH, «Reconnaissance Automatique de Logo de Voitures,» Université des Sciences et de la Technologie d'Oran Mohamed BOUDIAF, pp. 25-36, 2010.
- [17] S. Landrault et W. Henry, *Arduino : Premier pas en Informatique Embarquée*, Paris: Creative

Commons, pp. 10-18, 2014.

- [19] J. Nussey, *Arduino pour les nuls*, Paris: Editions First, pp. 45-76, 2017.
- [21] Y.Fei, «Contribution à la modélisation et à la régulation du trafic aux intersections : intégration des communications Véhicule-Infrastructure,» Université de Technologie de Belfort-Montbéliard, pp. 10-16, 2012.
- [22] S. Debashish, *Statistical physics of vehicular traffic and some related systems*, pp. 24-28, 2012.
- [23] B. Kerne, *Phase transitions in traffic flow*, Springer, pp.28-30, 2000.
- [24] J. Shibata et T. Yamamoto “Detection and control of congestion in urban road networks”. Dans : *Traffic engineering & control* , pp. 438–444, 1984.
- [25] C. S, *Ingénierie du Trafic Routier : Eléments de théorie du trafic et applications*, Ecole National des Ponts et Chaussées, 1993.
- [26] T. Kamata, «Concise encyclopedia of traffic and transportation systems,» chez *Detectors for road traffic pergamon Press*, pp. 17-20, 2009.
- [27] W. M. e. X. Yang, «Design and Evaluation of an Adaptive Bus Signal Priority System Base on Wireless Sensor Network,» p. 4, 2008.
- [28] W. Kavya et S. Jyothi, «Trafic Light Control Sytsem Using Image Processing,,» *IJIRCCCE*, vol. 2, n° 1,Special Issue 5, p.5, 2014.
- [29] W. Daywane, «Performance evaluation of Los Angeles adaptive traffic control system (ATCS) on an arterial corridor,» Los Angeles, pp. 38-45, 2012.
- [30] R. Gordon, *Traffic control systems handbook*, US Dept. of Transportation: Federal Highway Administration,p. 26, 2005.
- [31] J. Fullerton, «Manual of Traffic Signal Design,» *Englewood Cliffs*, vol. 3, pp. 33-38, 1998.
- [32] O. F., “The Traffic Signal Book. Englewood Cliffs, PrenticeHall, p.28, 1993.
- [33] L. Chen et Powers F, «Evaluation From Precison, Recall, and F-Factor to ROC,Informedness, Markedness & Correlation,» vol. 3, p. 12, 2007.
- [36] C. Francis et G. Emmanuell, *SYSTÈMES TEMPS RÉEL DE CONTRÔLE-COMMANDE: Conception et implémentation*, Paris: Dunod, pp. 57-70, 2005.
- [37] C. MBUYAMBA, *cours de Génie Logiciel*, FSTA-ULPGL, TECH 1, 2015.
- [38] P. Viola et J. Michael, *Robust Real-Time Face Detection*, 2001.

[39] V. Paul et J. Michael, «Rapid Object Detection using a Boosted Cascade of Simple Features,» *IEEE*, vol.2, 2001.

## **B. LIENS :**

[7] «Les transports intelligents» [En ligne]. Available: <http://www.transport-intelligent.net/>. [Accès le 25 septembre 2017].

[15] [En ligne]. Available: <http://www.clashinfo.com/dico/definition-p/art7-pixel.html>. [Accès le 15 novembre 2017].

[18] [En ligne]. Available: <https://www.bannerengineering.com/fr/fr/company/expert-insights/ultrasonic-sensors-101.html>. [Accès le 24 février 2018].

[20] «Arduino Mega 2560 Rev3,» [En ligne]. Available: <https://store.arduino.cc/usa/arduino-mega-2560-rev3>. [Accès le 22 janvier 2018].

[34] «[https://www.youtube.com/watch?v=6ixAA\\_eZ51U](https://www.youtube.com/watch?v=6ixAA_eZ51U),» [En ligne]. [Accès le 27 janvier 2018].

[35] [En ligne]. Available: <https://www.youtube.com/watch?v=WLvCVzd6Pn4>. [Accès le 25 janvier 2018].

[40] «IR Sensor» [En ligne]. Available: <http://www.electronicshub.org/ir-sensor/>. [Accès le 20 décembre 2017].

# ANNEXE A

---

## ALGORITHME DE VIOLA ET JONES

---

Une avancée majeure dans le domaine a été réalisée par les chercheurs Paul Viola et Michael Jones en 2001. Ces derniers ont proposé une méthode basée sur l'apparence (*Appearance-based methods*) [38].

La méthode de Viola et Jones est une méthode de détection d'objet dans une image numérique, elle fait partie des toutes premières méthodes capables de détecter efficacement et en temps réel des objets dans une image. Inventée à l'origine pour détecter des visages, elle peut également être utilisée pour détecter d'autres types d'objets comme es voitures ou des avions. La méthode de Viola et Jones est l'une des méthodes les plus connues et les plus utilisées, en particulier pour la détection de visages et la détection de personnes. En tant que *procédé d'apprentissage supervisé*, la méthode de Viola et Jones nécessite de quelques centaines à plusieurs milliers d'exemples de l'objet que l'on souhaite détecter, pour entraîner un classifieur. Une fois son apprentissage réalisé, ce classifieur est utilisé pour détecter la présence éventuelle de l'objet dans une image en parcourant celle-ci de manière exhaustive, à toutes les positions et dans toutes les tailles possibles. Considérée comme étant l'une des plus importantes méthodes de détection d'objet, la méthode de Viola et Jones est notamment connue pour avoir introduit plusieurs notions reprises ensuite par de nombreux chercheurs en vision par ordinateur, à l'exemple de la notion d'image intégrale ou de la méthode de classification construite comme une cascade de classifieurs boostés.

Cette méthode bénéficie d'une implémentation sous licence BSD dans OpenCV, la bibliothèque utilisée dans notre application.

### Principe

La méthode de Viola et Jones consiste à balayer une image à l'aide d'une fenêtre de détection de taille initiale 24px par 24px (dans l'algorithme original) et de déterminer si un visage y est présent. Lorsque l'image a été parcourue entièrement, la taille de la fenêtre est augmentée et le balayage recommence, jusqu'à ce que la fenêtre fasse la taille de l'image. L'augmentation de la taille de la fenêtre se fait par un facteur multiplicatif de 1.25. Le balayage, quant à lui, consiste simplement à décaler la fenêtre d'un pixel. Ce décalage peut être changé afin d'accélérer le processus, mais un décalage d'un pixel assure une précision maximale.

Cette méthode est une approche basée sur l'apparence, qui consiste à parcourir l'ensemble de l'image en calculant un certain nombre de caractéristiques dans des zones rectangulaires qui se chevauchent. Elle a la particularité d'utiliser des caractéristiques très simples mais très nombreuses.

Il existe d'autres méthodes mais celle de Viola et Jones est la plus performante à l'heure actuelle. Ce qui la différencie des autres est notamment :

- l'utilisation **d'images intégrales** qui permettent de calculer plus rapidement les caractéristiques.
- la **sélection par boosting** des caractéristiques.
- la combinaison en **cascade de classifieurs** boostés, apportant un net gain de temps d'exécution.

### Apprentissage du classifieur

Une étape préliminaire et très importante est l'apprentissage du classifieur. Il s'agit d'entraîner le classifieur afin de le sensibiliser à ce que l'on veut détecter, ici des visages. Pour cela, il est mis dans deux situations :

La première où une énorme quantité de cas positifs lui sont présentés et la deuxième où, à l'inverse, une énorme quantité de cas négatifs lui sont présentés. Concrètement, une banque d'images contenant des visages de personnes est passée en revue afin d'entraîner le classifieur. Ensuite, une banque d'images ne contenant pas de visages humains est passée.

Dans le cas présent, Viola et Jones ont entraîné leur classifieur à l'aide d'une banque d'images du MIT. Il en résulte un classifieur sensible aux visages humains. Il se présente sous la forme d'un fichier XML.

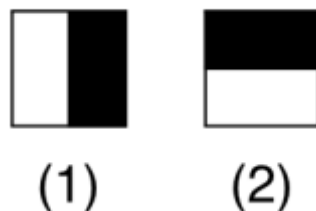
Dans l'absolu, on serait en mesure de détecter n'importe quel signe distinctif à partir d'un classifieur entraîné à cela.

### Les caractéristiques

*Une caractéristique est une représentation synthétique et informative, calculée à partir des valeurs des pixels.* Les caractéristiques utilisées ici sont les caractéristiques pseudo-haar. Elles sont calculées par la différence des sommes de pixels de deux ou plusieurs zones rectangulaires adjacentes.

*Prenons un exemple :* Voici deux zones rectangulaires adjacentes, la première en blanc, la deuxième en noire.

Les caractéristiques seraient calculées en soustrayant la somme des pixels noirs à la somme des pixels blancs. Les caractéristiques sont calculées à toutes les positions et à toutes les échelles dans une fenêtre de détection de petite taille, typiquement de 24x24 pixels ou de 20x15 pixels. Un très grand nombre de caractéristiques par fenêtre est ainsi généré, Viola et Jones donnant l'exemple d'une fenêtre de taille 24 x 24 qui génère environ 160 000 caractéristiques



**Figure ANNEXE-A. 1: Caractéristiques pseudo-haar à seulement deux caractéristiques**

L'image précédente présente des caractéristiques pseudo-haar à seulement deux caractéristiques mais il en existe d'autres, allant de 4 à 14, et avec différentes orientations. Malheureusement, le calcul de ces caractéristiques de manière *classique* coûte cher en terme de ressources processeur, c'est là qu'interviennent les images intégrales.

## L'image intégrale

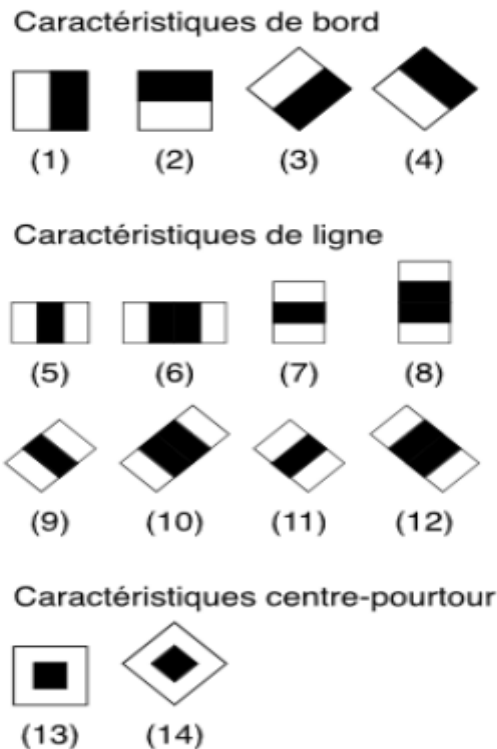
Pour calculer rapidement et efficacement ces caractéristiques sur une image, les auteurs proposent également une nouvelle méthode, qu'ils appellent *image intégrale*. C'est une représentation sous la forme d'une image, de même taille que l'image d'origine, elle contient en chacun de ses points la somme des pixels situés au-dessus et à gauche du pixel courant. Plus formellement, l'image intégrale  $ii$  au point  $(x; y)$  est définie à partir de l'image  $i$  par :

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y')$$

Le calcul de la somme des valeurs des pixels appartenant à une zone rectangulaire s'effectue donc en accédant seulement à quatre pixel de l'image intégrale : soit un rectangle  $ABCD$  dont les sommets sont nommés dans le sens des aiguilles d'une montre en commençant par le sommet supérieur gauche et soit  $x$  la valeur sous la représentation intégrale d'un sommet  $X$  du rectangle ( $X \in \{A; B; C; D\}$ ). La somme des valeurs des pixels appartenant à  $ABCD$  est, quelle que soit sa taille, donnée par  $c - b - d + a$ . Une caractéristique de Haar étant une combinaison linéaire de tels rectangles  $ABCD$ , son calcul se fait alors en un temps indépendant de sa taille.

## Algorithme d'apprentissage basé sur Adaboost

Le deuxième élément clé de la méthode de Viola et Jones est l'utilisation d'une méthode de boosting afin de sélectionner les meilleures caractéristiques. Le boosting est un principe qui consiste à construire un classifieur *fort* à partir d'une combinaison pondérée de classifieurs *faibles*, c'est-à-dire, donnant en moyenne une réponse meilleure qu'un tirage aléatoire. Viola et Jones adaptent ce principe en assimilant une caractéristique à un classifieur faible, en construisant un classifieur faible qui n'utilise qu'une seule caractéristique. L'apprentissage du classifieur faible consiste alors à trouver la valeur seuil de la caractéristique qui permet de mieux séparer les exemples positifs des exemples négatifs. Le classifieur se réduit alors à un couple (caractéristique, seuil).



**Figure ANNEX-A.:2: caractéristiques pseudo-haar avec différentes orientations**

L'algorithme de boosting utilisé est en pratique une version modifiée d'AdaBoost, qui est utilisée à la fois pour la sélection et pour l'apprentissage d'un classifieur *fort*. Les classifieurs faibles utilisés sont souvent des arbres de décision. Un cas remarquable, fréquemment rencontré, est celui de l'arbre de profondeur 1, qui réduit l'opération de classification à un simple seuillage.

L'algorithme est de type itératif, à nombre d'itérations déterminé. À chaque itération, l'algorithme sélectionne une caractéristique, qui sera ajoutée à la liste des caractéristiques sélectionnées aux itérations précédentes, et le tout va contribuer à la construction du classifieur fort final. Cette sélection se fait en entraînant un classifieur faible pour toutes les caractéristiques et en sélectionnant celui avec l'erreur la plus faible sur l'ensemble d'apprentissage.

L'algorithme tient également à jour une distribution de probabilité sur l'ensemble d'apprentissage, réévaluée à chaque itération en fonction des résultats de classification. En particulier, plus de poids est attribué aux exemples difficiles à classer, c'est-à-dire ceux dont l'erreur est élevée. Le classifieur *fort* final construit par AdaBoost est composé de la somme pondérée des classifieurs sélectionnés.

Plus formellement, on considère un ensemble de  $n$  images  $(x_1, \dots, x_n)$  et leurs étiquettes associées  $(y_1, \dots, y_n)$ , qui sont telles que  $y_i = 0$  si l'image  $x_i$  est un exemple négatif et  $y_i = 1$  si  $x_i$  est un exemple de l'objet à détecter. L'algorithme de boosting est constitué d'un nombre  $T$  d'itérations, et pour chaque itération  $t$  et chaque caractéristique  $j$ , on construit un classifieur faible  $h_j$ . Idéalement, le but est d'obtenir un classifieur  $h$  qui prédise exactement les étiquettes pour chaque échantillon, c'est-à-dire  $y_i = h(x_i) \forall i \in \{1, \dots, n\}$ . En pratique, le classifieur n'est pas parfait et l'erreur engendrée par ce classifieur est donnée par :

$$\epsilon_j = \sum_{i=1}^n \omega_i |h_j(x_i) - y_i|$$

les  $w_i$  étant les poids associés à chaque exemple et mis à jour à chaque itération en fonction de l'erreur obtenue à l'itération précédente. On sélectionne alors à l'itération  $t$  le classifieur  $h_t$  présentant l'erreur la plus faible :  $\epsilon_t = \min(\epsilon_j)$ . Le classifieur fort final  $h(x)$  est construit par seuillage de la somme pondérée des classifieurs faibles sélectionnés :

$$h(x) = \begin{cases} 1 & \text{si } \sum_{t=1}^T \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t \\ 0 & \text{sinon} \end{cases}$$

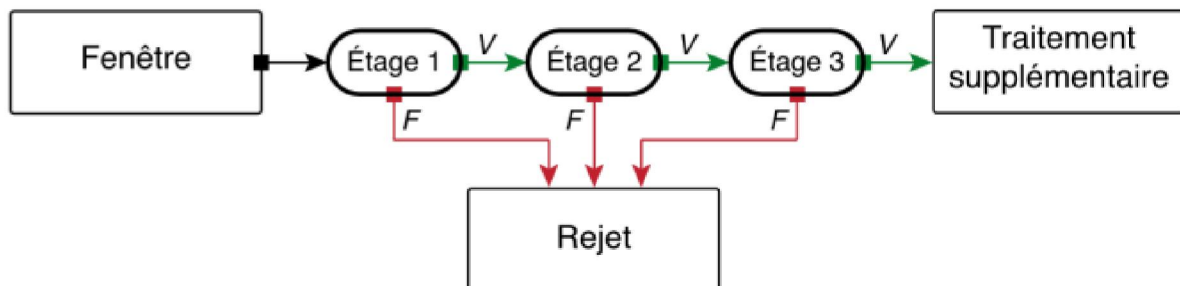
Les  $\alpha_t$  sont des coefficients calculés à partir de l'erreur  $\epsilon_t$  :

$$\alpha_t = \frac{1}{2} \ln \frac{1 - \epsilon_t}{\epsilon_t}$$

### Cascade de classifieurs

La méthode de Viola et Jones est basée sur une approche par recherche exhaustive sur l'ensemble de l'image, qui teste la présence de l'objet dans une fenêtre à toutes les positions et à plusieurs échelles. Cette approche est cependant extrêmement coûteuse en calcul. L'une des idées-clés de la méthode pour réduire ce coût réside dans l'organisation de l'algorithme de détection en une cascade de classifieurs.

Appliqués séquentiellement, ces classifieurs prennent une décision d'acceptation ; la fenêtre contient l'objet et l'exemple est alors passé au classifieur suivant, ou de rejet ; la fenêtre ne contient pas l'objet et dans ce cas l'exemple est définitivement écarté. L'idée est que l'immense majorité des fenêtres testées étant négatives (c.-à-d. ne contiennent pas l'objet), il est avantageux de pouvoir les rejeter avec le moins possible de calculs. Ici, les classifieurs les plus simples, donc les plus rapides, sont situés au début de la cascade, et rejettent très rapidement la grande majorité des exemples négatifs. Cette structure en cascade peut également s'interpréter comme un arbre de décision dégénéré, puisque chaque nœud ne comporte qu'une seule branche [39].



## ANNEXE B

---

### EXTRAIT DU CODE CPU

---

```
#include<opencv2/core/core.hpp>

#include<opencv2/highgui/highgui.hpp>

#include<opencv2/opencv.hpp>

#include<opencv2/objdetect/objdetect.hpp>

#include "opencv2/objdetect/objdetect.hpp"

#include "opencv2/highgui/highgui.hpp"

#include "opencv2/imgproc/imgproc.hpp"

#include <iostream>

#include <stdio.h>

#include <fstream> // for File

#include <thread>

#include <mutex>

using namespace std;

using namespace cv;

int tin(int sizeC){

    ofstream fs("K:/OpenCv/DataSetCar/RoadA.txt");

    if (fs.is_open()) {

        fs<<sizeC;

        fs.close();

    }else {

        cout <<"Enable To Open The File"<<endl;

    }

}
```

```

    return 0;
}

int printHome(){
    time_t now = time(0);

    // convert now to string form
    cout<<"-----"<<endl;

    cout<<"    Traffic Highway System @2018"<<endl;

    cout<<"    Accident Detection System @2018"<<endl;

    cout<<" @ Powered by Bienfait RUKUNDO [ BIN-PC ]-ULPGL "<<endl;

    cout<<"-----"<<endl;

    cout<<"-Date System :"<< ctime(&now)<<endl;

    cout<<"*****"<<endl<<endl;

    return 0;
}

// Scalar(0, 0, 255) which means Red in BGR

int main( int argc, char **argv )
{
    time_t now = time(0);

    printHome();

    const string sourceReference = "K:/OpenCv/DataSetCar/AC12.mp4";

    int nCars=0;

    int saveS=0;

/ //Load the cascade classifier file

    CascadeClassifier cascadeDetect;

// string cascadeFile = "K:/OpenCv/DataSetCar/AC8.xml";

    string cascadeFile = "K:/OpenCv/DataSetCar/AC10.xml";

    if (!cascadeDetect.load(cascadeFile)) {

        cerr << "Error loading the classifier file" << endl;

```

```

return 0;
}
// Open the video file
cv::VideoCapture capture(sourceReference);
// check if video successfully opened
if (!capture.isOpened()){
    cout << "Error loading the image that was passed in" << endl;
    return 1;
}
// Get the frame rate
double rate= capture.get(CV_CAP_PROP_FPS);
bool stop(false);
cv::Mat frame; // current video frame
cv::namedWindow("Extracted Frame",WINDOW_AUTOSIZE);
// cv::namedWindow("Extracted Frame",WINDOW_AUTOSIZE);//cv2.resizeWindow('image',
600,600)
// Delay between each frame in ms
int delay= 1000/rate;
// for all frames in video
while (!stop) {

// read next frame if any
if (!capture.read(frame))
break;

//Look for matches and store them into our vector
vector<Rect> detected;

cascadeDetect.detectMultiScale(frame, detected, 1.1, 2, 0, Size(35,35));

```

```

// minSize – Minimum possible object size. Objects smaller than that are ignored. maxSize –
Maximum possible object size. Objects larger than that are ignored.

// Draw circles on the detected faces

for( int i = 0; i < detected.size(); i++ )
{
    Point center( detected[i].x + detected[i].width*0.5, detected[i].y + detected[i].height*0.5
);

    ellipse( frame, center, Size( detected[i].width*0.5, detected[i].height*0.5), 0, 0, 360,
Scalar( 0, 0, 255 ), 2, 8, 0 );
}

cv::imshow("Extracted Frame",frame);

if (detected.size())>1) {

    saveS ++;//initialize Counter after 3 Frame then I can chage the Value

    // cout<<"i="<<saveS;

    // cout << "Car N => " << detected.size() << endl;

    cout << "Accident detected => " << detected.size() <<"at MABANGA on [ "<<
ctime(&now)<<"]"<<endl;

}

waitKey(0);

return 1;

}

```

## ANNEXE C

---

### EXTRAIT DU CODE Microcontrôleur (ATMega2560)

---

```
#include <SoftwareSerial.h>

#include <LiquidCrystal.h>

#include <avr/wdt.h>

int g1 = 53,y1 = 52,r1 = 51;

int g2 = 24,y2 = 23,r2 = 22;

int g3 = 14,y3 = 15,r3 = 16;

int g4 = 13,y4 = 12,r4 = 11,ambulance1=7,ambulance2=10;

int btn13=20,btn24=21;

int counter1=1,counter2=1,timeSec13=0,timeSec24=0;

int buttonState24 = LOW,buttonState13=LOW,oldVal24=LOW,oldVal13=LOW;

long randSize13=0,randSize24=0,randTime13=0,randTime24=0;//random Data

// the setup routine runs once when you press reset:

void setup() {

  // initialize the digital pin as an output.

  pinMode(g1, OUTPUT); //LINE 1

  pinMode(y1, OUTPUT);

  pinMode(r1, OUTPUT);

  pinMode(g2, OUTPUT); //LINE 2

  pinMode(y2, OUTPUT);

  pinMode(r2, OUTPUT);

  pinMode(g3, OUTPUT); //LINE 3

  pinMode(y3, OUTPUT);
```

```

pinMode(r3, OUTPUT);

pinMode(g4, OUTPUT); //LINE 4

pinMode(y4, OUTPUT);

pinMode(r4, OUTPUT);

//ambulance signal

pinMode(ambulance1, OUTPUT);

pinMode(ambulance2, OUTPUT);

pinMode(btn13, INPUT);

pinMode(btn24, INPUT);

}

// the loop routine runs over and over again forever:

void loop() {

//First rando Data

randSize13=random(0, 40);

randSize24=random(0, 40);

if(randSize13 >=0 &&randSize13 <15){

// randSize13=3000;

randTime13=300;

} else if(randSize13 > 16 &&randSize13 < 41){

// randSize13=6000;

randTime13=6000;

} //Line 13

//Line 24

if(randSize24 > 1 &&randSize24 <15 ){

//randSize24=3000;

randTime24=3000;

} else if( randSize24 > 16 &&randSize24 <41 ){

```

```

//randSize24=6000;

randTime24=6000;

}

starLight(HIGH,oldVal13,oldVal24,randTime13,randTime24);// call Main Data

}

void starLight(int statusData,int oldVal13,int oldVal24,int randTime13,int randTime24){

Serial.println("Press 1 for Start/reset, 2 for elapsed time");

digitalWrite(ambulance1, LOW);

digitalWrite(ambulance1, LOW);

//Init LINE 1-Line 3 // is same Like LINE 1

digitalWrite(g1, HIGH); // turn the LED on (HIGH is the voltage level)

digitalWrite(g3, HIGH);

//Line 2/4

digitalWrite(r2, HIGH); // turn the LED on (HIGH is the voltage level)

digitalWrite(r4, HIGH);

// making a delay of 15 sec evry Time to check value

delay(2000); // must be 15 Sec

//MAking a loop to detect Emergancy evry 10 sec

int divVar13=randTime13/1000,divVar24=randTime24/1000;

for(int i=0;i<divVar13;i++){

    emergancyDetect(24,oldVal13,oldVal24,randTime13,randTime24    }

//==Yello LED Manager End Line 1-3

digitalWrite(g1, LOW);

digitalWrite(g3, LOW);

//line 2/4

digitalWrite(r2, LOW);

digitalWrite(r4, LOW);

```

```

digitalWrite(y1, HIGH);
digitalWrite(y3, HIGH);
//Line 2/4
digitalWrite(y2, HIGH);
digitalWrite(y4, HIGH);
delay(5000); //End Yellow Manager Line1-3

/** Line 2-4 Init
digitalWrite(y1, LOW);
digitalWrite(y3, LOW);
//Line 2/4
digitalWrite(y2, LOW);
digitalWrite(y4, LOW);
digitalWrite(r1, HIGH);
digitalWrite(r3, HIGH);
digitalWrite(g2, HIGH);
digitalWrite(g4, HIGH);
//LINE 2/4
delay(5000);
for(int i=0;i<divVar24;i++){
emergancyDetect(13,oldVal13,oldVal24,randSize13,randSize24    }
digitalWrite(r1, LOW);
digitalWrite(r3, LOW);
digitalWrite(g2, LOW);
digitalWrite(g4, LOW);
digitalWrite(y1, HIGH); // turn the LED on (HIGH is the voltage level)
digitalWrite(y3, HIGH);
//LINE 2 AND 4

```

```

digitalWrite(y2, HIGH); // turn the LED on (HIGH is the voltage level)

digitalWrite(y4, HIGH);

delay(5000); //Start Yellow Manager Line2-4

digitalWrite(y1, LOW);

digitalWrite(y3, LOW);

//LINE 2 AND 4

digitalWrite(y2, LOW);

digitalWrite(y4, LOW);

//End Yellow Manager Line2-4

}

void emergencyDetect(int frontSide,int odlVal13,int odlVal24,int randTime13,int randTime24){

if((digitalRead(btn13))==HIGH && odlVal13==LOW){//Read Emergency Signal emergency is at
//LINE 1-3

    digitalWrite(ambulance1, HIGH); //Activete LED

    oldVal13=HIGH;

    digitalWrite(g2, LOW); //

    digitalWrite(g4, LOW);

//Line 2/4

    digitalWrite(r2, LOW);

    digitalWrite(r4, LOW);

    digitalWrite(r1, LOW);

    digitalWrite(r3, LOW);

// making first Yello To All Front

    digitalWrite(y1, HIGH); // turn the LED on (HIGH is the voltage level)

    digitalWrite(y3, HIGH);

    digitalWrite(y2, HIGH);

    digitalWrite(y4, HIGH);

```

```

delay(1000);

digitalWrite(y1, LOW); // turn the LED on (HIGH is the voltage level)

digitalWrite(y3, LOW);

digitalWrite(y2, LOW);

digitalWrite(y4, LOW);

    digitalWrite(r2, HIGH);

    digitalWrite(r4, HIGH);

    digitalWrite(g1, HIGH);

    digitalWrite(g3, HIGH);

    delay(90000); ///time for the emergency Vehicule

    //turnig Off

    digitalWrite(r1, LOW);

    digitalWrite(r3, LOW);

digitalWrite(r2, LOW);

digitalWrite(r4, LOW);

    digitalWrite(g1, LOW);

    digitalWrite(g3, LOW);

    digitalWrite(ambulance1, LOW);

    }

else if(digitalRead(btn24)==HIGH && oldVal24==LOW){// emergency is at LINE 2-4

if((digitalRead(r2)==HIGH) | (digitalRead(r4)==HIGH)){

digitalWrite(ambulance2, HIGH);

oldVal24=HIGH;

        digitalWrite(g2, LOW);

        digitalWrite(g4, LOW);

digitalWrite(r2, LOW); //

digitalWrite(r4, LOW);

```

```

digitalWrite(g1, LOW);

digitalWrite(g3, LOW);

digitalWrite(y1, HIGH);

digitalWrite(y3, HIGH);

digitalWrite(y2, HIGH);

digitalWrite(y4, HIGH);

delay(5000);

digitalWrite(y1, LOW);

digitalWrite(y3, LOW);

digitalWrite(y2, LOW);

digitalWrite(y4, LOW);

    digitalWrite(r1, HIGH);

digitalWrite(r3, HIGH);

digitalWrite(g2, HIGH);

digitalWrite(g4, HIGH);

delay(2000); //time for the emergency Vehicule

    digitalWrite(r1, LOW);

    digitalWrite(r3, LOW);

    digitalWrite(g2, LOW);

    digitalWrite(g4, LOW);

digitalWrite(ambulance2, LOW);// Emergency LED

digitalWrite(btn24, LOW);// turn Off Emergency Signal

    }

else {

    delay(5000);

}

}

```