

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université de la Manouba
École Nationale des Sciences de L'Informatique



Rapport du Projet de Conception et de Développement

SUJET :

***Conception et Réalisation d'un Système de Surveillance
Basé sur le Web***

Élaboré par :

Mohamed Said Mezghani

Sous l'encadrement de :

M. Imed Abdessalem

Année Universitaire :

2012-2013

Signature de l'encadrant

**M. Imed Abdessalem
(ENSI)**

Remerciements

Nous tenons à adresser nos remerciements à notre encadrant M. Imed Abdessalem pour ses bonnes directives, sa disponibilité et pour ses conseils précieux qui nous ont permis de mener à bon terme notre projet de conception et de développement.

Qu'il nous soit permis de remercier également tous les membres du Club robotique ENSI pour leurs aides précieuses ainsi que toute l'équipe pédagogique de l' École Nationale des Sciences de l'Informatique qui ont assuré notre formation.

Nous remercions pareillement tous les membres de l'honorable jury d'avoir accepté de juger notre travail.

Nous exprimons également notre gratitude à toute personne qui a contribué d'une façon ou d'une autre à la bonne réalisation du projet.

Table des matières

| | |
|---|-----------|
| Introduction générale | 2 |
| 1 État de l'art | 3 |
| 1.1 La vidéo | 3 |
| 1.1.1 PAL/SECAM | 4 |
| 1.1.2 NTSC | 4 |
| 1.1.3 La vidéo numérique | 5 |
| 1.2 La vidéosurveillance | 5 |
| 1.2.1 La prévention de la criminalité | 5 |
| 1.2.2 La sécurité routière | 6 |
| 1.2.3 La sécurité industrielle | 7 |
| 1.3 Étude de l'existant | 8 |
| 1.3.1 Un serveur de caméra | 8 |
| 1.3.2 Une caméra réseau | 8 |
| 1.4 Travail demandé | 10 |
| 2 Analyse et Spécification des besoins | 11 |
| 2.1 Aperçu globale sur le projet | 11 |
| 2.2 Analyse des besoins | 12 |
| 2.2.1 Les acteurs du système | 12 |
| 2.2.2 Les besoins fonctionnels | 13 |
| 2.2.3 Les besoins non fonctionnels | 13 |
| 2.3 Spécification des besoins | 14 |
| 2.3.1 Diagramme de cas d'utilisation | 14 |
| 2.3.2 Les scénarios d'utilisation | 15 |

TABLE DES MATIÈRES

| | | |
|----------|--|-----------|
| 2.3.2.1 | Scénario d'utilisation 1 : Authentification | 15 |
| 2.3.2.2 | Scénario d'utilisation 2 : mode de fonctionnement normal | 16 |
| 2.3.2.3 | Scénario d'utilisation 3 : mode de fonctionnement assisté | 17 |
| 2.3.2.4 | Scénario d'utilisation 4 : mode de fonctionnement avec détection de mouvement | 17 |
| 2.3.2.5 | Scénario d'utilisation 5 : consultation des images enregistrées | 18 |
| 2.3.2.6 | Scénario d'utilisation 6 : changement des paramètres de connexion des utilisateurs | 19 |
| 3 | Conception globale | 20 |
| 3.1 | Conception globale de l'application de surveillance | 20 |
| 3.1.1 | Diagramme de composants de l'application | 21 |
| 3.1.2 | Diagramme de composant de la page | 21 |
| 3.2 | Diagramme de déploiement : Architecture matérielle du système | 23 |
| 4 | Conception et réalisation de la partie opérative | 25 |
| 4.1 | Conception détaillé du dispositif de surveillance | 25 |
| 4.1.1 | Conception du circuit d'alimentation | 25 |
| 4.1.2 | Conception du dispositif mécanique | 26 |
| 4.2 | Environnement matériel | 27 |
| 4.2.1 | Choix de l'environnement matériel | 27 |
| 4.2.1.1 | Raspberry Pi | 27 |
| 4.2.1.2 | Une webcam Logitech HD C310 | 28 |
| 4.2.1.3 | Deux servomoteurs Hitec HS-311 | 29 |
| 4.2.2 | Justification du choix de l'environnement matériel | 29 |
| 4.2.2.1 | Raspberry Pi | 30 |
| 4.2.2.2 | Webcam Logitech HD C310 | 30 |
| 4.2.2.3 | Servomoteur Hitec HS-311 | 31 |
| 4.3 | Réalisation de la partie opérative | 31 |
| 4.3.1 | Réalisation du circuit d'alimentation | 31 |
| 4.3.2 | Réalisation du dispositif mécanique | 32 |
| 5 | Conception et réalisation de la partie logicielle | 35 |
| 5.1 | Conception détaillée de l'application | 35 |

TABLE DES MATIÈRES

| | | |
|-----------------------------------|--|-----------|
| 5.1.1 | Diagramme de classe | 35 |
| 5.1.2 | Diagramme de séquences | 36 |
| 5.1.2.1 | Diagramme de séquences de l'authentification | 36 |
| 5.1.2.2 | Diagramme de séquences du mode d'utilisation normal | 37 |
| 5.1.2.3 | Diagramme de séquences du mode de fonctionnement assisté | 38 |
| 5.1.2.4 | Diagramme de séquences du mode d'utilisation avec détection du mouvement | 38 |
| 5.1.3 | Diagramme d'état/transition | 39 |
| 5.1.4 | Schéma de la base de donnée | 40 |
| 5.2 | Environnement Logiciel | 41 |
| 5.2.1 | Choix de l'environnement logiciel | 41 |
| 5.2.1.1 | La distribution Linux « Arch Linux ARM » | 41 |
| 5.2.1.2 | Node.js | 41 |
| 5.2.1.3 | MJPG-Streamer | 42 |
| 5.2.1.4 | Pi-Blaster | 42 |
| 5.2.2 | Justification du choix de l'environnement logiciel | 42 |
| 5.2.2.1 | La distribution Linux « Arch Linux ARM » | 42 |
| 5.2.2.2 | Node.js | 42 |
| 5.2.2.3 | Mjpg-Streamer | 43 |
| 5.2.2.4 | Pi-Blaster | 44 |
| 5.3 | Réalisation de l'interface de surveillance | 44 |
| 5.3.1 | Interface de connexion | 44 |
| 5.3.2 | Interface d'accueil | 45 |
| 5.3.3 | Interface de l'utilisateur | 45 |
| 5.3.3.1 | Mode de fonctionnement normal | 45 |
| 5.3.3.2 | Mode de fonctionnement assisté | 46 |
| 5.3.3.3 | Mode de fonctionnement avec détection du mouvement | 47 |
| 5.3.4 | Interface de l'administrateur | 49 |
| 5.3.5 | Interface d'aide | 50 |
| Conclusion et perspectives | | 51 |
| Bibliographie | | 52 |

TABLE DES MATIÈRES

| | |
|--|-----------|
| Netographie | 54 |
| A Les servomoteurs | 55 |
| A.1 Les servomoteurs | 55 |
| A.2 Commande des servomoteurs | 56 |
| B Régulateur de tension | 58 |
| B.1 Régulateur de tension | 58 |
| B.2 Avantage des régulateurs à découpage par rapport aux régulateurs linéaires . . . | 58 |
| B.3 Les régulateurs LM1576-ADJ | 59 |
| B.4 Le schéma du circuit d'alimentation | 59 |
| Résumé | 60 |

Table des figures

| | | |
|-----|---|----|
| 1.1 | Exemples de captures de vidéos dans le cas de prevention de criminalite | 6 |
| 1.2 | Cas d'application en circulation routière | 7 |
| 1.3 | Cas d'application en sécurité industrielle | 7 |
| 1.4 | Serveur Camera | 8 |
| 1.5 | Camera Réseau | 9 |
| 2.1 | Aperçu du travail à réaliser | 12 |
| 2.2 | Diagramme de cas d'utilisation | 15 |
| 2.3 | Diagramme de séquence : Authentification | 16 |
| 2.4 | Diagramme de séquence : mode de fonctionnement normal | 16 |
| 2.5 | Diagramme de séquence : mode de fonctionnement assisté | 17 |
| 2.6 | Diagramme de séquence : mode de fonctionnement avec détection de mouvement | 18 |
| 2.7 | Diagramme de séquence : consultation des images enregistrées | 18 |
| 2.8 | Diagramme de séquence : changement des paramètres de connexion des utilisateurs | 19 |
| 3.1 | Diagramme de composants de l'application de surveillance | 21 |
| 3.2 | Modèle MVR : architecture de base | 22 |
| 3.3 | Diagramme de composants Page Web | 23 |
| 3.4 | Diagramme de déploiement système de surveillance | 24 |
| 4.1 | Schéma du circuit imprimé | 26 |
| 4.2 | Schéma du dispositif mécanique | 27 |
| 4.3 | Raspberry Pi | 28 |
| 4.4 | Webcam | 29 |
| 4.5 | Servo-moteur | 29 |

TABLE DES FIGURES

| | | |
|------|--|----|
| 4.6 | Circuit d'alimentation | 32 |
| 4.7 | Vue externe du système | 33 |
| 4.8 | Vue interne du système | 33 |
| 5.1 | Diagramme de classe | 36 |
| 5.2 | Diagramme de séquence : Authentification | 37 |
| 5.3 | Diagramme de séquence : mode de fonctionnement normal | 37 |
| 5.4 | Diagramme de séquence : mode de fonctionnement assisté | 38 |
| 5.5 | Diagramme de séquence : mode d'utilisation avec détection du mouvement | 39 |
| 5.6 | Diagramme d'état/transition | 40 |
| 5.7 | Schéma de la base de donnée | 40 |
| 5.8 | Interface de connexion | 44 |
| 5.9 | Interface d'accueil | 45 |
| 5.10 | Mode de fonctionnement normal | 46 |
| 5.11 | Mode de fonctionnement assisté | 47 |
| 5.12 | Mode de fonctionnement avec détection du mouvement | 48 |
| 5.13 | Mode de fonctionnement avec détection du mouvement | 49 |
| 5.14 | Interface de l'administrateur | 49 |
| 5.15 | Interface d'aide | 50 |
| A.1 | Un servomoteur et ses accessoires | 55 |
| A.2 | Signal PWM | 56 |
| A.3 | Signal modulé en code d'impulsion | 57 |
| B.1 | Schéma du régulateur utilisé | 59 |
| B.2 | Schéma du circuit d'alimentation | 59 |

Liste des tableaux

| | | |
|-----|---|----|
| 4.1 | Tableau Comparatif | 30 |
| 5.1 | Comparatif entre les logiciel d'extraction des images | 43 |

Introduction Générale

LA vidéosurveillance est un système de caméras et de transmission d'images, disposé dans un espace public ou privé pour le surveiller. Les images obtenues avec ce système, peuvent être traités automatiquement ou visionnées puis archivées. La surveillance a pour but de contrôler les conditions de respect de la sécurité, de la sûreté ou de l'exécution d'une procédure particulière.

Les causes de l'installation de systèmes de vidéo-surveillance sont diverses, toutefois la sécurité publique ainsi que la protection des biens mobiliers ou immobiliers font office d'éléments phares dans la justification de la vidéo-surveillance.

La mise en place de la vidéo-surveillance permet une amélioration de la gestion des incidents ainsi qu'une augmentation de l'efficacité et de la rapidité d'intervention. Par exemple, dans la prévention du suicide ou encore lors d'accidents qui pourraient survenir sur la voie publique. Elle permet ainsi indirectement, de maintenir les primes d'assurances à un niveau raisonnable. La surveillance des axes routiers sert à informer en temps réel les automobilistes sur les conditions du trafic.

Outre les buts de sécurisation des biens et des personnes pour les entreprises et les commerces, les nouvelles technologies de vidéo-surveillance permettent désormais de réaliser le comptage des clients dans une boutique, ou des véhicules entrant et sortant d'une entreprise, d'incruster dans l'image vidéo le montant d'un ticket de caisse ou les informations d'un badge ou d'une carte d'accès, de reconnaître automatiquement les plaques d'immatriculations des véhicules (permet l'accès d'un parking sans ticket, clé ou badge), de signaler automatiquement un objet égaré, abandonné ou volé.

La vidéo-surveillance permet aussi de lever les angles morts, citons l'exemple de certains camions de ramassage d'ordures qui permettent au chauffeur de voir les mouvements des employés à l'arrière de son véhicule, ou des techniques utilisées dans les transports public.

Introduction Générale

Notre application consiste à implémenter une caméra de surveillance mobile dans une pièce qui nous permettra de couvrir les quatre angles et de détecter les mouvements dans cette pièce.

Le présent rapport présentant notre travail réalisé, s'articule autour de cinq chapitres divisés comme suit : Nous commençons par présenter l'état de l'art. Puis nous passons à l'analyse des besoins, où nous spécifions les besoins de l'utilisateur. Ensuite le troisième chapitre sera consacré à la conception globale. Et en dernier lieu nous consacrerons deux chapitres à l'étude conceptuelle et réalisation où nous détaillerons l'environnement matériel et logiciel utilisés ainsi que les interfaces de l'application.. Enfin nous terminons par les tests et validation de notre prototype.

Chapitre 1

État de l'art

Introduction

La phase d'étude théorique s'avère être importante pour la réalisation de tout projet, tout au long de cette phase essentielle une certaine recherche s'impose dans le but de définir les concepts et les terminologies à utiliser et de mettre en évidence l'importance des caméras de surveillance.

Dans ce chapitre, nous parlerons de la vidéosurveillance en générale. Puis, nous étudierons et critiquerons l'existant et nous finirons par présenter le travail demandé.

1.1 La vidéo

La vidéo regroupe l'ensemble des techniques permettant l'enregistrement ainsi que la restitution d'images animées, accompagnées ou non de son, sur un support électronique. En théorie un flux vidéo est composé d'une succession d'images qui défilent à un rythme fixe pour donner l'illusion du mouvement. Chaque image est décomposée en lignes horizontales, chaque ligne étant une succession de points. La lecture et la restitution d'une image s'effectue donc séquentiellement ligne par ligne comme un texte écrit de gauche à droite puis de haut en bas.

On distingue généralement plusieurs grandes familles d'images animées :

- Le cinéma, consistant à stocker sur une pellicule la succession d'images en négatif. La restitution du film se fait alors grâce à une source lumineuse projetant, à partir d'une

copie en positif, les images successives sur un écran.

- La vidéo analogique, représentant l'information comme un flux continu de données analogiques, destiné à être affichées sur un écran de télévision basé sur le principe du balayage. Il existe plusieurs normes pour la vidéo analogique. Les trois principales sont :

PAL

NTSC

SECAM

- La vidéo numérique, consistant à coder la vidéo en une succession d'images numériques.

1.1.1 PAL/SECAM

Le format PAL/SECAM (Phase Alternating Line/Séquentiel Couleur avec Mémoire), utilisé en Europe pour la télévision hertzienne, permet de coder les vidéos sur 625 lignes (576 seulement sont affichées car 8% des lignes servent à la synchronisation) à raison de 25 images par seconde à un format 4 :3 (c'est-à-dire que le rapport largeur sur hauteur vaut 4/3).

Or à 25 images par seconde, de nombreuses personnes perçoivent un battement dans l'image. Ainsi, étant donné qu'il n'était pas possible d'envoyer plus d'informations en raison de la limitation de bande passante, il a été décidé d'entrelacer les images, c'est-à-dire d'envoyer en premier lieu les lignes paires, puis les lignes impaires. Le terme "champ" désigne ainsi la "demi-image" formée soit par les lignes paires, soit par les lignes impaires. L'ensemble constitué par deux champs est appelé trame entrelacé. Lorsqu'il n'y a pas d'entrelacement le terme de trame progressive est utilisé.

Grâce à ce procédé appelé "entrelacement", le téléviseur PAL/SECAM affiche 50 champs par seconde (à une fréquence de 50 Hz), soit 2x25 images en deux secondes.

1.1.2 NTSC

La norme NTSC (National Television Standards Committee), utilisée aux Etats-Unis et au Japon, utilise un système de 525 lignes entrelacées à 30 images/sec (donc à une fréquence de 60Hz). Comme dans le cas du PAL/SECAM, 8% des lignes servent à synchroniser le récepteur. Ainsi, étant donné que le NTSC affiche un format d'image 4 :3, la résolution réellement affichée est de 640x480.

1.1.3 La vidéo numérique

La vidéo numérique consiste à afficher une succession d'images numériques. Puisqu'il s'agit d'images numériques affichées à une certaine cadence, il est possible de connaître le débit nécessaire pour l'affichage d'une vidéo, c'est-à-dire le nombre d'octets affichés (ou transférés) par unité de temps.

Ainsi le débit nécessaire pour afficher une vidéo (en octets par seconde) est égal à la taille d'une image que multiplie le nombre d'images par seconde.

Soit une image true color (24 bits) ayant une définition de 640 pixels par 480. Pour afficher correctement une vidéo possédant cette définition il est nécessaire d'afficher au moins 30 images par seconde.

1.2 La vidéosurveillance

La vidéosurveillance est un système de caméras et de transmission d'images, disposé dans un espace public ou privé pour le surveiller. Les images obtenues avec ce système, peuvent être traités automatiquement et/ou visionnées puis archivées ou détruites. La surveillance a pour but de contrôler les conditions de respect de la sécurité, de la sûreté ou de l'exécution d'une procédure particulière.

L'objectif général d'un système de télésurveillance est de contribuer à la sécurité de biens et/ou de personnes. En effet elle permet plusieurs fonctionnalités.

1.2.1 La prévention de la criminalité

Dans cette approche, les systèmes de vidéosurveillance sont censés, selon leurs promoteurs, permettre de surveiller ou décompter les clients d'une boutique, ou des véhicules entrant et sortant d'une entreprise, voir, potentiellement, d'incruster dans l'image vidéo le montant d'un ticket de caisse ou les informations d'un badge ou d'une carte d'accès, de reconnaître automatiquement les plaques d'immatriculations des véhicules (permet l'accès d'un parking sans ticket, clé ou badge), de signaler automatiquement un objet égaré, abandonné ou volé).



FIGURE 1.1 – Exemples de captures de vidéos dans le cas de prevention de criminalité

1.2.2 La sécurité routière

Dans le domaine de la sécurité routière, la vidéosurveillance permet l'accès en temps réel de la circulation routière des voies les plus fréquentées (autoroutes, voies rapides) ainsi que pour la détection d'accident, elle permet notamment de lever les angles morts. Ainsi, dans certains camions de ramassage d'ordures, le chauffeur peut voir les mouvements des employés l'arrière de son véhicule. Cette technique est également utilisée dans les transports public, dans le cadre des équipements à agent seul.



FIGURE 1.2 – Cas d'application en circulation routière

1.2.3 La sécurité industrielle

Dans le domaine de la sécurité industrielle, des sites de production sont équipés de systèmes de vidéosurveillance permettant notamment de multiplier les points d'observation en temps réel de l'état des installations et du déroulement du procédé. Divers lieux (sites industriels, installations stratégiques, objets du patrimoine...) sont vidéosurveillés pour prévenir les intrusions et les dégradations de la part de personnes malveillantes.



FIGURE 1.3 – Cas d'application en sécurité industrielle

Afin de réaliser les fonctionnalités citées ci-dessus l'utilisateur a le choix d'utilisation entre deux principales familles. Un serveur de caméra ou une caméra réseau.

Nous détaillerons dans la partie suivante leurs modes de fonctionnement tout en mettant l'accent sur leurs manques.

1.3 Étude de l'existant

1.3.1 Un serveur de caméra

Il contient toute la partie serveur d'une caméra réseau, on lui ajoute une caméra PAL pour la vidéosurveillance. Il doit aussi être reliées par une prise réseau RJ45 et au moins une prise coaxiale.



FIGURE 1.4 – Serveur Camera

Cette technique est la première à être utilisé du moment qu'elle est couteuse en terme d'installation et des couts de câblage, et est aussi limité en distance.

1.3.2 Une caméra réseau

C'est une caméra qui se connecte directement au réseau. Elle ne possède qu'une prise RJ45 pour connexion directe avec le hub ou le switch. Elle est équipé d'un serveur web interne.

Ce type de caméra présente deux avantages par rapport au serveur caméra (les couts de câblages et l'illimitation de la distance).

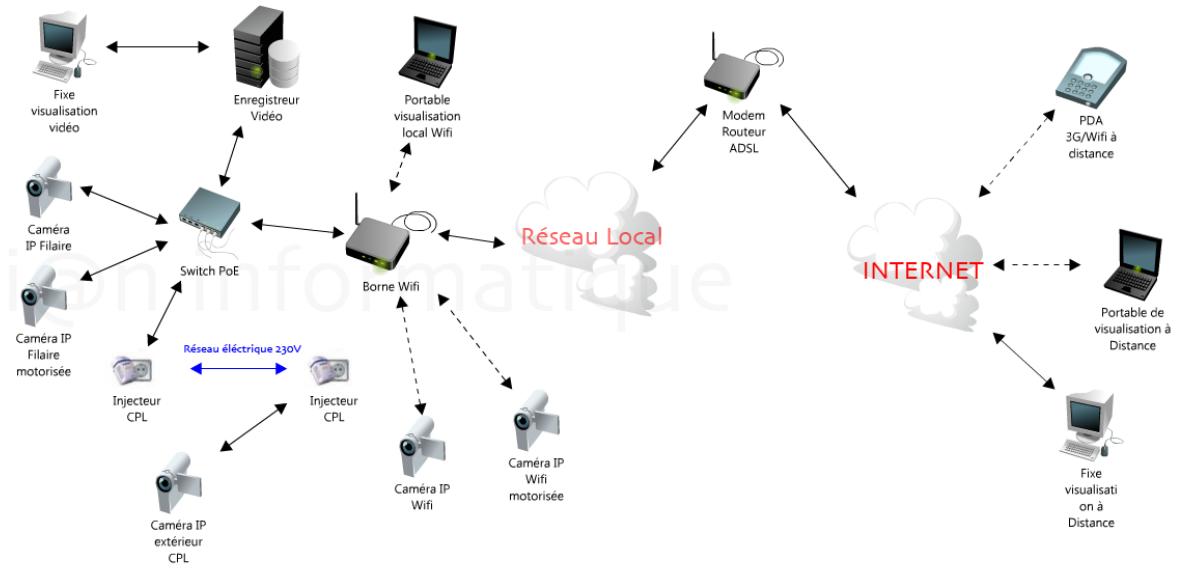


FIGURE 1.5 – Camera Réseau

Cependant quelques autres limitations n'ont pas encore été surpassées :

- Dépendance de l'utilisateur : L'utilisateur doit rester vigilant à regarder l'écran afin de détecter une intrusion ce qui est impossible. Par ailleurs l'expérience des Britanniques le pays le plus surveiller au monde a été qualifié par un «fiasco» de la part du responsable de la sécurité « les caméras n'ont pas garanti la sécurité des citoyens,du fait de l'incapacité de la police à gérer le flux d'images ... »
- Cout élevé des caméras (Caméra réseau) ou des câblages (serveurs caméras)
- Limitation des fonctionnalités : En effet l'utilisateur ne peut que bénéficier des fonctionnalités implémenter par le constructeur et ne peut ni les modifier ni utiliser la caméra sur n'importe quel système d'exploitation désiré.
- Mobilité : Les caméras utilisées sont fixes elles ne sont pas déplaçables suivant les désirs de l'utilisateur.

La plus part des caméras de surveillance actuelles nécessitent beaucoup de ressources matérielles pour leur bon fonctionnement et limitent de façon considérable les choix des utilisateurs.

Nous proposerons dans la partie qui suit des solutions permettant de surmonter ses limita-

tions tout en expliquant le travail demandé

1.4 Travail demandé

Nous élaborerons une caméra de surveillance composé d'une webcam HD, relié a un RaspberryPi a son tour fixé sur un support transportable et tournant sur deux axes, ce qui nous permettra :

- Indépendance de l'utilisateur : La caméra de surveillance détecte automatiquement une intrusion ce qui permettra a l'utilisateur de plus de liberté. Il n'a plus besoin à regarder tout le flux afin de détecter une intrusion.
- Cout faible : 270 Dinars pour tout le système
- Mobilité : Le système est facile a transporter et couvre toute une pièce.
- Fonctionnalité : Le flux peut être détecter de n'importe quel équipement relié à internet.

Conclusion

Ce chapitre nous a permis d'amener une étude préalable de notre projet et ceci en comprenant le principe de fonctionnement de la vidéo, en exploitant les types de caméras existants sur le marché, en évaluant leurs performances et en proposant des solutions plus adaptées afin de remédier aux insuffisances qu'ils présentaient. Les objectifs étant éclaircis, nous spécifierons dans le chapitre suivant clairement les fonctionnalités de notre système.

Chapitre 2

Analyse et Spécification des besoins

Introduction

Avant d'entamer la phase de conception d'un projet, il est primordiale de bien définir son environnement ainsi importante que l'aspect technique du projet, une définition des besoins qui en découlent ne pourra qu'être bénéfique grâce au gain précieux de temps qu'elle offre.

2.1 Aperçu globale sur le projet

Le travail réalisé s'articule sur deux axes principaux à savoir :

- **Partie développement temps réel sur PC** : conception et mise au point d'un serveur qui permet d'envoyer les images aux clients connectés.
- **Partie développement embarqué** : conception d'un système embarqué qui permet le mouvement de la caméra en rotation et qui va être commandé par l'utilisateur.



FIGURE 2.1 – Aperçu du travail à réaliser

2.2 Analyse des besoins

Cette phase est importante pour garantir la performance et la robustesse du projet. En effet, à partir des objectifs définis dans le cahier des charges on détermine l'ensemble des besoins à satisfaire. Il existe plusieurs services que notre système doit fournir. Cette partie comporte trois sous-parties : une première sous-partie au court de laquelle nous spécifierons les acteurs en interaction avec le système et dans les deux dernières sous-parties nous allons détailler les besoins éprouvés par ces acteurs.

2.2.1 Les acteurs du système

Dans une première étape, nous allons spécifier les acteurs auxquels le système offre ses services. Nous avons principalement deux types d'acteurs : un utilisateur normal soit unique ou plusieurs personnes et un administrateur unique pour tout le système. L'administrateur est un utilisateur à la base mais qui a le privilège de changer les paramètres du système de surveillance.

2.2.2 Les besoins fonctionnels

Ces exigences doivent répondre à la question : "A quoi sert notre application ? "

Notre projet a pour finalité de satisfaire les besoins suivants :

- **Authentification de l'utilisateur :**

Cela permet de vérifier l'identité présumée de l'utilisateur pour lui garantir l'accès au système via Internet. La procédure de l'authentification se fait grâce à une page web dynamique tout en remplaçant les champs requis ce qui permet d'une part la sécurité de l'accès et d'autre part la gestion du client. Une fois connecté l'utilisateur peut accéder aux différents services mentionnés dans le site.

- **Affichage du flux vidéo sur la page :**

La page web permet au propriétaire de consulter son camera depuis n'importe où et de visualiser les images. L'utilisateur peut aussi enregistrer les images sur le disque dur, appliquer un traitement simple sur le flux et extraire la vidéo.

- **Contrôler la direction de la caméra :**

En plus des besoins décrits précédemment, la page web doit permettre aussi de contrôler la direction de la caméra via des boutons. Ce qui permettra de couvrir tout le champ de vision accessible par un mouvement de rotation qui se fait selon deux axes perpendiculaires.

- **Affichage du backup d'images :**

L'utilisateur aura la possibilité de consulter les images enregistrées ainsi que les stocker sur un autre support physique

- **Détecter le mouvement par la caméra :**

Le système doit permettre la détection du mouvement par analyse d'images qui se suivent.

- **Alerter l'utilisateur :**

Au cours du traitement d'images et dans le cas où un mouvement d'un objet à l'endroit où est placé la caméra fut détecté, l'utilisateur doit être averti par la page web.

2.2.3 Les besoins non fonctionnels

Ces exigences ne concernent pas le comportement ou le fonctionnement du système. Parmi ces besoins nous citons :

- **Utilisabilité :**

L'utilisation du système ne doit pas nécessiter des connaissances pré-requises que ce soit

en informatique ou en mécanique. En d'autres termes, le système doit être facile à comprendre, illustré et adapté au public.

– **Performance :**

Le système doit supporter le déroulement de plusieurs tâches à la fois et doit être capable de les gérer sans produire de bugs. De plus, le délai de chargement de la page web ne devrait pas prendre beaucoup de temps pour s'afficher. Concernant la transmission du flux vidéo, sa fréquence doit être inférieure à 3 Hertz soit d'une moyenne de 3 images par secondes .

– **Efficacité :**

Le système doit parvenir à produire les résultats escomptés et réaliser des objectifs fixés d'une façon optimale lors du traitement des informations externes.

– **Fiabilité :**

Le système doit détecter le mouvement des objets avec une précision bien définie.

– **Ergonomie :**

Le système doit offrir une page web avec une interface graphique conviviale bien structurée du point de vue contenu informationnel et facile à utiliser.

– **Maintenabilité :**

Le système doit être facile à maintenir et réparer par le moyen des mises à jour et d'une façon cohérente.

2.3 Spécification des besoins

Dans cette partie, nous abordons la phase spécification des besoins. Ainsi, nous utiliserons le langage UML comme moyen simple et compréhensible afin de décrire les principaux points du projet qui seront différents besoins déjà éprouvés au cours de l'analyse à travers deux diagrammes. Un premier diagramme des cas d'utilisation qui décrit les différents scénarios d'utilisation.

2.3.1 Diagramme de cas d'utilisation

Nous présentons le diagramme de cas d'utilisation dans la figure 2.2. Ce diagramme montre les fonctionnalités offertes par le système à chaque type d'acteur.

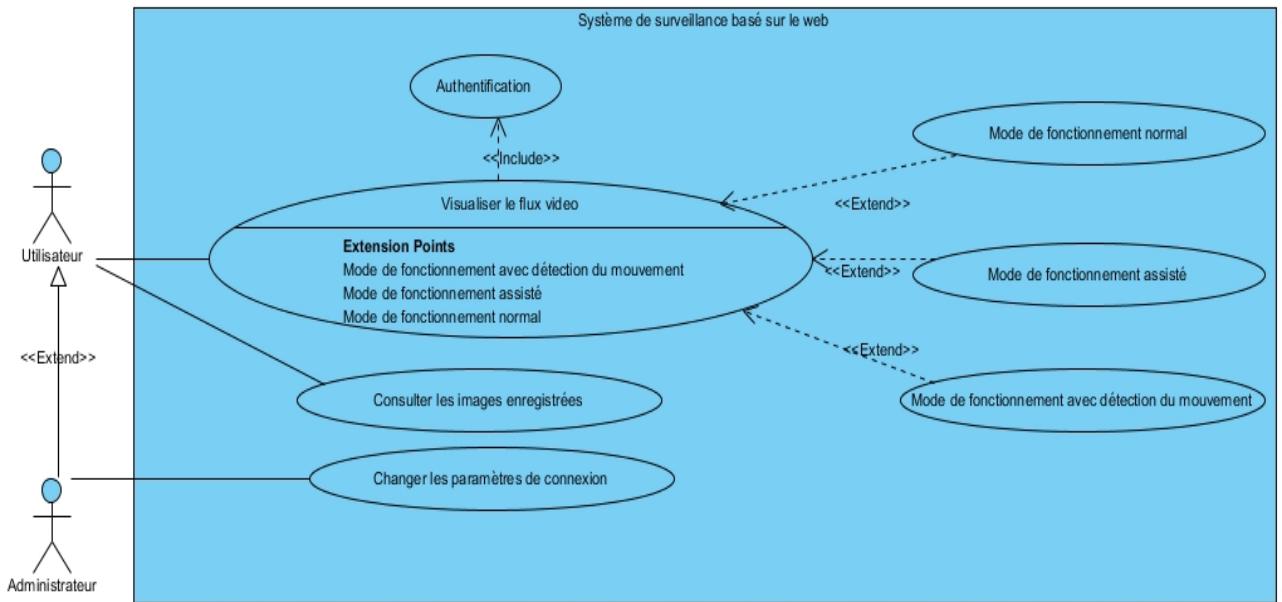


FIGURE 2.2 – Diagramme de cas d'utilisation

2.3.2 Les scénarios d'utilisation

Dans cette partie nous détaillerons les différentes interactions entre l'utilisateur et le système de surveillance selon plusieurs modes d'utilisation. Cette interaction sera mise en valeur par le moyen des diagrammes de séquences.

2.3.2.1 Scénario d'utilisation 1 : Authentification

Toute utilisation pratique de notre système requière une première étape d'authentification. Un utilisateur non authentifié ne peut pas utiliser le système de surveillance, pour des raisons de sécurité. Le diagramme de la figure 2.3 expose le scénario d'authentification d'un utilisateur. L'administrateur, étant un utilisateur privilégié, il suit les mêmes étapes d'authentification.

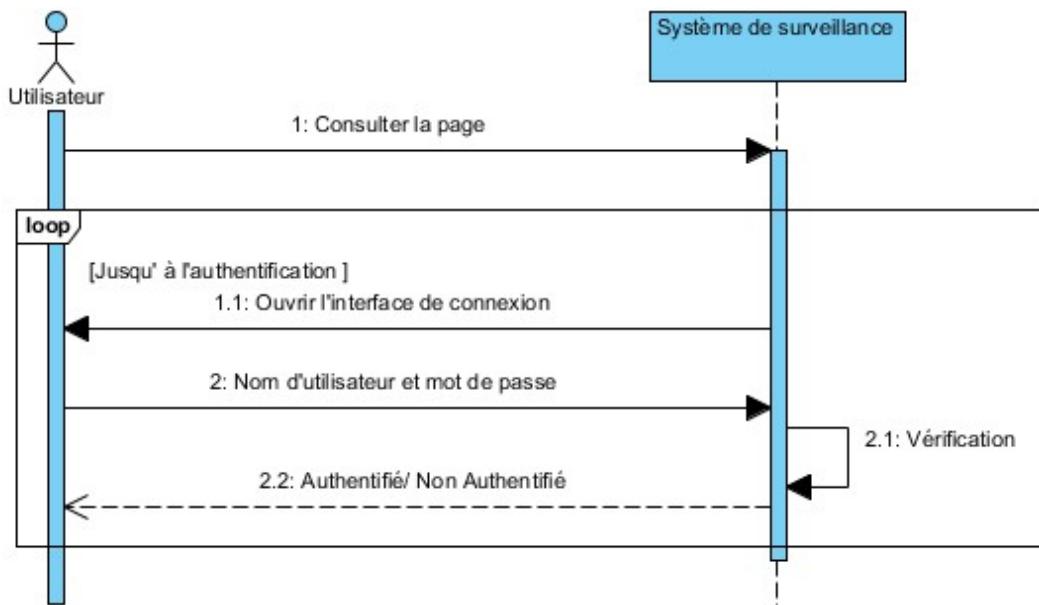


FIGURE 2.3 – Diagramme de séquence : Authentification

2.3.2.2 Scénario d'utilisation 2 : mode de fonctionnement normal

La figure 2.4 illustre les différentes interactions entre l'utilisateur et le système de surveillance en mode d'utilisation normal.

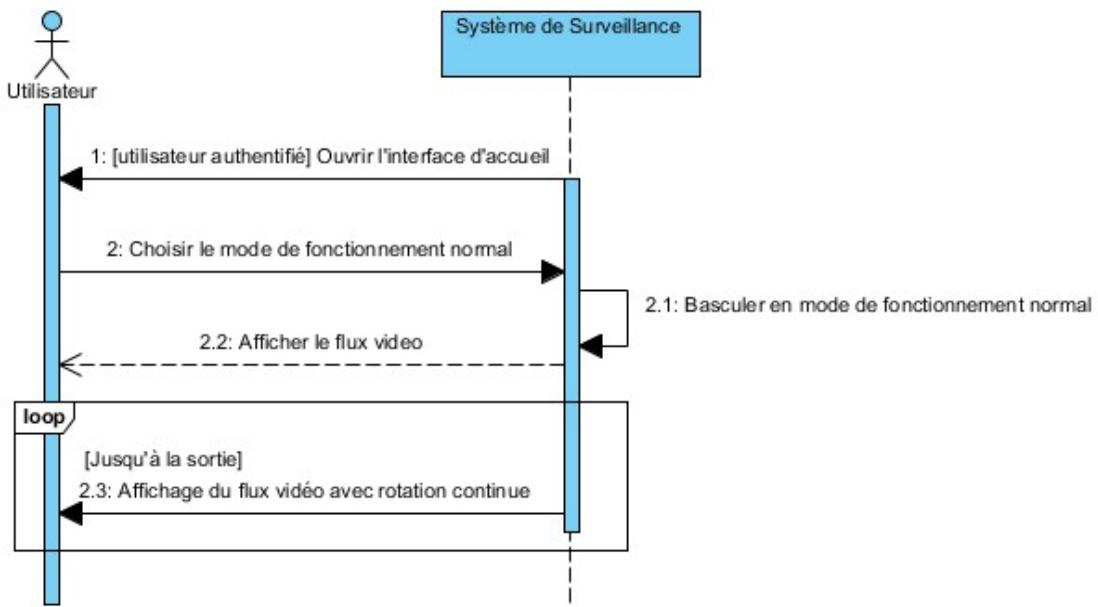


FIGURE 2.4 – Diagramme de séquence : mode de fonctionnement normal

2.3.2.3 Scénario d'utilisation 3 : mode de fonctionnement assisté

Un second scénario possible qui montre l'interaction entre l'utilisateur et le système en mode assisté sera illustré par la figure 2.5

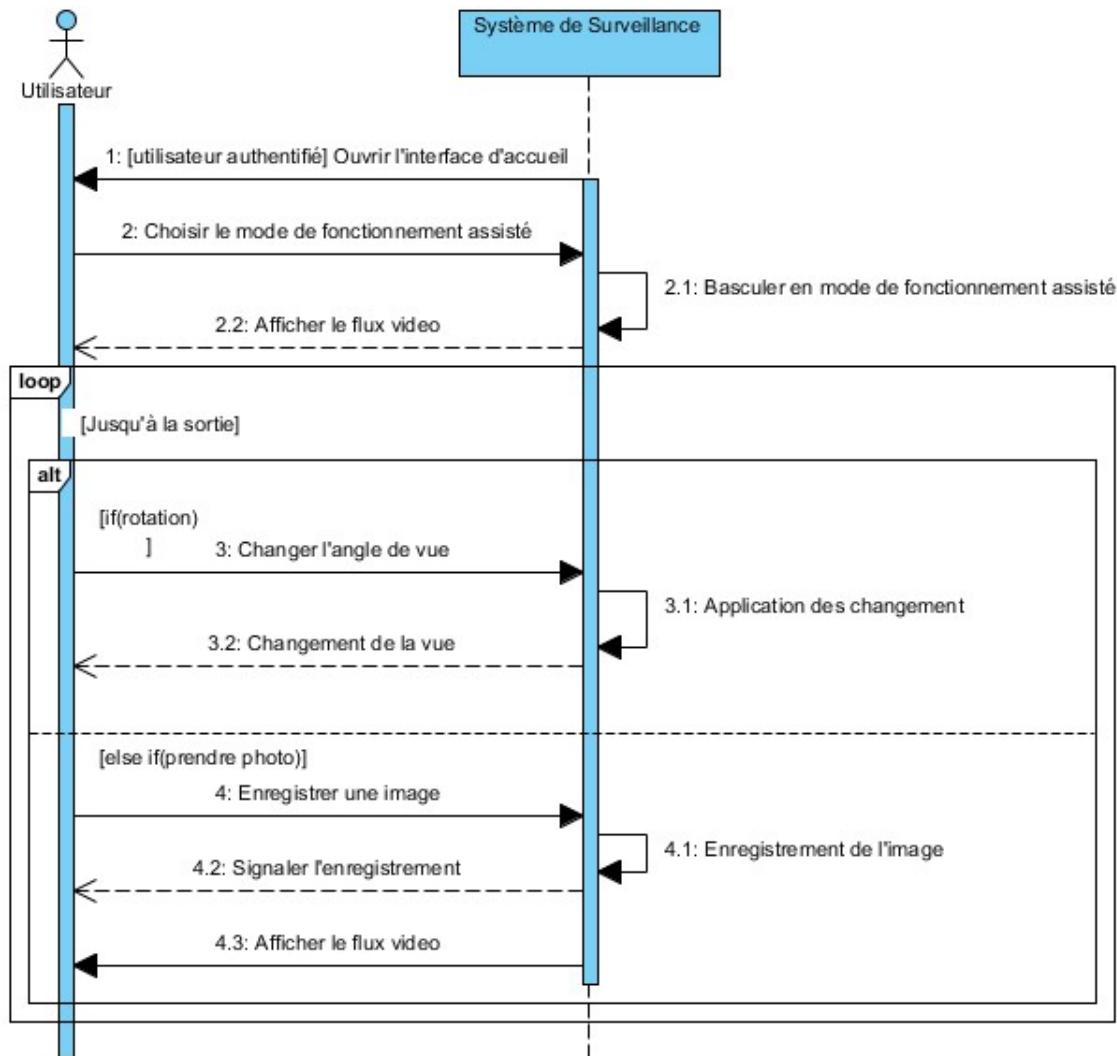


FIGURE 2.5 – Diagramme de séquence : mode de fonctionnement assisté

2.3.2.4 Scénario d'utilisation 4 : mode de fonctionnement avec détection de mouvement

Un troisième et dernier mode de fonctionnement possible est le mode de détection de mouvement. Le diagramme de séquence de la figure 2.6 illustre ce scénario d'utilisation.

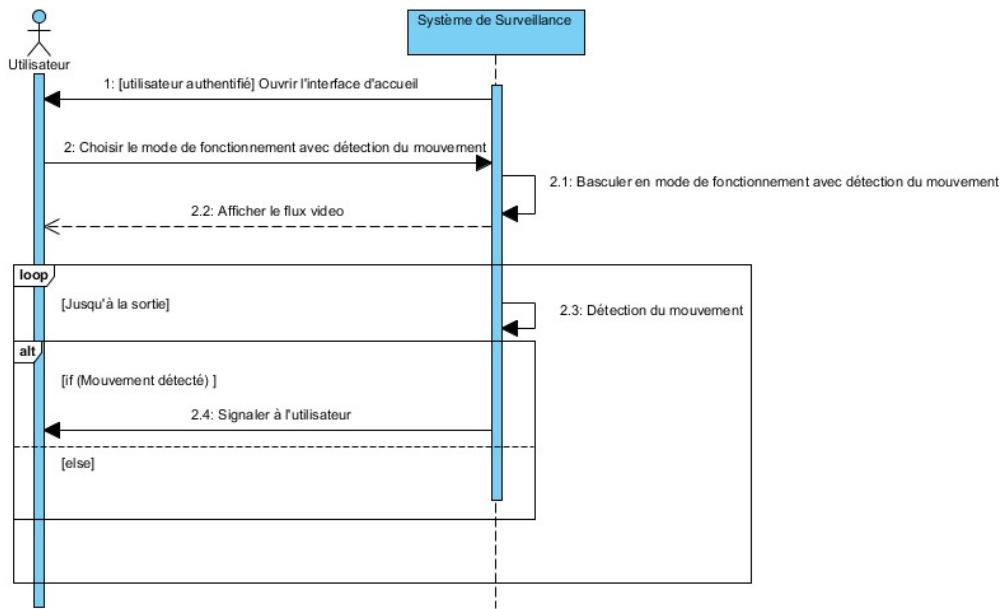


FIGURE 2.6 – Diagramme de séquence : mode de fonctionnement avec détection de mouvement

2.3.2.5 Scénario d'utilisation 5 : consultation des images enregistrées

Ce scénario illustre l'utilisation du système par l'utilisateur via la page web afin de visualiser les images enregistrées.

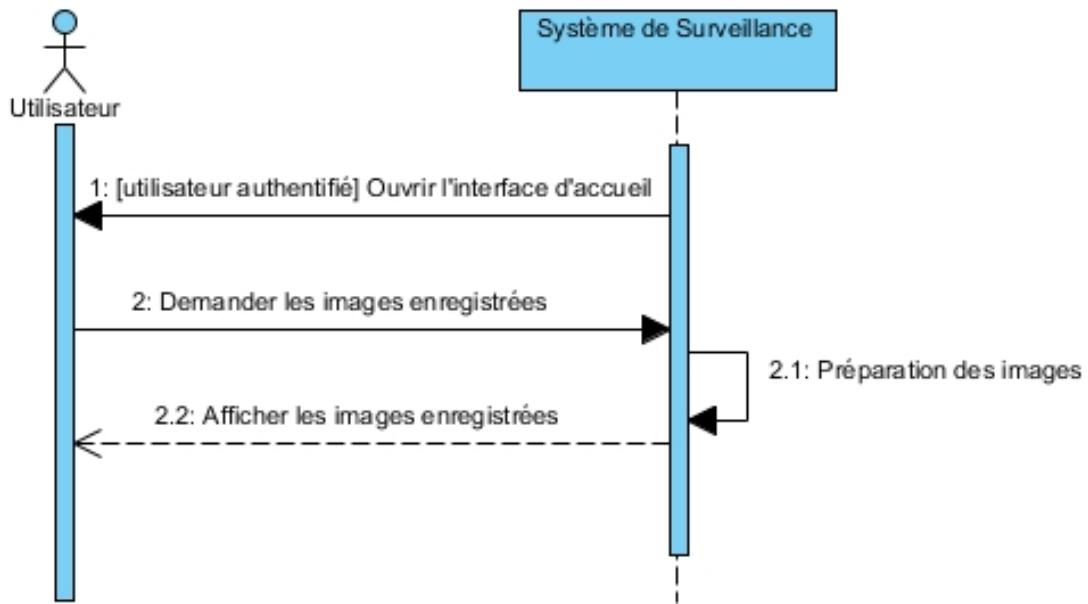


FIGURE 2.7 – Diagramme de séquence : consultation des images enregistrées

2.3.2.6 Scénario d'utilisation 6 : changement des paramètres de connexion des utilisateurs

Ce scénario d'utilisation met l'accent sur le privilège donné à l'administrateur du système de surveillance. En effet ce dernier peut gérer tous les paramètres de connexion des utilisateurs de notre système de surveillance.

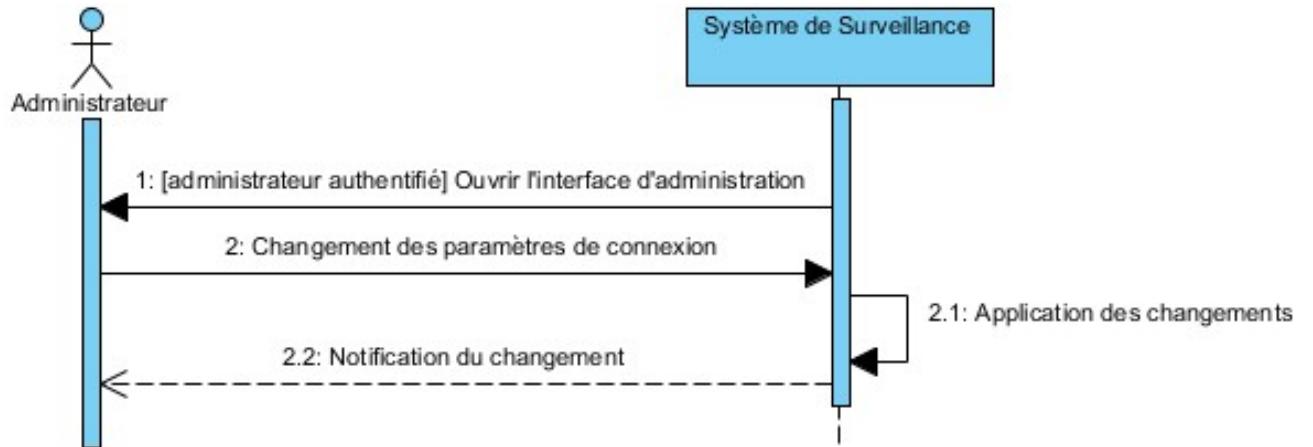


FIGURE 2.8 – Diagramme de séquence : changement des paramètres de connexion des utilisateurs

Conclusion

Ce chapitre a pour but de détailler les différentes fonctionnalités que le système de surveillance offre en partant des besoins éprouvés par les utilisateurs. Ce qui nous permettra d'aborder la partie suivante au court de laquelle nous proposerons une conception architectural de notre solution.

Chapitre 3

Conception globale

Introduction

Le présent chapitre sera consacré à la présentation de la conception globale de notre projet. Après avoir présenté les différents besoins fonctionnels et non fonctionnels de notre système et discuté les différents cas d'utilisation et les scénarios possibles, nous entamons à présent la phase de conception.

En fait, cette étape représente la squelette du cycle de vie d'un logiciel puisqu'elle permet de décider comment satisfaire les différents besoins dégagés dans la phase de spécification. Dans cette section , nous allons détaillé les modèles utilisé pour la conception architecturale de notre projet. En gros , cette partie comportera deux sous-parties : une première qui traite de la conception globale de l'application de surveillance et une deuxième qui s'intéresse a la conception globale du dispositif de surveillance.

3.1 Conception globale de l'application de surveillance

Le diagramme de composants décrit l'organisation du système du point de vue purement logiciels. On s'intéresse ainsi aux différents modules que contient notre application , des données et aux éléments de configuration . Ce diagramme permet de mettre en évidence les dépendances entre les composants logiciel de notre application à travers des interfaces fournies et requises.

3.1.1 Diagramme de composants de l'application

La figure 3.1 est la représentation de l'architecture logicielle de notre application de surveillance. Comme l'illustre cette figure notre application est formée principalement de quatre composantes logicielle à savoir une application qui assure l'interaction entre les différents composants du système d'autre part et le client de l'autre part, un composant qui extrait le flux vidéo à partir de la caméra, un logiciel qui permet de commander les servomoteur et un dernier composant qu'est une base de donnée qui permet de vérifier l'identité de l'utilisateur. Il est aussi à noter que l'application de surveillance contient entre autre son propre serveur web donc nous n'avons pas besoin d'utiliser une autre composante logicielle qui assure cette fonction.

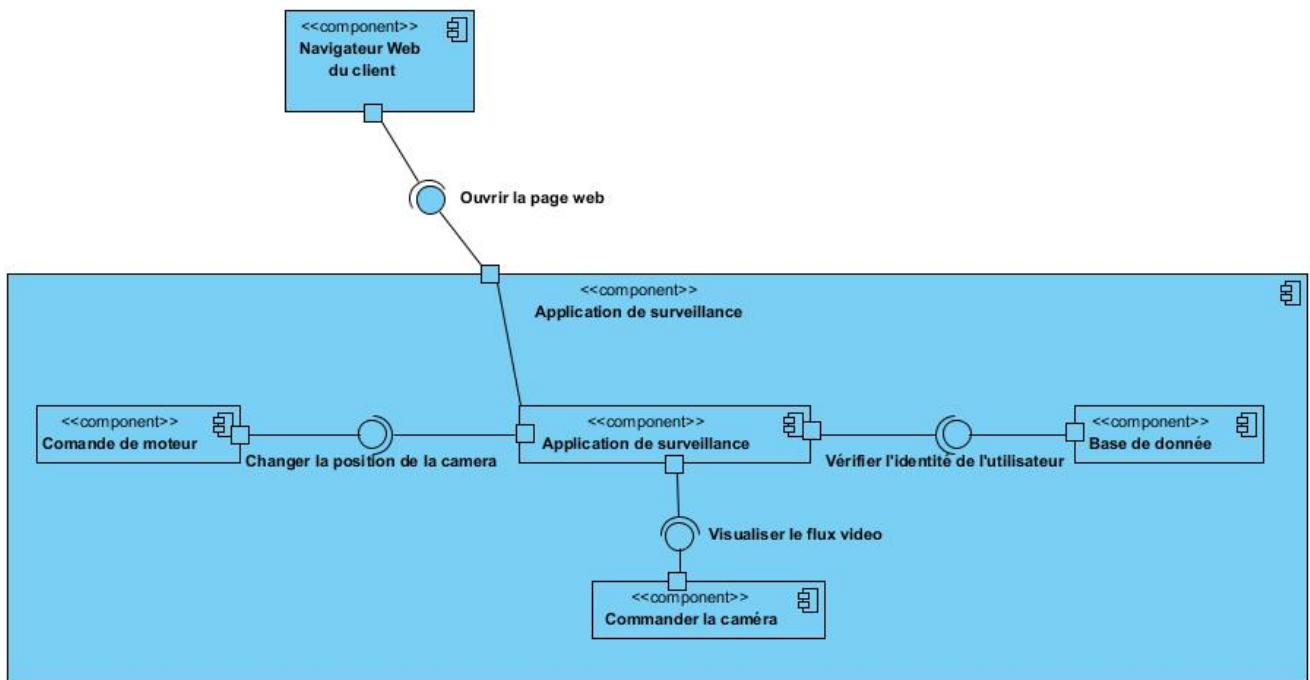


FIGURE 3.1 – Diagramme de composants de l'application de surveillance

3.1.2 Diagramme de composant de la page

Dans cette partie nous allons nous intéresser par l'architecture de notre application principale qui assure l'interaction entre le client et les différents composants de notre application. En effet cette application est une page web conçu suivant le modèle “Modèle Vue Routeur” (MVR). Ce modèle comme son nom l'indique est basé principalement sur trois composants logiciels :

- Le modèle représente le cœur battant de notre application .En effet, cette partie comporte tous les traitements que peut assurer notre application . En particulier cette partie com-

portera les algorithmes de traitement des images , de détection des mouvements , de commande des servomoteurs ...

- La vue est une représentation simpliste des données et des résultats de traitement effectués dans le modèle, en effet cette partie est l'une des plus importantes dans notre application car elle représente l'interface en interaction directe avec l'utilisateur du système de surveillance. C'est grâce à ce composant logiciel que l'utilisateur peut faire bon usage des algorithmes codés dans la partie modèle.
- Le routeur s'occupe du routage entre les différentes pages web de notre application. Ce composant paraît d'une grande importance dans la gestion des droits d'accès aux différentes pages de notre système de surveillance. Il permet ainsi de générer les différentes vues de notre système et d'éviter les accès inappropriés.

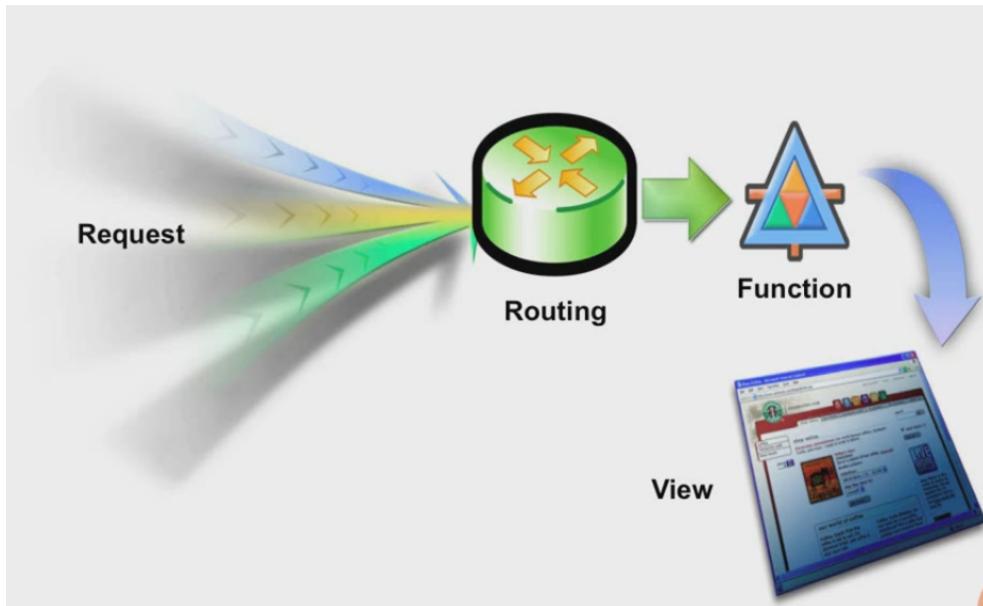


FIGURE 3.2 – Modèle MVR : architecture de base

La figure 3.3 permet de détailler l'architecture logicielle de notre page web qui n'est autre que l'aspect terminal de notre application de surveillance .

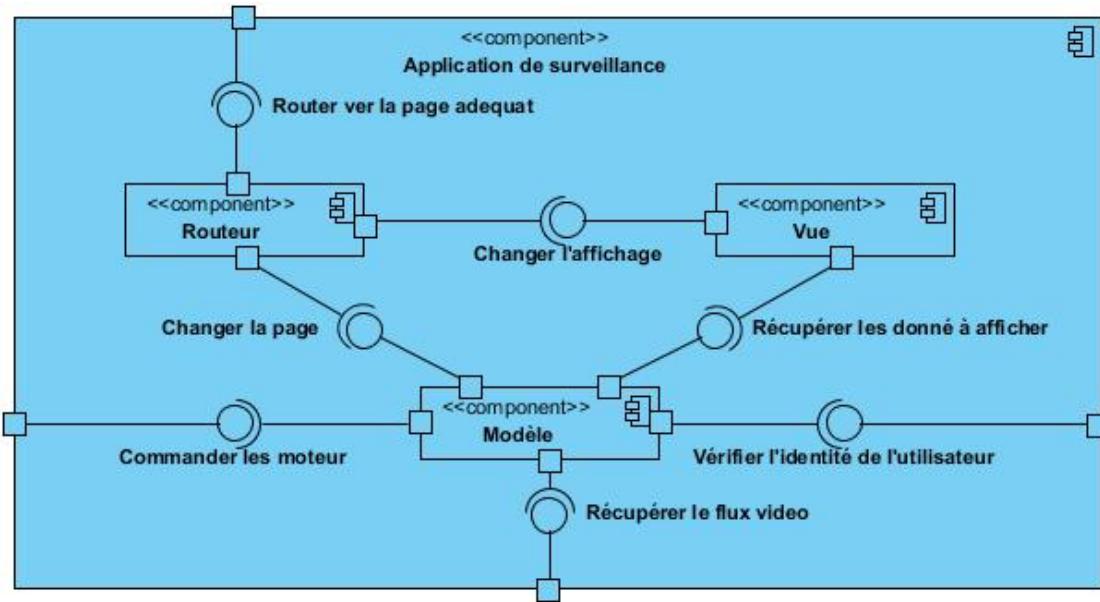


FIGURE 3.3 – Diagramme de composants Page Web

3.2 Diagramme de déploiement : Architecture matérielle du système

Vue la nature embarqué de notre projet , il paraît crucial de détailler l'architecture de notre système de surveillance . Dans le langage UML l'un des diagrammes qui permettent ceci est le diagramme de déploiement. En effet, ce dernier est une représentation statique de l'utilisation de l'infrastructure physique du système, de la manière dont ces composants sont répartis et des relations entre ces composants. Ce diagramme met en valeur les composants physiques du système et leurs caractéristiques en termes de ressources matérielles physiques et de support de communication.

La figure 3.4 illustre le diagramme de déploiement de notre système. Ce dernier comporte une unité de traitement de donnée, une caméra, deux moteurs et un circuit d'alimentation.

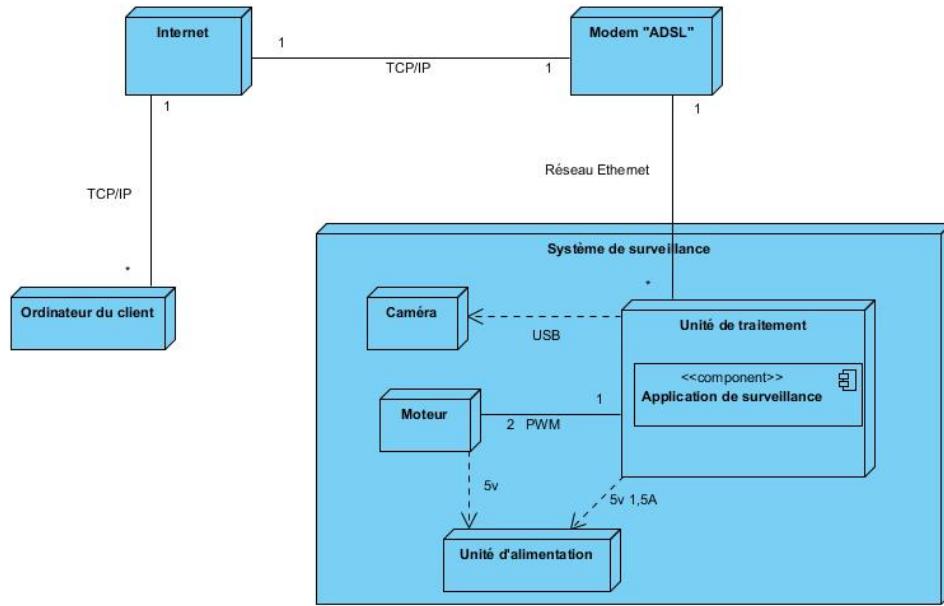


FIGURE 3.4 – Diagramme de déploiement système de surveillance

Conclusion

Nous clôturons par le biais de cette partie la première phase de conception de notre application. Tout au long de ce chapitre nous avons détaillé la conception globale. Nous entamerons la conception détaillé dans les deux chapitres qui suivent afin de développer des idées claires. Cela aura une grande importance dans la concrétisation de ces idées dans un projet complet et fonctionnel.

Chapitre 4

Conception et réalisation de la partie opérative

Introduction

Dans une première étape de la réalisation du projet, nous commencerons par la mise en place de la partie opérative. Cette partie comporte les actionneurs qui sont principalement les deux servomoteurs et la caméra. Nous allons aussi mettre en place un circuit d'alimentation qui va nous permettre d'alimenter les différentes parties de notre dispositif de surveillance. Ce chapitre est composé principalement de trois sections : une première qui traite la partie conception, une seconde qui justifie les différents choix technologiques du matériel et une dernière section qui expose les étapes de réalisation du dispositif.

4.1 Conception détaillé du dispositif de surveillance

Cette partie se déroule en deux étapes :

- La conception du circuit d'alimentation du dispositif.
- La conception de la solution mécanique qui permet le mouvement de la caméra.

4.1.1 Conception du circuit d'alimentation

L'enjeu de l'alimentation du dispositif est d'une importance majeure pour notre projet. En effet une mauvaise alimentation en courant ou en tension peut induire un dysfonctionnement du système et au pire des cas griller l'une des composantes du système. Par suite il faut prendre

soin de concevoir une alimentation qui soit suffisamment fiable pour assurer la sécurité des utilisateurs du système.

Dans ce cadre nous avons choisi de préparer notre propre alimentation et de ne pas se contenter d'acheter une toute prête. Pour ce faire il faut recourir à un logiciel de conception assisté par ordinateur (CAO) des circuits imprimés. Nous avons donc utilisé le logiciel répondu et assez simpliste qui s'appelle "ARES PCB Design" élément de la suite logiciel "Proteus". Le résultat est illustré par le schéma de la figure 4.1

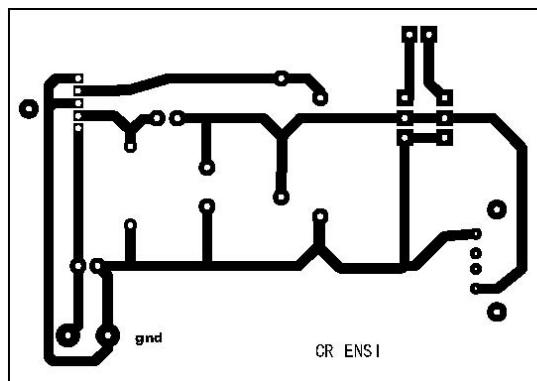


FIGURE 4.1 – Schéma du circuit imprimé

4.1.2 Conception du dispositif mécanique

La disposition mécanique du système de surveillance doit permettre à la caméra de surveillance de surveiller l'endroit où elle se trouve dans tous les degrés de liberté possible. Cette contrainte du système qui est mécanique par excellence, doit être traitée avec beaucoup d'imagination et de créativité. Une des solutions qui nous permet d'aboutir rapidement à une solution optimale est d'utiliser un logiciel de conception assisté par ordinateur de pièce mécanique tel que SolidWorks. Un tel logiciel permet d'étudier la possibilité de réalisation d'une éventuelle solution sans avoir à la fabriquer réellement. On peut donc réduire le coût de réalisation des prototypes de test. Et avec les révolutions technologiques on peut même réaliser les composantes mécaniques à l'aide d'une imprimante 3D. Mais cette solution ne se pose pas dans notre cas vue nos ressources financières très limitées.

Le dispositif mécanique illustré par la figure 4.2 est celui que nous allons adopté pour notre projet. Dans ce dispositif le premier moteur est fixé sur un support stable et sur ce même moteur on fixe un deuxième support mouvant pour le second moteur. Le second moteur ainsi fixé il suffit d'utiliser un troisième support qui sert à la fixation de la caméra sur le deuxième mo-

teur. De point de vue degrés de liberté, les deux moteurs doivent couvrir 180 ° de mouvement pour couvrir toute la demi-sphère de telle sorte à surveiller tout l'endroit.

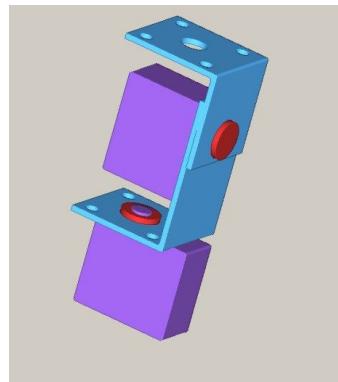


FIGURE 4.2 – Schéma du dispositif mécanique

4.2 Environnement matériel

Pour un projet embarqué le choix de l'environnement matériel peut être un élément décisif de la réussite ou de l'échec du projet. Nous avons donc pris soin de choisir, en fonction de ce qui était possible et de nos moyens, la meilleure solution. Donc l'un des premiers choix à faire était de choisir la carte de développement de notre projet. Et dans le domaine très vaste de l'embarqué notre choix est passé de la carte STM32 à la carte Arduino sans trouver l'idéal pour notre projet pour arriver finalement à la carte de développement Linux Embarqué Raspberry Pi. En ce qui concerne le choix des deux autres composantes matérielles à savoir la webcam et les servomoteurs le choix est plutôt technologique et non pas d'une performance bien déterminée.

4.2.1 Choix de l'environnement matériel

Dans cette partie nous allons exposer les différents composants matériels utilisés pour la réalisation du projet.

4.2.1.1 Raspberry Pi

Le Raspberry Pi est un ordinateur monocarte à base de processeur ARM qui possède les caractéristiques suivantes :

- une Soc Broadcom dotée d'un processeur ARMv6 700MHz, d'une GPU et d'un FPU

- 256 Mo de RAM
- 2 sorties vidéo dont une Composite et une HDMI
- 1 sortie audio stéréo Jack
- unité de lecture-écriture de carte mémoire
- 2 ports USB 2.0
- 1 port réseau Fast Ethernet (10/100 Mbits/s)
- prise pour alimentation micro-USB
- 26 pins de types GPIO, I2C, SPI...



FIGURE 4.3 – Raspberry Pi

4.2.1.2 Une webcam Logitech HD C310

Les caractéristiques de la webcam sont :

- connectivité filaire USB 2.0 haut débit
- type couleur
- résolution 1280 x 720
- vidéo HD 720p
- photos 5 mégapixels
- microphone intégré
- compatible avec V4l de Linux.



FIGURE 4.4 – Webcam

4.2.1.3 Deux servomoteurs Hitec HS-311

Les caractéristiques des servomoteurs sont :

- système de contrôle : signal PWM 1,5 milli-second
- tension de fonctionnement : 4,8 à 6,0 Volts
- vitesse de fonctionnement (4,8 V) : 0.19sec / 60 ° à vide
- couple de blocage (4.8V) : 3,0 kg / cm
- angle de fonctionnement : 45 °
- direction : multidirectionnel



FIGURE 4.5 – Servo-moteur

4.2.2 Justification du choix de l'environnement matériel

Dans ce qui suit nous allons justifier chacun des choix matériel effectué.

4.2.2.1 Raspberry Pi

Au moment où nous avons commencé les recherches à propos de ce sujet, nous étions contraint de trouver une carte qui soit capable de faire du traitement d'image et qui soit dans la mesure de notre pouvoir financier. Aussitôt le choix de la carte Raspberry Pi s'est imposé vue que cette carte est caractérisée par un rapport qualité prix respectable. Le tableau ci-dessous montre les différentes possibilités du marché.

| Nom du Board | Prix euro | Processeur | Clock Speed | SoC | GPU | RAM | Mémoire | Max Mémoire | GPIO | USB | Ethernet | Wifi | HDMI | VGA | SD | Audio Out | Sata | Linux | Android |
|----------------|-----------|------------|-------------|-----------|---------------|--------|---------|-------------|------|-----|----------|------|------|-----|------|-----------|------|-------|---------|
| Arduino uno | 20 | AT mega328 | 16Mhz | / | / | 2 KB | 32 KB | 32 KB | 14 | 1 | add. | add. | non | non | add. | add. | non | non | non |
| Arduino Due | 39 | AT91 SAM3 | 85Mhz | / | / | 96 KB | 512 KB | 512 KB | 54 | 1 | add. | add. | non | non | add. | add. | non | non | non |
| Raspberry Pi | 26 | ARM 11 | 700Mhz | Broadcom | Video Core IV | 512 KB | non | 32 KB | 26 | 2 | oui | add. | oui | non | yes | yes | non | oui | add. |
| Cubie Board | 36 | ARM Cortex | 1Ghz | Allwinner | ARM 400 | 1 GB | 4 GB | 32 GB | 96 | 2 | oui | add. | oui | non | non | oui | oui | oui | oui |
| Gooseberry | 46 | ARM Cortex | 1Ghz | Allwinner | ARM 400 | - | 4 GB | 32 GB | non | 0 | oui | oui | oui | non | non | oui | non | add. | non |
| APC Rock | 59 | ARM Cortex | 800Mhz | Wondmedia | ARM 400 | - | 4 GB | 32 GB | 24 | 2 | oui | add. | oui | non | non | oui | non | add. | non |
| A13 Olinu Xino | 55 | ARM Cortex | 1Ghz | Allwinner | ARM 400 | 512 MB | 4 GB | 32 GB | 68 | 3 | add. | oui | non | non | non | oui | non | - | oui |
| A10 Olinu Xino | - | ARM Cortex | 1Ghz | Allwinner | ARM 400 | 1 GB | 4 GB | 32 GB | 132 | 3 | oui | add. | non | non | non | oui | oui | - | oui |
| Hackberry | 48 | ARM Cortex | 1.2Ghz | Allwinner | ARM 400 | 512 MB | 4 GB | 32 GB | non | 3 | oui | oui | oui | non | oui | oui | oui | oui | oui |

TABLE 4.1 – Tableau Comparatif

4.2.2.2 Webcam Logitech HD C310

Au cours des premiers tests nous avons opté pour les webcams low-cost qui fournissent un flux d'image brute (YUV ou YUV2) mais on s'est vite rendu compte que cette solution n'est pas efficace. En effet pendant ces tests nous avons remarqué que ce flux n'est pas facile à traiter donc il est nécessaire de faire l'encodage des images sous un autre format plus adapté. On a tout de suite pensé à l'encodage avec la GPU du Raspberry Pi mais, pour des limitations matérielles, les API offert par le constructeur ne permettent pas l'encodage des images en provenance directe

d'une webcam. Et l'alternative d'encodage logicielle ne se pose même pas vu les ressources très limitées du Raspberry Pi. Il est donc inévitable d'acheter une camera HD qui soit dotée d'un encodeur qui offre un flux d'image encodé sous format JPG et donc facile à manipuler.

4.2.2.3 Servomoteur Hitec HS-311

Pour assurer le mouvement de la camera sur plusieurs degrés il était nécessaire d'utiliser des moteurs. Alors notre choix s'est porté principalement sur deux types de moteur : soit les moteurs pas à pas soit les servomoteurs. Nous avons tout de suite choisi les servomoteurs puisque ces derniers sont facilement pilotable à l'aide d'un signal PWM généré par le moyen des pins du Raspberry Pi. De plus il ne nécessite pas l'utilisation d'une carte de commandes très compliquée comme c'est le cas des moteurs pas à pas.

4.3 Réalisation de la partie opérative

Une fois la conception achevée, nous pouvons entamer la réalisation de la partie opérative qui comporte deux volets essentiels ; le circuit d'alimentation et le dispositif mécanique.

4.3.1 Réalisation du circuit d'alimentation

- Régulateur : Notre choix c'est fixé sur le LM2576T-5.0. En effet ce régulateur permet de fournir un courant de tension 5v (qui alimentera le Rasberry Pi et les deux moteurs) tout en assurant une intensité de 3A. Ce régulateur sera alimenté par une tension variant de 7 à 40 V.
- Condensateur $100 \mu F$: Il sera placé entre l'alimentation et le régulateur.
- Condensateur $1000 \mu F$: Il sera placé avant la sortie.
- Bobine 100^*H : Placée avant la sortie, elle assurera une intensité de 1,5A.
- Diode : Elle aura un rôle de sécurité.
- Connecteurs mâles : Nous placerons en premier lieu deux qui seront reliées au RasberryPi pour la transmission des commandes spécifiques au deux servomoteurs, suivie de 6 pines ; 4 pour l'alimentation et 2 pour les commandes des deux servomoteurs .



FIGURE 4.6 – Circuit d'alimentation

Pour le bon fonctionnement de notre système nous avons besoin d'un courant de tension 5v et d'intensité 3 ampères, or l'alimentation par défaut du RaspberryPi ne peut délivrer qu'une intensité de 1,5 ampère au maximum, ce qui nous a amené à développer une source d'alimentation externe.

Cette dernière est composé de : Afin de relier ces composants nous avions eu besoin d'imprimer notre propre carte électronique (PCB).

La réalisation d'une carte de circuit imprimé passe par plusieurs étapes.Une fois l'étape de conception de la carte est finie, on imprime le dessin obtenu sur une feuille de transparent, ensuite nous passons à l'étape de gravure en utilisant une insoleuse à ultra-violet, le révélateur basique et le perchlorure de fer. Enfin on effectue le perçage de la carte on soude les différents composants dans leur emplacement approprié.

4.3.2 Réalisation du dispositif mécanique

Nous allons dans cette partie détailler l'assemblage de notre système et sa réalisation.

Vue de l'extérieur notre système comporte une caméra et un servomoteur le tout bien fixé sur un boitier comme le montre bien la figure 4.7



FIGURE 4.7 – Vue externe du système

La servomoteur fixé sur le boitier permettra à la caméra d'effectuer une rotation de 180° degrés et ainsi de couvrir le demi-plan vertical. Le boitier permet non seulement la fixation de la caméra et du servomoteur mais surtout l'abstraction des composantes internes ce qui permettra plus de sécurité.

La figure 4.8 expose les composantes internes de notre système.

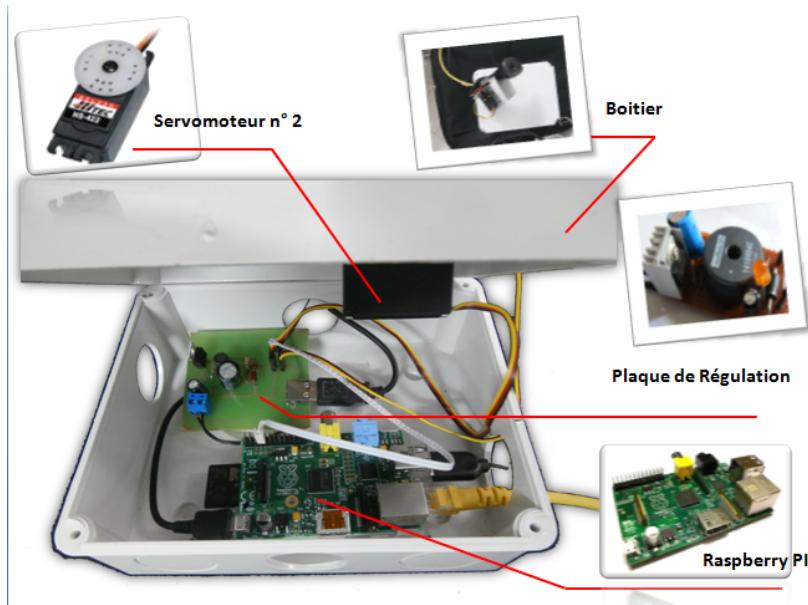


FIGURE 4.8 – Vue interne du système

Comme le montre la figure 4.8 le deuxième servomoteur est placé juste à l'intérieur du boitier il permettra la rotation horizontale du système servomoteur externe, caméra et donc à la caméra de couvrir toute une pièce.

La Raspberry Pi est relié à la carte d'alimentation à travers deux pins ce qui nous permettra de passer les commandes PWM aux moteurs.

Conclusion

Dans ce chapitre, nous avons décrit la conception et la réalisation de la partie opérative de notre application et ce en mettant en place la partie opérative et la justification des choix technologiques du matériel ainsi qu'en montrant les étapes de réalisation du dispositif. C'est à ce niveau que s'arrête ce chapitre. Dans le prochain chapitre, nous détaillons les outils et les langages utilisées ainsi que les interfaces de notre application.

Chapitre 5

Conception et réalisation de la partie logicielle

Introduction

Au fil des chapitres, l'allure de notre projet se trace. Nous arrivons, petit à petit, à avoir une idée claire et concise sur la réalisation finale du système. Cette partie est le fruit d'un travail de recherche d'une solution qui peut assurer les fonctionnalités proposées. Il faut surtout prendre en considération les limitations de la partie matérielle.

Cette partie se décompose en trois grandes sous-parties : la conception détaillée de l'application, le choix de l'environnement de travail logiciel et finalement la réalisation de l'application.

5.1 Conception détaillée de l'application

Dans cette section nous nous intéresserons à la conception détaillée de notre application. Nous allons modéliser la vue statique de notre logiciel à travers le diagramme des classes. En ce qui concerne les vue dynamique nous allons la modéliser en utilisant les diagrammes de séquence et les diagrammes d'état/transition. Nous allons finir par la présentation du schéma de la base de donnée qui va servir de moyen d'authentification des utilisateurs.

5.1.1 Diagramme de classe

Le langage JavaScript est un langage de programmation orienté objet pur. Tous les variables définie sont des objets et possède des méthodes par défaut. Nous avons donc choisi de décom-

poser en une classe principale appelée app, une classe moteur qui englobe toutes les méthodes relatives au commande des servomoteurs et une dernière classe camera qui a pour but de gérer le flux des images à partir de la camera. La figure 5.1 montre les différentes classes à utiliser et les différentes relations qui les unies.

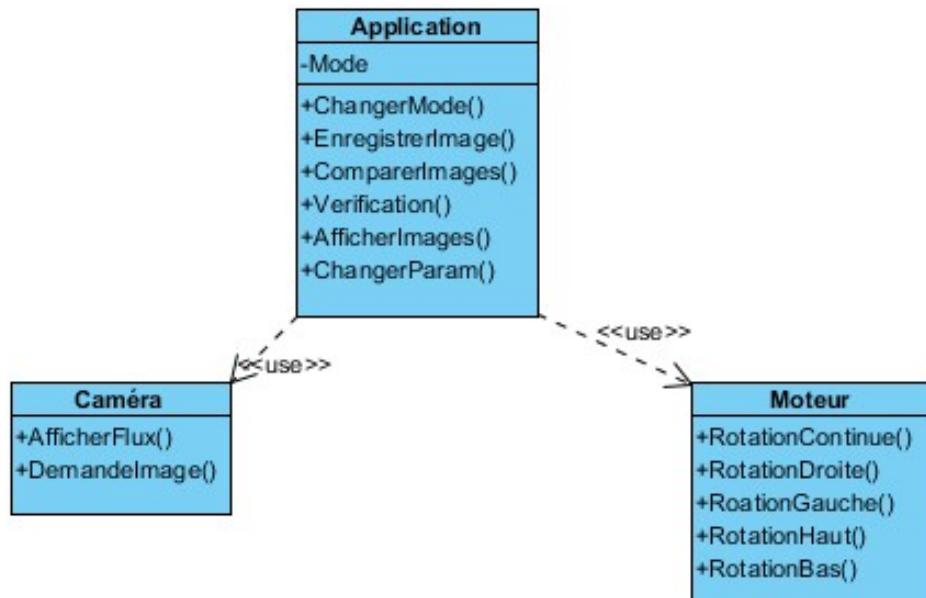


FIGURE 5.1 – Diagramme de classe

5.1.2 Diagramme de séquences

Afin de représenter les interactions entre les acteurs et le système selon un ordre chronologique, nous présentons quelques séquences des différents traitements à réaliser.

Donc, nous allons nous intéresser aux différents scénarios d'utilisation déjà abordé dans la partie spécification, et nous montrerons le rôle de chaque classe dans la réalisation de la fonction globale du système. Cette partie comportera donc trois sous partie. Chaque sous partie décrit une vue dynamique du système. Notons enfin que nous avons choisi de représenter les séquences de déroulement normales de l'exécution de l'application.

5.1.2.1 Diagramme de séquences de l'authentification

La figure 5.2 montre les interactions entre l'utilisateur et la classe application afin d'établir une connexion

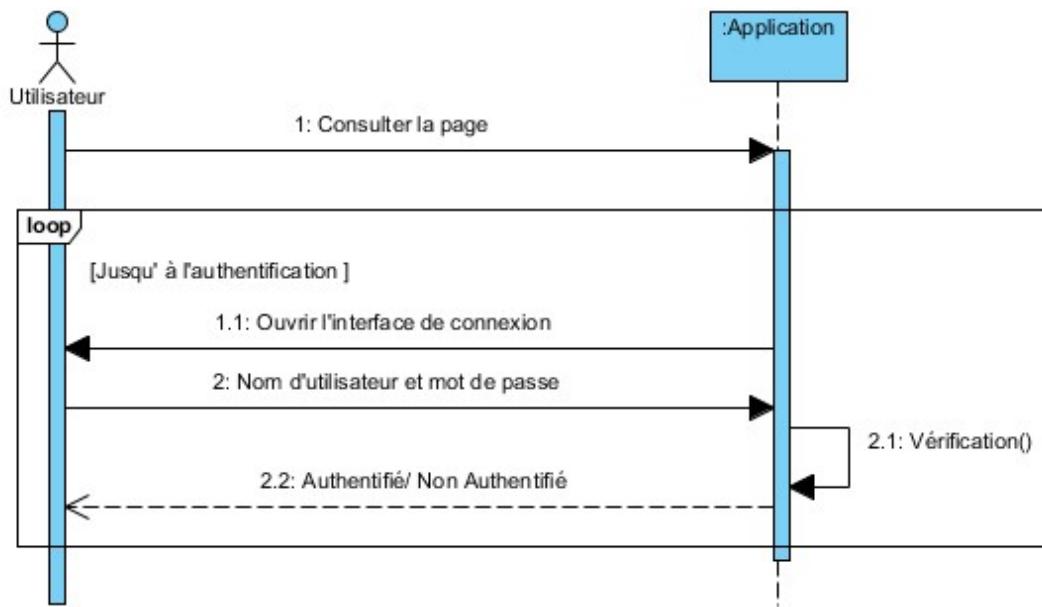


FIGURE 5.2 – Diagramme de séquence : Authentification

5.1.2.2 Diagramme de séquences du mode d'utilisation normal

Un utilisateur connecté peut utiliser l'application en mode normale. La figure 5.3 illustre les interactions entre les classes pour assurer le fonctionnement en mode normal.

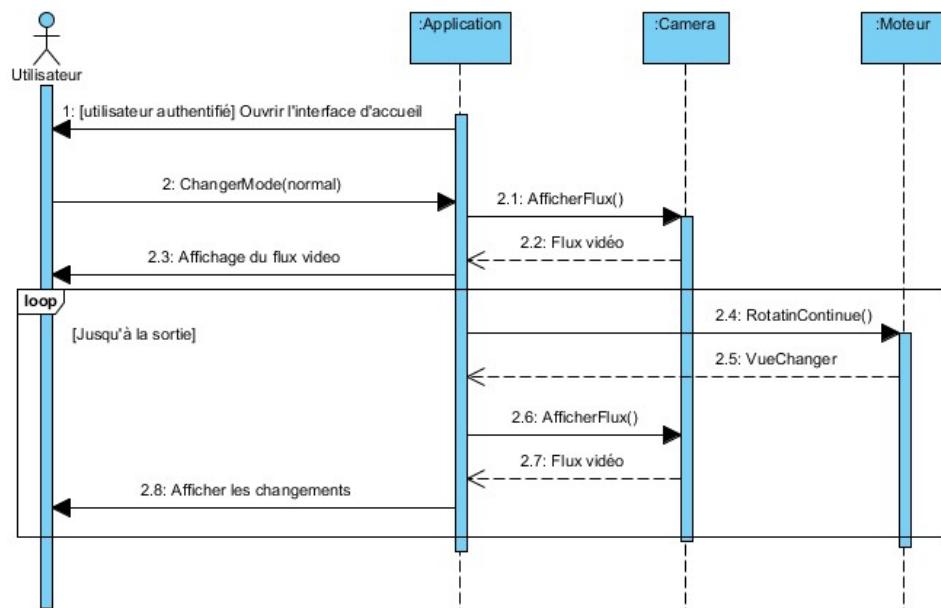


FIGURE 5.3 – Diagramme de séquence : mode de fonctionnement normal

5.1.2.3 Diagramme de séquences du mode de fonctionnement assisté

Dans ce mode de fonctionnement, l'utilisateur est libre de piloter le système de surveillance. La figure 5.4 met l'accent sur les interactions entre les classes afin de permettre à l'utilisateur de bénéficier de ce mode d'utilisation.

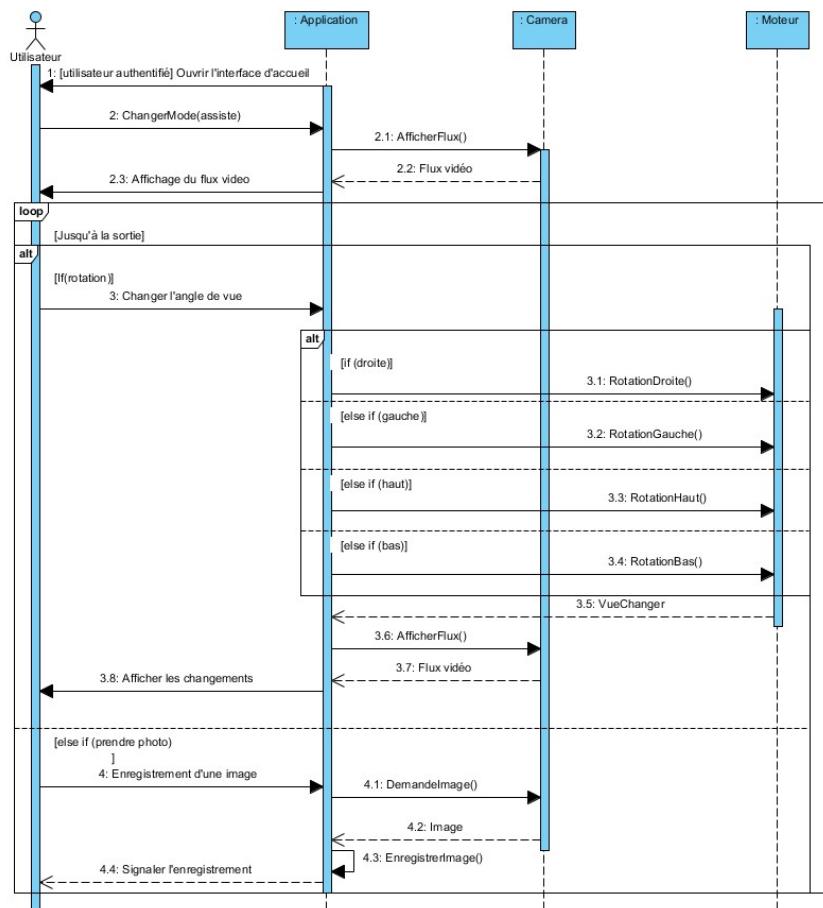


FIGURE 5.4 – Diagramme de séquence : mode de fonctionnement assisté

5.1.2.4 Diagramme de séquences du mode d'utilisation avec détection du mouvement

Ce mode de fonctionnement est certainement l'un des plus compliqué. Dans ce mode le système alertera l'utilisateur de l'existence d'un mouvement. Ce fonctionnement est illustré par la figure 5.5.

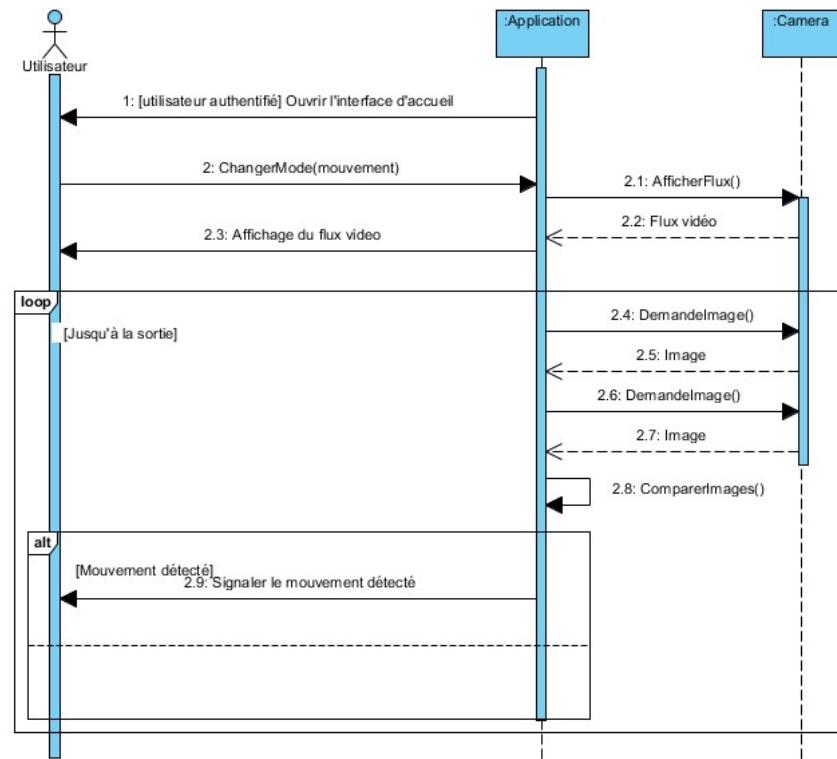


FIGURE 5.5 – Diagramme de séquence : mode d'utilisation avec détection du mouvement

5.1.3 Diagramme d'état/transition

Notre système de surveillance fonctionne principalement en trois modes. Ces modes sont le mode de fonctionnement standard, le mode de fonctionnement assisté et le mode de fonctionnement avec détection de mouvement. Pour ce faire, notre classe principale va passer par différents états en fonction du mode du système. Les différents états de notre application vont être modélisés par le diagramme d'états/transitions de la figure 5.6.

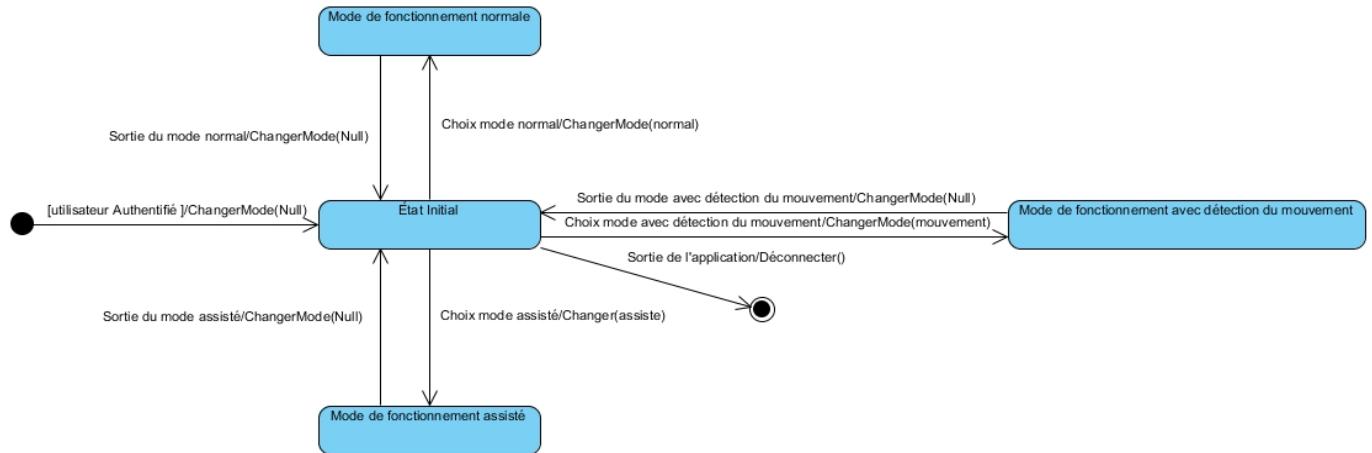


FIGURE 5.6 – Diagramme d'état/transitio

5.1.4 Schéma de la base de donnée

L'usage de notre application de surveillance nécessite une authentification de l'utilisateur. Pour stocker les paramètres d'authentification de manière sûre, il est impératif d'utiliser une base de données. Cette base de données contiendra essentiellement une seule table qui va contenir les paramètres que nous jugeons utiles pour la sécurité de notre application. La figure 5.7 nous renseigne sur le schéma de notre base de données.

| Authentification | | | |
|------------------|--------------------------|--------------------|--|
| | id | integer(10) | |
| | nom | varchar(20) | |
| | prenom | varchar(20) | |
| | username | varchar(20) | |
| | mot de passe | varchar(20) | |
| | email | varchar(20) | |
| | Date de naissance | date | |

FIGURE 5.7 – Schéma de la base de donnée

5.2 Environnement Logiciel

Notre environnement logiciel est composé principalement de quatres composants logiciels qui forment une base solide sur lequel tourne notre application. Nous pouvons donc citer la distribution « Linux Arch Linux ARM » qui constituera le système d'exploitation à installer sur le Raspberry Pi, Node.js qui fournira les API de programmation de notre page web, Mjpg-streamer qui assurera l'extraction du flux vidéo à partir de la camera et finalement Pi-Blaster le programme qui nous permettra de commander les servomoteurs à partir du Raspberry Pi. Cette section se décompose en deux sous-sections : choix de l'environnement qui contient une brève description de chaque composant de l'environnement et la justification des choix de ces derniers.

5.2.1 Choix de l'environnement logiciel

Nous nous intéressons dans cette partie à l'environnement logiciel de notre système.

5.2.1.1 La distribution Linux « Arch Linux ARM »

Arch Linux ARM est une adaptation de la distribution Arch Linux pour les processeurs ARM. Cette distribution vise la simplicité et un contrôle total à la carte utilisé. Cette distribution fournit une structure de base légère qui permet à l'utilisateur de façonner le système à ses besoins. Elle est aussi optimisée pour chaque version d'architecture des processeurs ARM pour permettre ainsi une efficacité maximale.

5.2.1.2 Node.js

Node.js est un Framework construit sur la machine d'exécution JavaScript de Google Chrome (connue sous le nom de V8) pour construire facilement des applications réseau rapides et évolutives. Node.js utilise un modèle à la fois event-driven et non-bloquant d'entrée / sortie qui le rend léger et efficace. Il trouve son intérêt pour le traitement des données volumineuses en temps réel et pour les applications réparties. Le langage utilisé pour programmer en node.js est le JavaScript.

5.2.1.3 MJPG-Streamer

"MJPG-streamer" est une application en ligne de commande qui copie les images JPG d'une seule source d'entrer vers des clients multiples. Il peut être utilisé pour diffuser un flux d'image JPG en continu sur un réseau IP à partir d'une webcam. Ce flux peut alors être lu par un navigateur web ou toute autres applications. MJPEG-streamer est conçu pour les systèmes embarqués avec des ressources très limitées en termes de mémoire et de processeur. Et il est compatible principalement avec les systèmes d'exploitation à base de noyau GNU/Linux.

5.2.1.4 Pi-Blaster

Ce logiciel conçu sur la base du projet ServoBlaster fourni un moyen de générer un signal MLI (PWM) de manière logicielle pour pouvoir ainsi commander plusieurs servomoteur à la fois. Ce logiciel s'installe directement sur le Raspberry Pi et peut être utilisé via l'interpréteur de commande de Linux.

5.2.2 Justification du choix de l'environnement logiciel

Dans ce qui suit nous allons justifier chacun des choix logiciel effectué.

5.2.2.1 La distribution Linux « Arch Linux ARM »

Cette distribution Linux est la plus légère de toutes les distributions disponibles sur le Raspberry Pi .Elle ne contient pas d'interface graphique par conséquent elle démarre en 10 s et ne contient un nombre très limité paquet préinstallé. C'est aux développeurs de prendre soin du choix des paquets à installer selon le domaine d'application. Un autre atout de cette distribution est qu'elle est caractérisé par un dépôt de logicielle en ligne très riche ce qui nous permet d'installer plusieurs logiciel en ligne et ainsi évité les problèmes liés à la compilation et à la cross-compilation des logiciel à partir des sources. En plus de ça cette distribution est caractérisé par un modèle de développement « Rolling Release » ce qui assuré d'avoir la dernier version stable de chaque logiciel. Et donc ainsi d'avoir les meilleures performances possibles.

5.2.2.2 Node.js

D'un point de vue assez simpliste node.js est un interpréteur de commande en langage JavaScript qui permet non seulement de créer des applications web, mais aussi d'exécuter du

code JavaScript du côté serveur de manière totalement transparente par rapport au client et c'est là tout son intérêt pour notre projet. En effet cette propriété va nous permettre à notre système de fonctionné de manière autonome. En d'autre terme même s'il n'y a aucun client connecté à notre page web le système peut effectuer la détection du mouvement et l'enregistrement des images sans l'intervention de quiconque. Un autre atout majeur de node.js par rapport aux autres outils de développement web est qu'il est basé sur un modèle non bloquant d'entrée sorti. Ce qui fait que toutes les instructions du code JavaScript sont non bloquantes donc tant qu'il y a du code à exécuter node.js ne va jamais s'arrêter. On obtient ainsi une exécution pseudo-parallèle des instructions et donc on garantit un haut degré de concurrence le tout avec un seul thread et un processeur monocorde, ce qui est le cas du Raspberry Pi.

5.2.2.3 Mjpg-Streamer

Node.js est encore en phase de développement à l'heure où nous rédigeons le rapport on est à la version 0.10.5. Donc il manque beaucoup d'API de programmation notamment celle relative à la récupération d'un flux vidéo à partir de la caméra et le traité. Mais heureusement, pour nous qu'on peut compléter le fonctionnement de ce dernier par le moyen des paquetages de bibliothèque et des différents paquets tiers qui complète le fonctionnement de ce dernier. Dans ce contexte, nous allons recourir à Mjpg-Streamer, ce programme Linux est capable d'extraire un flux d'image brute sans aucun traitement ni modification ce qui va nous permettre un gain considérable en terme de ressource matériel notamment en utilisation du processeur. Mjpg-streamer constitue un choix crucial pour le bon fonctionnement de notre système. En effet on a testé plusieurs autres logiciels qui fournissent le même service mais on s'est trouvé avec une demande exorbitante en terme de processeur qui ne permettent pas l'exécution d'autre programme sur la carte et la table 5.1 illustre un petit comparatif entre les différents logiciels utilisés et leurs consommation en % du processeur en utilisant le même environnement matériel et la même distribution.

| | Fonctionnalité assurée | % Demande en ressource CPU |
|---------------|------------------------|----------------------------|
| Motion | Flux transcodé | Entre 30% et 60% |
| ffmpeg | Flux transcodé | Entre 60% et 100% |
| Mjpg-streamer | Flux brute | Entre 3% et 5% |

TABLE 5.1 – Comparatif entre les logiciels d'extraction des images

5.2.2.4 Pi-Blaster

Pi-Blaster est logiciel qui peut être installer sur le Raspberry Pi et qui permettra de générer un signal MLI (PWM) en tapant en ligne de commande terminal les paramètres choisis. Le choix de ce logiciel par rapport aux autres possibilités est parce qu'il utilise une technique extrêmement efficace : il produit des impulsions très stables et n'est pas gourmand en terme de CPU. En plus de ça il est facile d'installation et d'utilisation avec node.js par le moyen d'un paquetage de bibliothèque.

5.3 Réalisation de l'interface de surveillance

Dans cette partie nous allons décrire les différentes parties de notre interface web ainsi que leur principe de fonctionnement. Cette partie se décompose principalement en sept pages web dont trois réalisent les fonctionnalités principales du système.

5.3.1 Interface de connexion

Cette page web permettra à l'utilisateur de se connecter au site web de notre système de surveillance. L'utilisateur est demandé de fournir le nom d'utilisateur et le mot de passe. La vérification de l'identité de l'utilisateur se fait grâce au système de gestion de base de données « MongoDB ». La figure 5.8 montre l'interface web avec les différents champs à remplir.

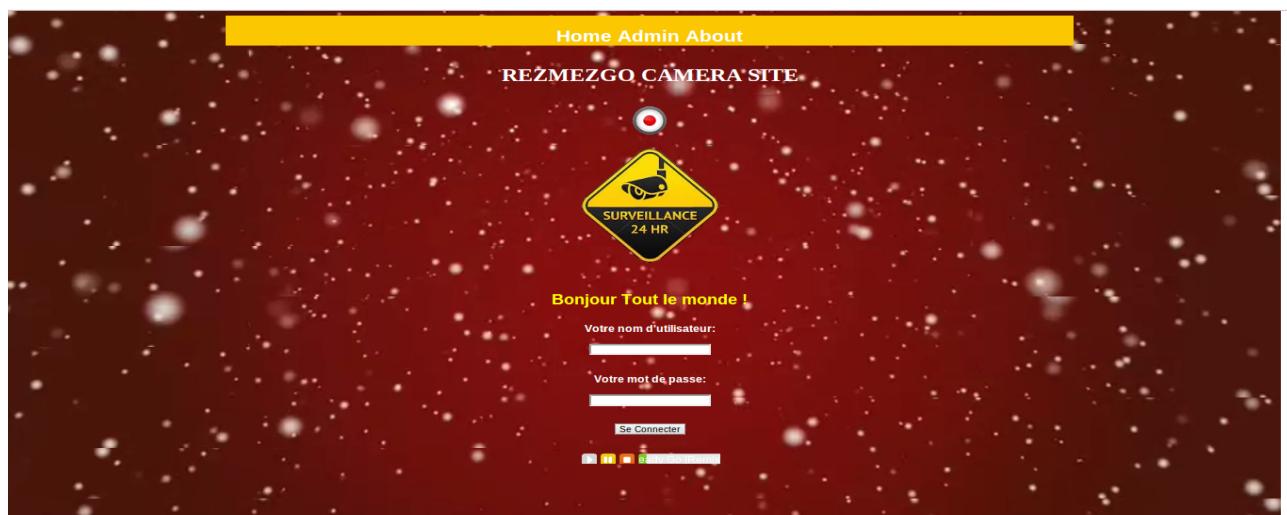


FIGURE 5.8 – Interface de connexion

5.3.2 Interface d'accueil

La page d'accueil des utilisateurs est une interface qui va nous permettre d'afficher les différentes images stockées dans notre système de surveillance. L'utilisateur peut alors récupérer les images supprimer les images. Après récupération des images l'utilisateur peut, en utilisant un logiciel convenable, combiner le flux d'images dans une vidéo. La figure 5.9 montre l'allure de l'interface d'accueil.

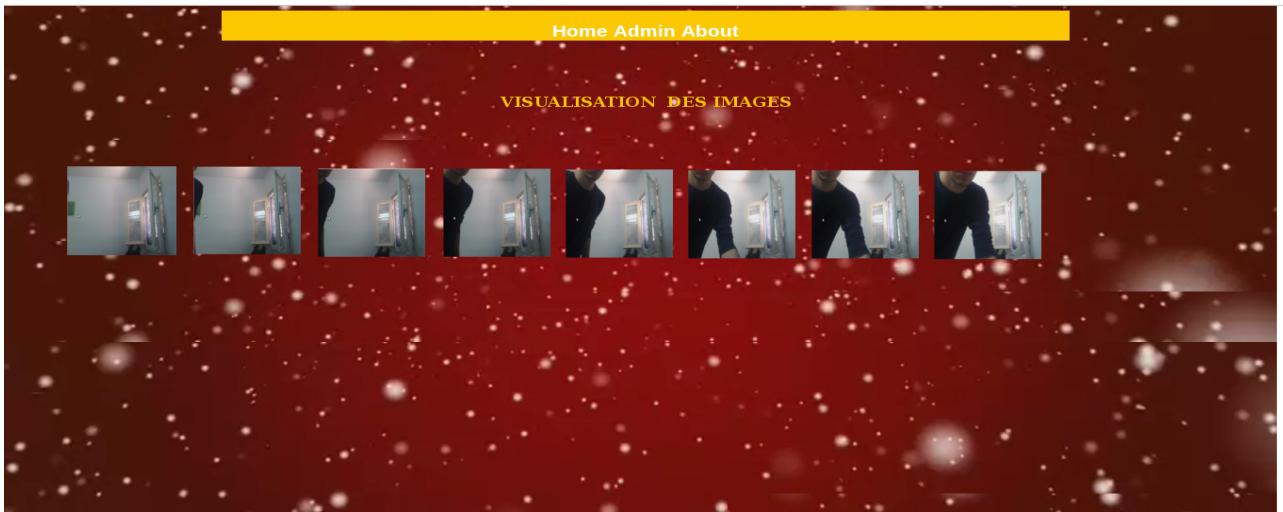


FIGURE 5.9 – Interface d'accueil

5.3.3 Interface de l'utilisateur

L'interface utilisateur se décompose principalement en trois sous interfaces selon le mode d'utilisation du système.

5.3.3.1 Mode de fonctionnement normal

En mode de fonctionnement normal l'utilisateur peut visualiser le flux vidéo uniquement. Le système se comporte de façon autonome et effectue une rotation continue dans plusieurs sens pour couvrir le maximum d'espace possible. La figure 5.10 montre l'interface du système en mode fonctionnement normal.



FIGURE 5.10 – Mode de fonctionnement normal

5.3.3.2 Mode de fonctionnement assisté

Dans ce mode de fonctionnement le système permet à l'utilisateur de prendre le contrôle du système de surveillance à l'aide des moteurs il peut changer l'angle de vue de la camera. La commande des servomoteurs nécessite l'utilisation d'une technique de génération de signal appelé modulation de largeur d'impulsions. Ce signal généré avec le logiciel Pi-Blaster nous permettra de commander les deux servomoteurs. La figure 5.11 montre l'interface de fonctionnement assisté.

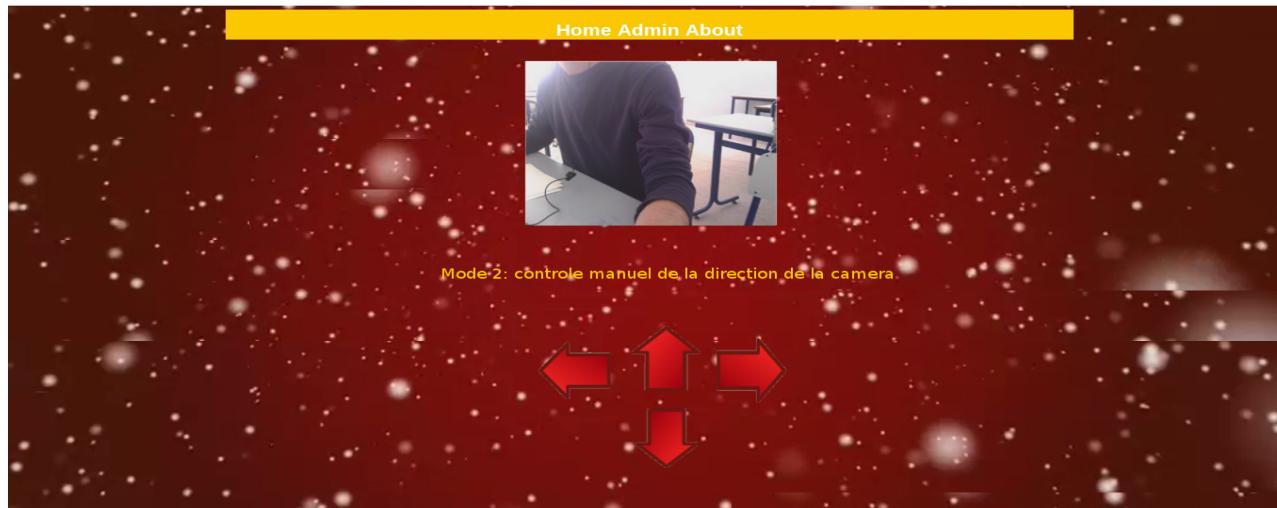


FIGURE 5.11 – Mode de fonctionnement assisté

5.3.3.3 Mode de fonctionnement avec détection du mouvement

La détection du mouvement suppose l'utilisation de technique spécifique de traitement d'image. Faute de limitations du système en terme de calcul nous avons opté pour une méthode classique de détection du mouvement qui est la soustraction pixel par pixel. La camera génère un flux vidéo continu en images. Afin de traiter le flux vidéo, l'application devra réaliser les fonctions suivantes :

- Séquencer le flux vidéo en images.
- Comparer les images deux à deux de façon successive.
- Déetecter les points de différences entre les images comparées.
- Signaler le mouvement à l'utilisateur.

Nous pouvons aussi améliorer la vitesse de traitement de cette application en redimensionnant l'image obtenue à partir de la camera lors de la comparaison de telle sorte que le temps de traitement des images diminue de façon considérable.

La première phase de traitement consiste en une différence entre les deux images ainsi obtenu en une matrice de taille celle obtenu lors de la capture (ou celle que nous avons redimensionné). Chaque case de la matrice étant une valeur entre 0 et 255 si on fait la différence en appliquant la valeur absolue on obtient des valeurs entre 0 et 255.

La deuxième phase du programme consiste en un seuillage de cette matrice de différence. Nous

utiliserons une valeur seuil qui représente le taux de précision de l'application. Toute valeur au-dessus de ce seuil passe à 255 et toute valeur en dessous de ce seuil passe à 0. Le résultat est donc une image noire et blanc. En calculant le nombre de bit blanc on peut vérifier s'il y a un mouvement ou pas.

La figure 5.12 montre l'interface utilisateur en mode de détection du mouvement avec aucun mouvement détecté .



FIGURE 5.12 – Mode de fonctionnement avec détection du mouvement

La figure 5.13 montre l'interface utilisateur en mode de détection du mouvement avec un mouvement détecté .

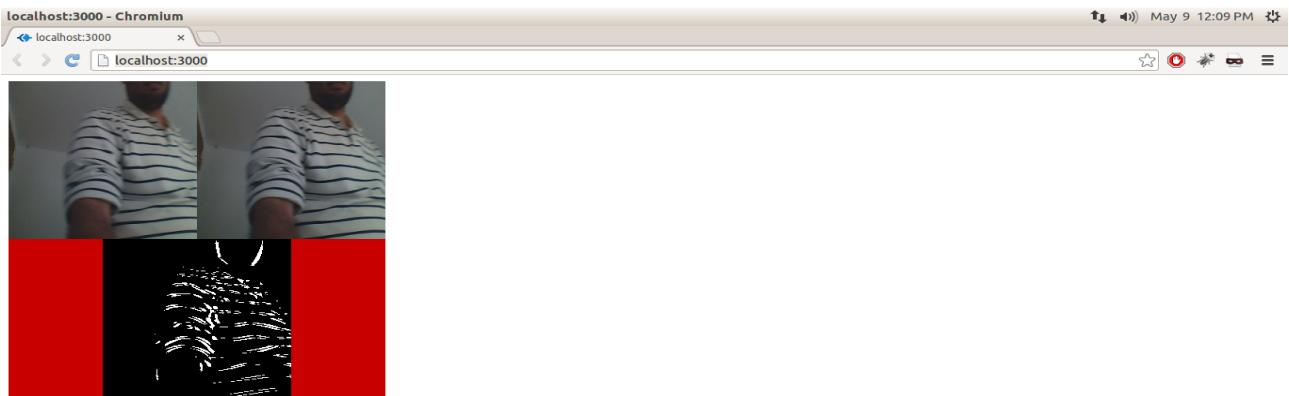


FIGURE 5.13 – Mode de fonctionnement avec détection du mouvement

5.3.4 Interface de l'administrateur

L'administrateur est un utilisateur privilégié. Il est unique pour notre système de surveillance. Il a le privilège de gérer les comptes de tous les utilisateurs du système. La figure 5.14 montre l'interface d'administration avec toutes les fonctionnalités qu'elle offre.

A screenshot of an "Interface Admin" web page. The header features a yellow navigation bar with links for "Home", "Admin", and "About". Below the header, the title "Interface Admin" is centered. The main content is a form for creating a new user account. The fields are labeled "Nom", "Prenom", "Username", "Password", ".email", and "Date de naissance". Each field has a corresponding input text box below it. At the bottom of the form is a single button labeled "Valider". The background of the page is a dark red color with a subtle starry texture.

FIGURE 5.14 – Interface de l'administrateur

5.3.5 Interface d'aide

La page d'aide est une page informative qui permet à l'utilisateur de connaitre le contexte de notre projet, les fonctionnalités du système et la manière avec laquelle on peut exploiter le système. La figure 5.15 nous donne un aperçu global sur l'interface d'aide.

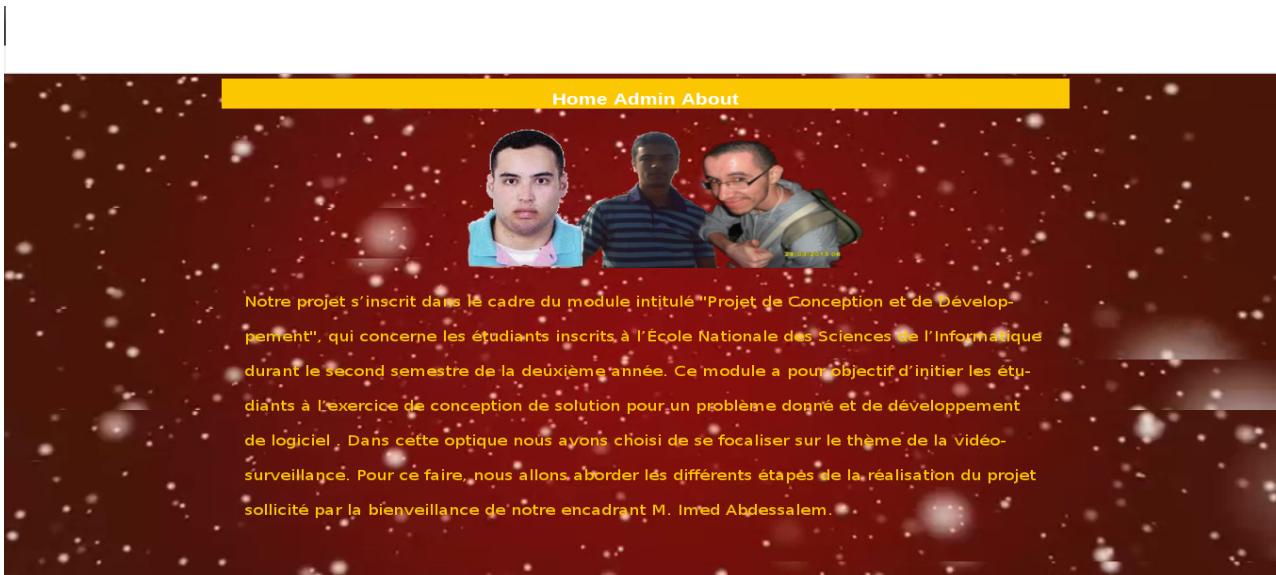


FIGURE 5.15 – Interface d'aide

Conclusion

En concluant ce chapitre, nous aurons fini avec la partie de conception et réalisation logicielle. Cette partie nous a permis de bien choisir les composants logiciels de notre application en fonction des fonctionnalités proposées.

Conclusion et perspectives

Atravers ce projet, nous nous sommes intéressés à concevoir et fabriquer un système de vidéo-surveillance fonctionnel.

Notre travail a débuté par une phase théorique, à travers laquelle nous avons présenté la vidéo-surveillance. Nous avons exposé en deuxième lieu l'analyse et la spécification des besoins. Ensuite le troisième chapitre a été consacré à la conception globale. En dernier lieu, nous avons conclu par la conception détaillée et réalisation qui décrit l'environnement matériel et logiciel utilisés ainsi que les interfaces de l'application.

Nous avions réussi à accomplir le projet et satisfaire les besoins mentionnées dans le cahier des charges, même si nous avons passé plus de temps que prévu dans la phase de l'étude de l'art étant donnée que nous n'avions pas assez de documentation sur la Raspberry Pi.

À défaut de temps, nous ne sommes pas arrivée à assurer l'authentification exclusive ce qui aurait laissé l'accès à un seul utilisateur à la fois. En effet, l'étendu de notre système n'est pas limité. Même si nous nous sommes limités à une seule caméra, deux servomoteurs, nous pouvons interfaçer d'autres capteurs et actionneurs, un autre type de caméra capable de fonctionner en mode nuit ou même plus de caméra et ceci selon le besoin de l'utilisateur, l'environnement d'exécution, etc.

Bibliographie

Getting Started with Raspberry Pi

By Matt Richardson, Shawn Wallace, O'RAILLY, December 2012

Getting To Know The \$35 ARM-Powered Linux Computer

Meet the Raspberry Pi

By Eben Upton , Gareth Halfacree , Kindle Edition, July 2012

The essential preview guide to getting started with Raspberry Pi ® computing and programming

Raspberry Pi - A Quick-Start Guide

By Maik Schmidt , Kindle Edition, November 6, 2012

The MagPi

Issue 06 , October 2012

Netographie

Node.js

Dernière visite le 2 Avril 2013

<http://node.codeschool.com/levels/1#>

Arch Linux On Raspberry Pi :Installing xorg and lxde

Dernière visite le 12 Avril 2013

<http://sudoxanthippe.wordpress.com>

Mjpg-Streamer

Dernière visite le 9 Mai 2013

<http://sourceforge.net/apps/mediawiki/mjpg-streamer/index.php>

Node.js Official web site

Dernière visite le 14 Avril 2013

<http://nodejs.org/>

Raspberry Pi Tutorials

Conclusion et perspectives

Dernière visite le 12 Avril 2013

http://elinux.org/RPi_Tutorials

Raspberry Pi Official Web Site

Dernière visite le 24 Avril 2013

<http://www.raspberrypi.org/>

Annexe A

Les servomoteurs

A.1 Les servomoteurs

Un servomoteur est un dispositif typiquement utilisé en modélisme pour, par exemple, diriger une voiture télécommandée. Il comprend : L'utilisateur envoie à chaque instant un signal modulé représentatif d'un angle. Le servomoteur tourne vers cet angle et s'y maintient.



FIGURE A.1 – Un servomoteur et ses accessoires

- Un moteur électrique de petite taille.
- Un réducteur en sortie de ce moteur pour avoir moins de vitesse et plus de couple.
- Un capteur : un potentiomètre qui produit une tension variable en fonction de l'angle de l'axe de sortie.
- Un asservissement électronique pour contrôler la position de cet axe de sortie.

A.2 Commande des servomoteurs

Un connecteur à trois fils permet de commander un servomoteur : Deux fils pour la masse et la tension (environ 5v). Ces deux fils sont en général noirs et rouges, et un troisième fil pour commander le servomoteur. Il est normalement blanc ou jaune. On pourrait dire sommairement qu'il faut fournir un signal PWM en tant que signal de commande, mais ce n'est pas tout à fait exact :

- Le PWM (Pulse Width Modulation : Modulation en largeur d'impulsion) est un signal dont le taux de modulation varie de 0 à 100% ($TM = \text{Temps/Tpériode}$).
- Pour un servomoteur on parle de Pulse-Code Modulated Signal (Signal modulé en code d'impulsion) où seule l'impulsion compte, pas la fréquence.

En clair, il faut fournir au servomoteur une impulsion à 1 (suivie d'un retour à 0). Le servomoteur va prendre en compte la largeur temporelle de cette impulsion qu'il va convertir de façon linéaire en un angle. La durée de retour à 0 de l'impulsion n'est pas critique, en pratique il faut éviter de dépasser 20ms entre deux fronts montants. À noter aussi que ce système présente un avantage par rapport au PWM : une absence de signal (toujours à 1, ou à 0) laisse le servomoteur en "roue libre" comme s'il n'était pas alimenté.

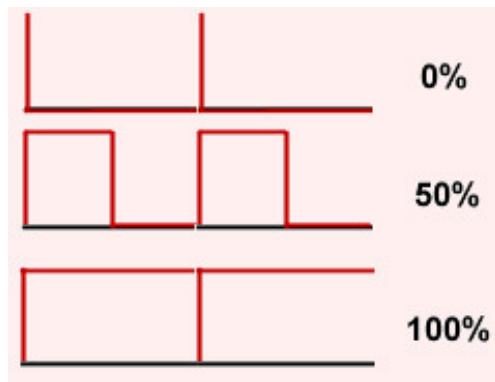


FIGURE A.2 – Signal PWM

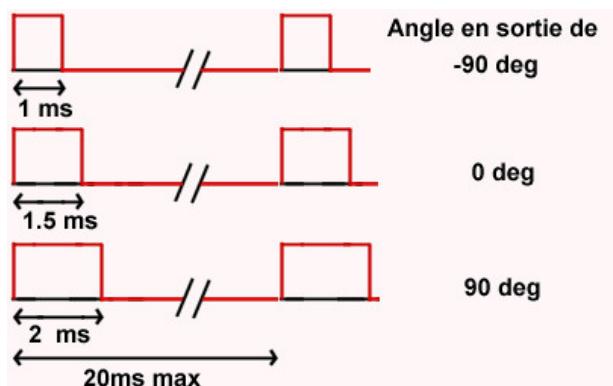


FIGURE A.3 – Signal modulé en code d'impulsion

Annexe B

Régulateur de tension

B.1 Régulateur de tension

Un régulateur de tension, est un composant électronique qui maintient à sa sortie une tension constante, indépendamment, de la charge et de la tension d'entrée. Il y a principalement deux types de régulateur : les régulateurs linéaires et les régulateurs à découpage. Nous avons opté pour les régulateurs à découpage.

B.2 Avantage des régulateurs à découpage par rapport aux régulateurs linéaires

Le principal avantage des régulateurs à découpage est leur rendement plus élevé et leur taille plus petite. Les régulateurs de type série et parallèle classiques fonctionnent avec un mode de conduction continu, et dissipent des quantités de puissance relativement élevées. Le rendement des régulateurs linéaires est habituellement de l'ordre de 40 à 50 %. Lorsque la différence de tension entre l'entrée et la sortie est élevée, le rendement qui en résulte est largement inférieur à 40 %.

Les régulateurs à découpage ont des rendements typiques de l'ordre de 60 à 90 %, bien plus élevés que ceux des régulateurs linéaire, de type série ou parallèle. Les régulateurs à découpage parviennent à ces rendements élevés grâce à leur transistor de puissance de découpage qui est toujours complètement à l'état passant ou à l'état bloqué, excepté lorsqu'il passe d'un état à l'autre. Il en résulte soit une tension faible, soit un courant faible pendant la plus grande partie de son fonctionnement. Les régulateurs à découpage utilisent le rapport cyclique entre

l'état passant et l'état bloqué du transistor de découpage pour réguler la tension et le courant de sortie. Leur fréquence étant beaucoup plus élevée que celle de la ligne, il est possible de réduire la taille, le poids et donc le coût des transformateurs, des condensateurs, des bobines et des autres éléments de filtrage.

B.3 Les régulateurs LM1576-ADJ

Les régulateurs LM1576-ADJ sont des circuits intégrés monolithiques basé sur un composant actif travaillant en commutation. Ces Composants disposent d'une tension de sortie, qui est réglable de 1,23 V à 37V, et est capable de délivrer jusqu'à 3A de courant avec une excellente régulation de tension et du courant.

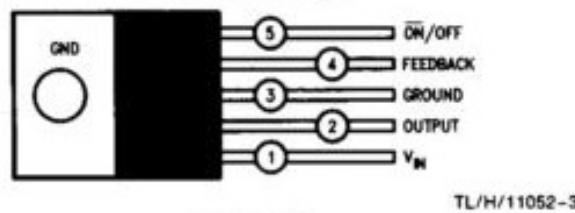


FIGURE B.1 – Schéma du régulateur utilisé

B.4 Le schéma du circuit d'alimentation

La figure B.2 montre le schéma du circuit conçu dans le cadre de ce projet.

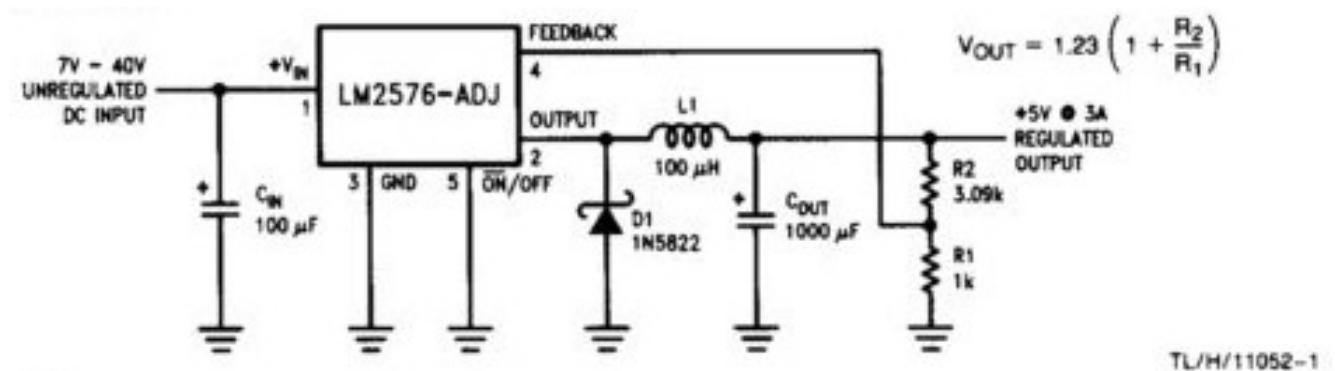


FIGURE B.2 – Schéma du circuit d'alimentation

Résumé

Ce projet consiste à construire un système de surveillance basé sur le web. Nous avons implémenté une interface web avec Node.JS qui de changer le champ de vision de la caméra par l'intermédiaire de deux moteurs. De plus notre application est capable détecter les mouvements.

Mots clefs : Raspberry Pi, Servomoteur, JavaScript, Node.JS, Mjpg-streamer, détection de mouvement...

Abstract

The aim of this project is to build a web-based surveillance system. We have implemented a web interface using Node.js that change the field of view of the camera by means of two motors. Moreover, our application is able to detect motion.

Keywords : Raspberry Pi, servomotor, JavaScript, Node.JS, MJPG-streamer, motion detection...

ملخص

يهدف هذا المشروع الى بناء نظام مراقبة يعتمد على شبكة الانترنت. و قمنا بإنشاء واجهة الويب، باستعمال نود.جي آس ، تسمح بتخزين الصور على الخادم و التحكم بها. وعلاوة على ذلك تطبيقنا قادرة على تغيير مجال رؤية الكاميرا عن طريق محركيين اثنين .
كلمات مفاتيح شريحة رازيري باي ، محرك سرفو موتور ، جافا سكريبت ، نود.جي آس ، ام جي بي جاي ستريتار ، كشف الحركة ...