

REPUBLIQUE TOGOLAISE

-----  
Travail – Liberté – Patrie

PRESIDENCE DE LA REPUBLIQUE

-----  
MINISTÈRE DE LA PLANIFICATION DU  
DÉVELOPPEMENT ET DE LA COOPÉRATION



Institut Africain d'Informatique  
Représentation du Togo (IAI-TOGO)

Tel: 22 20 47 00

Site web: [www.iai-togo.tg](http://www.iai-togo.tg)

07 BP 22456 Lomé 07 TOGO



Digital Deep VISION

Tel : 90 09 91 11 / 97 76 97 97

Email: [contact@digital-deepvision.com](mailto:contact@digital-deepvision.com)

Site web: [www.digital-deepvision.com](http://www.digital-deepvision.com)

**PROJET DE FIN DE FORMATION POUR L'OBTENTION DU DIPLOME DE LICENCE  
PROFESSIONNELLE EN INFORMATIQUE**

OPTION :

**ADMINISTRATION DES SYSTEMES ET RESEAUX**

Conception et implémentation d'un système de contrôle  
informatique de la consommation d'énergie électrique d'un habitat.

Période : du 16 mai au 13 aout 2022

Rédigé et soutenu par :

**GONCALVES Koku Jean Hilaire**

Etudiant en troisième année

Année Académique : 2021 – 2022

**SUPERVISEUR :**

M. DJAGRE Kofi Napo

Enseignant à IAI-TOGO

**MAITRE DE STAGE :**

M. KABATE M'Ponada Paul

Directeur Digital Deep Vision

## DEDICACES

### A

#### **mes parents**

Kodjogan Dayé Jeannot et Natekie  
Theresa, ce travail est le fruit de votre  
amour, votre encouragement, votre  
soutien et votre confiance.

#### **mes sœurs**

Sandrine et Edna, ce travail est le  
résultat de tous votre amour et votre  
disponibilité.

#### **A mes tantes**

Requia constance, mélanie et juguette  
Pour votre présence et vos  
encouragements inconditionnels.

## REMERCIEMENTS

---

Je tiens tout d'abord à remercier Dieu Tout Puissant, pour nous avoir accordé la vie et d'avoir permis que tout ceci soit possible.

Je tiens à remercier toutes les personnes qui ont contribué au succès de mon stage et qui m'ont aidé lors de la rédaction de ce mémoire.

Je tiens à remercier :

- M. AGBETI Kodjo, Représentant Résident de l'IAI-TOGO, pour tous les efforts fournis pour nous offrir une formation de qualité d'études ainsi que des études de qualité dans de bonnes conditions.
- M. AMEYIKPO Nicolas, Directeur des Affaires Académiques et de la Scolarité de l'IAI-TOGO pour son encadrement académique.
- toute l'équipe pédagogique de IAI-Togo.
- mon superviseur M. DJAGRE Kofi Napo, pour sa patience, sa disponibilité et surtout ses judicieux conseils, qui ont contribué à alimenter ma réflexion.
- M. KABATE M'Ponada Paul, mon Maître de stage, pour son accueil, son soutien et sa disponibilité malgré ses journées très chargées ;
- NADOR Ekuevi Mike Edem, notre Directeur Technique, pour son soutien et son implication dans la réalisation de ce projet.
- tous les membres du département informatique de Digital Deep Vision pour leur accueil et leur convivialité ;

Je tiens à témoigner toute ma reconnaissance aux personnes suivantes, pour leur aide dans la réalisation de ce mémoire :

- mes parents, mes sœurs, mes tantes pour leur soutien constant, leurs encouragements et leurs prières.
- mes amis qui ont toujours été là pour moi. Leur soutien inconditionnel et leurs encouragements ont été d'une grande utilité.

À tous ces intervenants, je présente mes remerciements, mon respect et ma gratitude.

# SOMMAIRE

DEDICACES .....	I
REMERCIEMENTS .....	II
RESUME .....	V
GLOSSAIRE .....	VI
LISTE DES FIGURES .....	VIII
LISTE DES TABLEAUX .....	X
LISTE DES PARTICIPANTS AU PROJET .....	XI
INTRODUCTION .....	1
PARTIE 1: PRESENTATION .....	2
1.1. BREVE PRESENTATION DE L'IAI-TOGO .....	3
1.2. PRESENTATION DU CADRE DE STAGE .....	6
PARTIE 2: ETUDE REALISATION DU PROJET .....	8
CHAPITRE 1: CONTEXTE DE TRAVAIL ET APPROCHE DE SOLUTIONS .....	9
1.1. ÉTATS DES LIEUX .....	10
1.2. CRITIQUE DE L'EXISTANT .....	11
1.3. PROBLEMATIQUE .....	12
1.4. INTERET DU SUJET .....	12
1.5. METHODOLOGIE DE RECHERCHE DE SOLUTION .....	13
CHAPITRE 2: GENERALITES ET DOCUMENTATION .....	14
2.1. SOLUTION DE GESTION D'HABITAT .....	15
2.2. LE SERVEUR (LA BOX) .....	19
2.3. LES MICROCONTROLEURS SUR CARTE .....	22
2.4. LES PROTOCOLES DE COMMUNICATION .....	29
2.5. SYNTHÈSE ET CHOIX DES ÉLÉMENTS DE RÉALISATION .....	32
CHAPITRE 3: MISE EN ŒUVRE ET PERSPECTIVES .....	34
3.1. LES ÉLÉMENTS DE RÉALISATION .....	35
3.2. MISE EN ŒUVRE .....	43

<b>3.3. TEST DE FONCTIONNEMENT .....</b>	<b>54</b>
<b>3.4. EVALUATION FINANCIERE .....</b>	<b>56</b>
<b>PARTIE 3: GUIDE D'UTILISATION .....</b>	<b>60</b>
<b>3.1. GUIDE D'UTILISATION POUR LES CLIENTS .....</b>	<b>61</b>
<b>CONCLUSION .....</b>	<b>64</b>
<b>BIBLIOGRAPHIE INDICATIVE.....</b>	<b>65</b>
<b>ANCIENS MEMOIRES CONSULTES .....</b>	<b>66</b>
<b>NOTES DE COURS .....</b>	<b>67</b>
<b>WEBOGRAPHIE INDICATIVE.....</b>	<b>ERREUR ! SIGNET NON DEFINI.</b>
<b>TABLE DES MATIERES.....</b>	<b>70</b>

## RESUME

---

Dans la perspective d'obtenir notre diplôme de Licence Professionnelle en Informatique, nous avons été amené à effectuer un stage de trois (03) mois dans la structure de DD VISON (Digital Deep Vision). Il nous a alors été soumis le thème « Conception et implémentation d'un système de contrôle informatique de la consommation d'énergie électrique d'un habitat ». Le but visé par ce projet est de concevoir et déployer un système nous permettant de contrôler la consommation d'un habitat, ce qui aiderai les habitats (ou entreprise) à gérer au mieux sa consommation énergétique, de la manière la plus sécurisée possible. Pour la réalisation de ce projet, une étape de modélisation de l'architecture système et de l'architecture réseau nous a permis de maitriser la complexité du système.

Le pare-feu iptables, le serveur web NodeJS, le protocole réseau web Socket, le Serveur applicatif Flask, le serveur de bases de données MongoDB nous ont permis de réaliser ce projet.

## GLOSSAIRE

### B

bus de terrain

Le terme bus de terrain est utilisé par opposition au bus informatique. ....27

### C

CPL

courants porteurs en ligne (ou CPL) permet de construire un réseau informatique sur le réseau électrique d'une habitation .... ix, 27, 29, 30

### D

DHCP

Dynamic Host Configuration Protocol ..9, 10, 35, 48, 49

### F

FAI

fournisseur d'accès à Internet .....9

### G

GPIO

Les ports GPIO (anglais General Purpose Input/Output, littéralement Entrée-sortie à usage général) ..... 17, 20, 22, 23

### H

HTML5

HyperText Markup Language » qu'on peut traduire par « langage de balises pour l'hypertexte.....15

### L

Logiciel ETS

ETS6 est un logiciel de paramétrage

indépendant dédié à accroître l'efficacité énergétique des bâtiments avec le protocole KNX. .... 27

LPDDR4

La LPDDR4 (en anglais

« Low Power Double Data Rate 4 », littéralemen

« débit de données double basse consommation 4») est un format de mémoire pour périphérique basse consommation ..... 19

LTS

signifie Long Term Support, ou support à long terme. .... 33, 45

### P

PCI-E

Le PCI Express est un bus local série et un connecteur qui sert à connecter des cartes d'extension sur la carte mère d'un ordinateur. .... 18, 20

### S

SBC

Un ordinateur monocarte ( Single Board Computer ) est un ordinateur complet construit sur une seule carte de circuit imprimé..... 17, 18, 20

SDK

est un ensemble d'outils logiciels destinés aux développeurs, facilitant le développement d'un logiciel sur une plateforme donnée. .... 21

SoC

« system on a chip », est un système  
complet embarqué sur un seul circuit  
intégré .....22

**T**

TCP/IP

La suite des protocoles Internet est  
l'ensemble des protocoles utilisés pour le  
transfert des données sur Internet ..... 21

triphasée

Un système de courant triphasé est  
constitué de trois courants sinusoïdaux  
de même fréquence et de même  
amplitude ..... 28



## LISTE DES FIGURES

FIGURE 1: ORGANIGRAMME DE L'IAI-TOGO.....	4
FIGURE 2: SITUATION GEOGRAPHIQUE DE L'INSTITUT .....	5
FIGURE 3: ORGANIGRAMME DE DIGITAL DEEP VISION .....	7
FIGURE 4: PLAN DE LOCALISATION DE LA STRUCTURE D'ACCUEIL .....	7
FIGURE 5: ARCHITECTURE RESEAU TYPE CLIENT OU CIBLE.....	11
FIGURE 6: LOGO DE L'ASUS THINKERBOARD.....	20
FIGURE 7: LOGO DE LA RASPBERRY PI .....	20
FIGURE 8: ESP8266 NODEMCU.....	24
FIGURE 9: ESP32 LOLIN32 .....	25
FIGURE 10: ARDUINO UNO .....	26
FIGURE 11: ARDUINO MEGA 2560 .....	27
FIGURE 12: KNX .....	29
FIGURE 13: Z-WAVE .....	30
FIGURE 14: LE WI-FI.....	31
FIGURE 15: LOGO DE UBUNTU .....	36
FIGURE 16: LOGO DNSMASQ .....	37
FIGURE 17: LOGO NODEJS .....	38
FIGURE 18: LOGO NPM .....	39
FIGURE 19: LOGO PYTHON.....	39
FIGURE 20: LOGO FLASK.....	40
FIGURE 21: LOGO MONGODB.....	40
FIGURE 22: CAPTEUR DE COURANT ACS712 DE AZDELIVERY .....	41
FIGURE 23: CAPTEUR DE PRESENCE HC-SR501 .....	42
FIGURE 24: DHT22 .....	43
FIGURE 25: RASPBERRY PI IMAGER TERMINANT L'ECRIURE DU MICROSD.....	44
FIGURE 26: CONFIGURATION DE L'INTERFACE ETH0 .....	45
FIGURE 27: FICHIER DE CONFIGURATION 01-NETWORK-MANAGER-ALL.YAML .....	46
FIGURE 28: SCHEMA ILLUSTRANT LES ECHANGES WEBSOCKET.....	49
FIGURE 29: DATAGRAMME D'ECHANGES DE DONNEE. ....	49
FIGURE 31: LOGO D'ARDUINO .....	52
FIGURE 33: CONFIGURATION DU FICHER UNITE .....	53
FIGURE 34: CONNEXION WEB SOCKET ENTRE LE CLIENT POSTMAN ET LE SERVEUR APPLICATIVE .....	55
FIGURE 35: SERVEUR APPLICATIVE .....	55
FIGURE 36: SERVEUR WEB SOCKET.....	55
FIGURE 37: CONSOLE PORT SERIE DU MCU .....	56
FIGURE 38: MESSAGE DES CAPTEURS DANS CONSOLE PORT SERIE DU MCU .....	56
FIGURE 39: PAGE D'AUTHENTIFICATION.....	61

FIGURE 40: BARRE DE NAVIGATION DE L'APPLICATION.....	62
FIGURE 41: BOUTON D'AJOUT D'UNE PIECE.....	62
FIGURE 42: PAGE D'AJOUT D'UNE PIECE .....	63

## LISTE DES TABLEAUX

---

TABLEAU 1 : LISTE DES PARTICIPANTS AU PROJET .....	XI
TABLEAU 2: SYNTHESE DES SERVICES DE GESTION. ....	18
TABLEAU 3: UNE COMPARAISON DES ORDINATEUR MONOCARTE (RASPBerry PI, ASUS THINKERBOARD) .....	21
TABLEAU 4: SYNTHESE SUR LES MICROCONTROLEURS SUR CARTE .....	28
TABLEAU 5: COMPARATIF WI-FI ET CPL .....	32
TABLEAU 6: RECAPITULATIF DES TECHNOLOGIES ET OUTIL RETENUS. ....	33
TABLEAU 7: TABLEAU DES COUTS LIES A LA MISE EN PLACE DE LA SOLUTION .....	57
TABLEAU 8: TABLEAU DES COUTS LIES A L'ACQUISITION DU MATERIELS .....	57
TABLEAU 9: COUTS TOTAUX DE L'IMPLEMENTATION DU SYSTEME .....	59

## LISTE DES PARTICIPANTS AU PROJET

Tableau 1 : liste des participants au projet

Nom et prénom(s)	Fonction	Rôle
M. DJAGRE Kofi Napo	Enseignant à IAI-TOGO	Superviseur
M. KABATE M'Ponada Paul	Directeur Digital Deep Vision	Maître de stage
GONCALVES Koku Jean Hilaire	Etudiant en troisième année Option Administration des Systèmes et Réseaux à l'IAI-TOGO	Réalisateur

# INTRODUCTION

---

Nos milieux de vie sont remplis de composants électroniques. Ces composants ont la spécificité de consommer de l'énergie qui est ressentie en fin de mois par chaque utilisateur, une fois la facture arrivée. Les habitats sont confrontés à des défis en ce qui concerne leurs consommations énergétiques. Ce défi est dû à certaine habitude des utilisateurs à faire tourner des appareils sans qu'ils ne soient utilisés ou qu'il n'y ait personne dans la pièce.

C'est dans cette optique que Digital Deep Vision a voulu développer une solution pour pallier ce problème. Il nous a alors confié la conception et l'implémentation d'une architecture pouvant répondre à ces besoins, d'où le thème « Conception et implémentation d'un système de contrôle informatique de la consommation d'énergie d'un habitat ».

Un habitat est un milieu géographique propre à la vie. Cela peut être une maison, un bureau, un immeuble, bref le milieu de vie d'un individu. Et un système de contrôle informatique de la consommation d'énergie, est l'ensemble des matériels et logiciels qui nous permettront de centraliser le contrôle de la plupart des appareils. Le système nous permettra de suivre la consommation énergétique de chaque appareil. L'automatisation de ce système permet à un habitat de s'adapter à certaines conditions dans le but de faciliter la vie à l'occupant, et surtout de l'aider à optimiser sa consommation.

Il nous incombe donc de concevoir un système informatique permettant à ces différents équipements (appareils) de communiquer, c'est-à-dire recevoir des informations du Serveur et les traiter puis y répondre ; d'implémenter les différentes solutions qui nous permettront d'atteindre nos objectifs et de pouvoir sécuriser l'installation.

Dans ce document, nous présenterons tout d'abord l'IAI-Togo et l'entreprise Digital Deep Vision, ensuite nous ferons une étude de l'existant et de la problématique qui débouchera sur une phase de recherche et du choix de solution, puis de la mise en œuvre de la solution

# **PARTIE 1: PRESENTATION**

## I.I. BREVE PRESENTATION DE L'IAI-TOGO

---

L'IAI-TOGO est une école inter-Etats d'enseignement supérieur en informatique. Il est membre du réseau créé le 29 janvier 1971 à Fort Lamy (actuel N'Djaména) en République du TCHAD.

En application de la décision du Conseil d'Administration de délocaliser l'IAI, la Représentation du Togo (IAI-TOGO) a ouvert ses portes le 24 octobre 2002. L'accord d'établissement entre la République Togolaise et l'Institut Africain d'Informatique a été signé le 12 mai 2006.

L'IAI-TOGO propose trois (03) filières à savoir le Génie Logiciel et Systèmes d'Information (GLSI), l'Administration des Systèmes et Réseaux (ASR) puis le Multimédia, Technologie Web et Infographie (M-TWI).

L'IAI-TOGO propose actuellement le cycle de formation des ingénieurs de travaux informatiques (Licence professionnelle en informatique).

L'Institut Africain d'Informatique, représentation du Togo est doté d'une structure organisationnelle de type hiérarchique et comprend :

- la Direction Générale, à la tête de laquelle se trouve le Représentant Résident.

Cette direction est composée de plusieurs cellules et services à savoir :

- un Secrétariat Central ;
- un Secrétariat Particulier ;
- une Cellule de Contrôle interne ;
- une Cellule Communication et Relations Extérieures ;
- la Direction des Affaires Académiques et de la Scolarité (DAAS) ;
- la Direction Administrative et Financière (DAF).

Voici, ci-après l'organigramme de l'IAI-TOGO (Figure 1) :

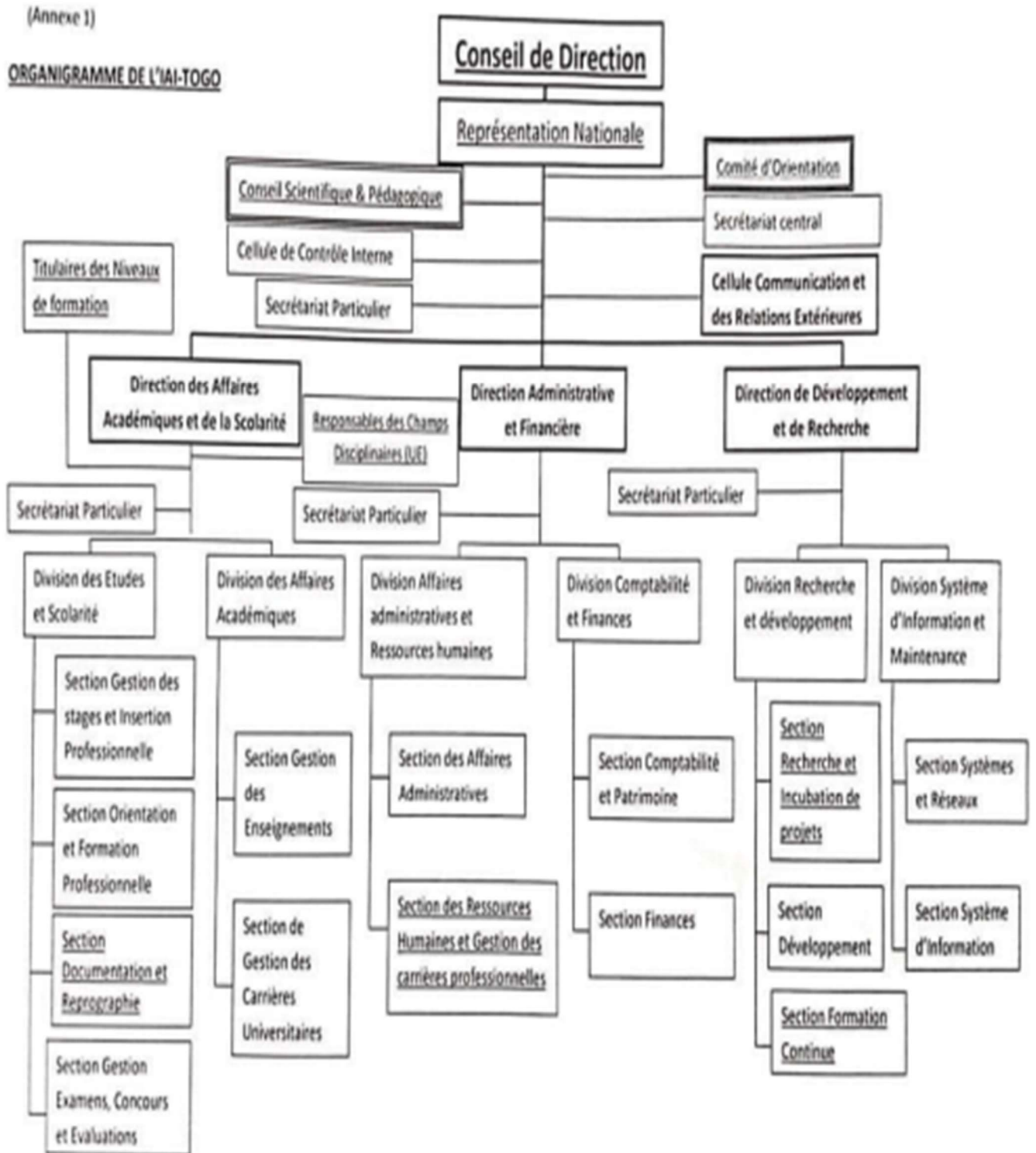
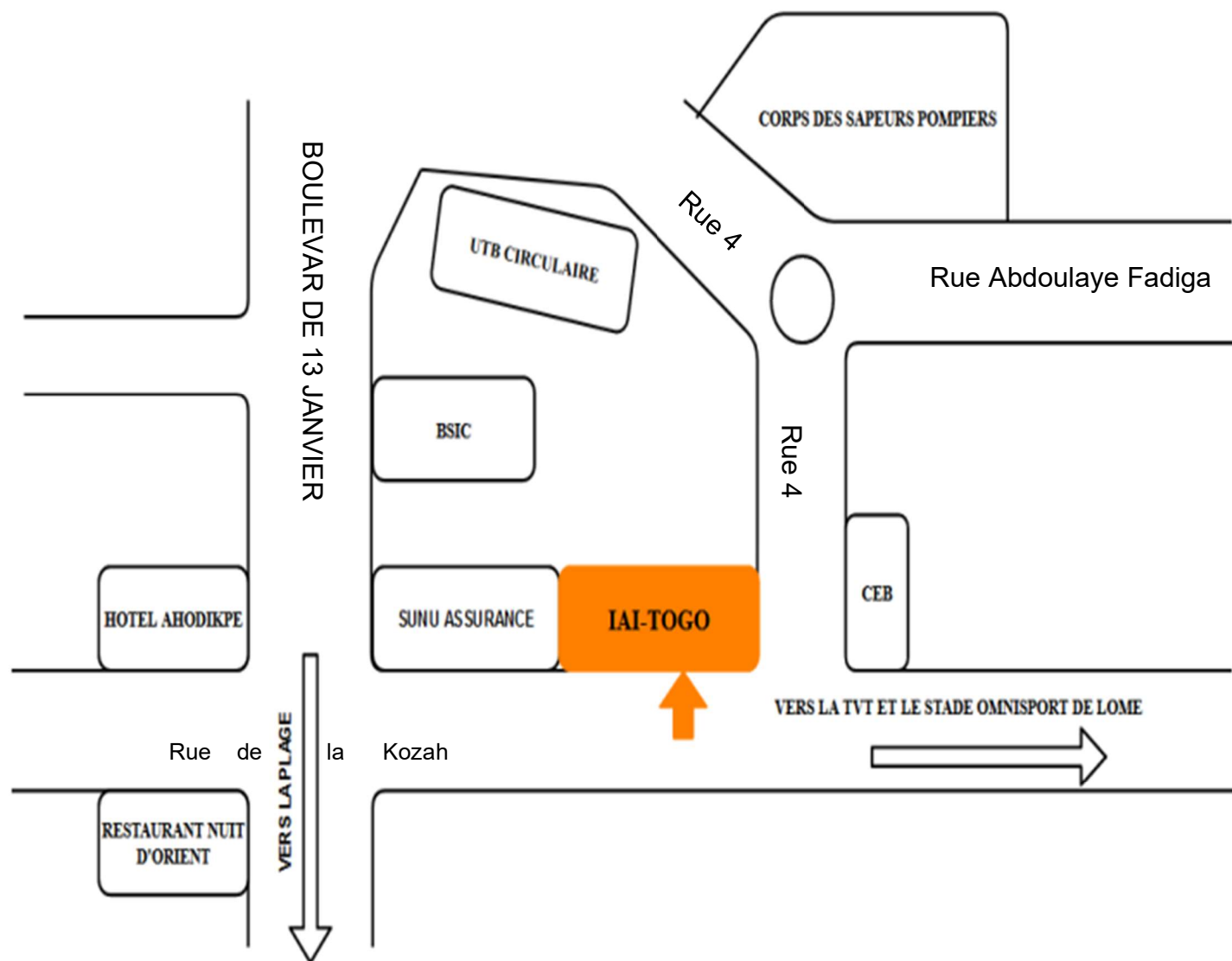


Figure 1: Organigramme de l'IAI-TOGO

IAI-TOGO se situe dans le quartier administratif, dans la rue de la Kozah, derrière SUNU assurance, dans le bâtiment du Centre National d'Etudes et de Traitements Informatiques (CENETI). La figure 2 illustre le plan de localisation de l'IAI-TOGO :





*Figure 2: Situation géographique de l'institut*

## I.2. PRESENTATION DU CADRE DE STAGE

---

### I.2.1. Statut

La société Digital Deep Vision (DIGITAL DEEP VISION SARL) est une société à responsabilité limitée siégeant à Lomé depuis 2022. Elle est située à Agoè Anomé, à environ 200m du collège « Les éléphants ».

### I.2.2. Mission

Digital Deep Vision s'est fixé plusieurs missions dont les plus importantes sont :

- **la conception et réalisation des systèmes électroniques :**

DIGITAL DEEP VISION réalise des systèmes électroniques sur mesure comme des systèmes domotiques, de détecteurs de présence, de gestion d'énergie etc...

- **le développement d'applications desktop, mobile et web :**

DIGITAL DEEP VISION dispose d'une équipe de développeurs dotés de compétences nécessaires pour développer tout type de logiciels et applications.

- **l'architecture, installation réseau et TV :**

DIGITAL DEEP VISION dispose d'architectures d'installations réseaux et TV selon les besoins des clients après une étude des lieux.

- **le conseil et le suivi informatique :**

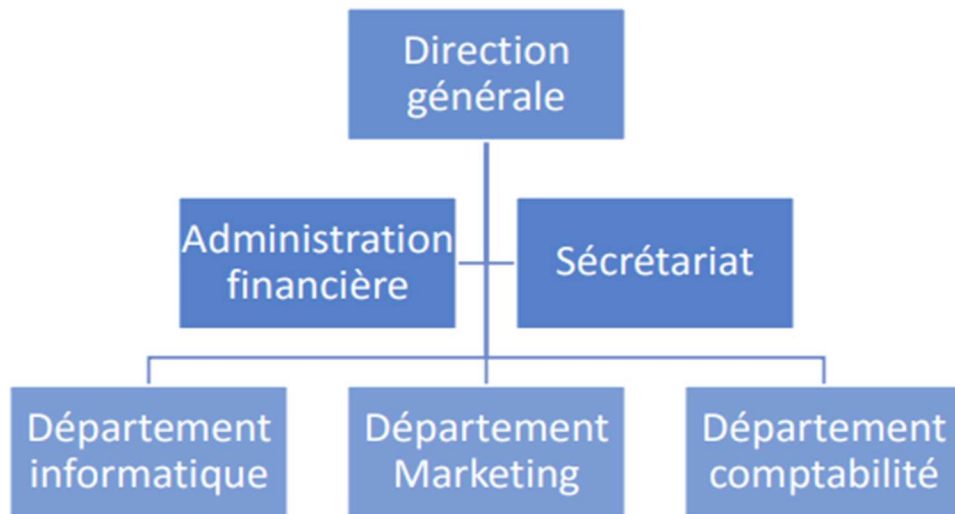
La société DIGITAL DEEP VISION ; fournit, par le moyen de ses experts, des conseils et le suivi dans les domaines informatique et électronique.

### I.2.3. Organigramme

Digital Deep Vision dispose principalement de trois (3) départements :

- le département informatique ;
- le département marketing ;
- le département comptabilité.

Ces divers départements sont coordonnés par la direction générale assistée par le secrétariat.

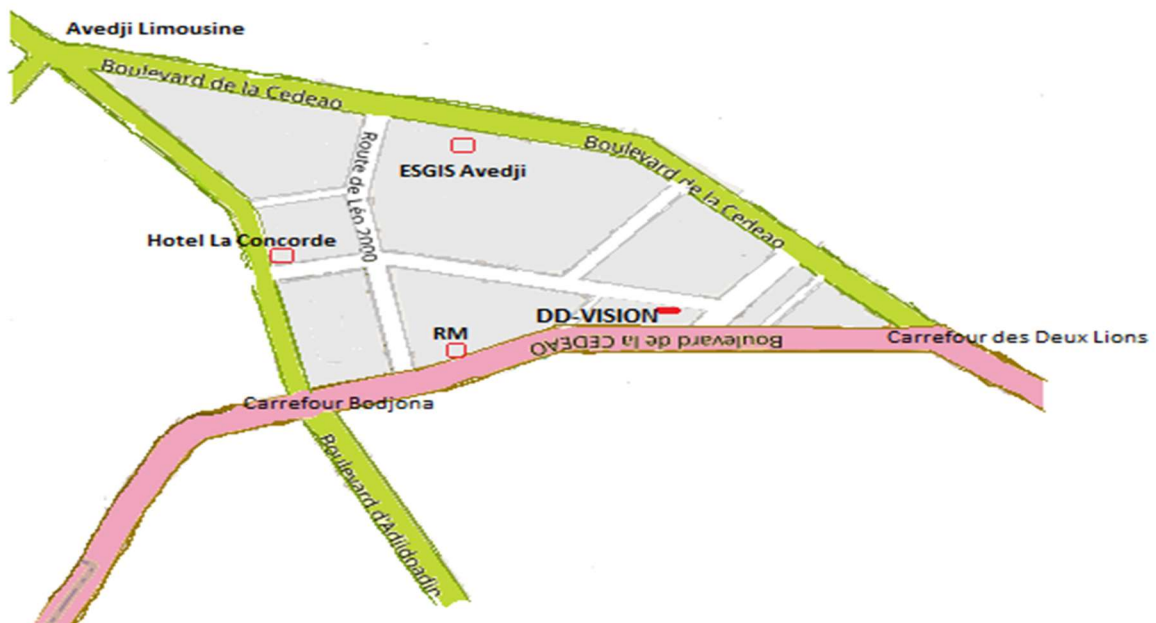


*Figure 3: Organigramme de Digital Deep Vision*

## I.2.4. Service d'accueil

En ce qui concerne la réalisation de notre stage de fin de formation, nous avons intégré le département informatique de la Société Digital Deep Vision.

## I.2.5. Plan de localisation



*Figure 4: plan de localisation de la structure d'accueil*

## **PARTIE 2:     ETUDE REALISATION DU PROJET**

# **CHAPITRE 1: CONTEXTE DE TRAVAIL ET APPROCHE DE SOLUTIONS**

Dans cette section, nous ferons une présentation de l'architecture système et physique de notre environnement de test qui est un petit ménage de deux chambre salon plus une cuisine.

Notre architecture fera état de la structure générale inhérente aux systèmes informatiques des habitations togolaises, l'organisation des différents éléments du système (logiciels et/ou matériels et/ou humains et/ou informations) et des relations entre les éléments.

En fonction de notre existant, nous allons suivre une méthodologie de recherche qui nous permettra d'opter pour la solution la plus fiable, afin répondre aux différentes attentes.

## I.I.ÉTATS DES LIEUX

---

### I.I.I. Architecture système

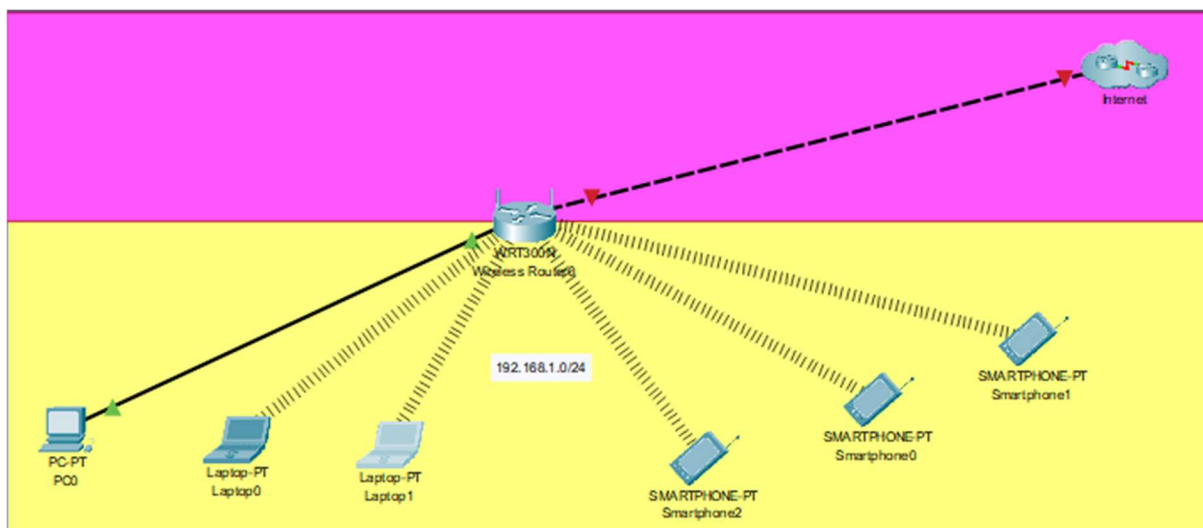
Il est à noter que l'architecture est la même ou presque dans la plupart des habitats, donc dans l'architecture système de nos habitats nous avons :

- un routeur installé par le fournisseur d'accès Internet (TOGOCOM ou Group Vivendi Africa) qui est souvent de marque Nokia ; elle fournit les services de base tels que la résolution de nom, et le DHCP sur une plage de 192.168.1.64 à 192.168.1.150,
- des smartphones de marque différentes et variées tournant sur IOS ou Android,
- des ordinateurs portables et de bureau sous Windows ou sur Mac OS.

### I.I.2. Architecture réseau (topologie physique et topologie logique)

Le routeur installé par le fournisseur accès internet dispose d'un réseau WLAN utilisant la technologie wifi norme 802.11g. L'interconnexion de ses équipements se fait selon le mode infrastructure. Ce mode est un réseau sans fil qui possède un point d'accès centralisé au cœur du réseau. Ainsi, dans ce réseau, les smartphones et ordinateurs

communiquent entre eux et sont connectés à internet avec un débit de 10 - 200 Mbits par le biais du modem routeur fibre optique fourni par le FAI, qui est l'équipement central (point d'accès). Les paramètres réseaux sont attribués dynamiquement aux machines via la fonction DHCP activée sur le modem routeur. Les adresses IP sont attribuées suivant la plage d'adresse 192.168.1.64 à 192.168.1.150.



*Figure 5: Architecture Réseau type client ou cible*

## I.2. CRITIQUE DE L'EXISTANT

Concernant la critique de l'existant, nous allons relever les points fort du réseau type client ou cible, puis nous allons relever des insuffisances de ce même réseau.

### I.2.1. Points forts

- un bon débit, l'accès internet est de meilleure qualité,
- le réseau permet l'interconnexion effective de chaque appareil connecté,
- le circuit électrique préexistante.

### I.2.2. Points faibles

Le réseau type clients représenté ici ne présente aucun handicap majeur à la conception de notre système. Toutefois, certaines absences ou manquement peuvent être notés, tels que :

- une absence d'un routeur,
- un routeur de FAI, peu manageable,
- un système de courant (ou tension) triphasé,
- une consommation inutile de l'énergie électrique,
- aucun suivi de la consommation

## I.3. PROBLEMATIQUE

---

Dans l'étude de l'existant, nous avons constaté des insuffisances. Pour aider les particuliers togolais à optimiser facilement leur consommation en énergie électrique, c'est-à-dire éviter les consommations inutiles et suivre sa consommation, l'entreprise Digital Deep Vision nous a confié la réalisation d'un système qui permettra à ses utilisateurs de mieux contrôler leur consommation énergétique.

Au vu de ce qui précède, comment pouvons-nous, grâce à nos compétences développées durant notre cursus, aider au mieux à optimiser la consommation énergétique ?

Notre système pourrait-il :

- connaître la consommation énergétique d'un ou de plusieurs appareil(s) ?
- savoir si ces derniers sont éteints ou allumés ?
- savoir si une pièce est occupée ou pas ?
- communiquer (Envoyer et recevoir) ces données aux différents appareils ?

## I.4. INTERET DU SUJET

---

### I.4.1. Objectifs

Comme pour toute étude, des objectifs doivent être atteints, afin de prétendre avoir rempli la tâche donnée. Dans notre cas, les objectifs attendus sont de :

- savoir l'état (allumée ou éteint) et le communiquer au serveur.



- connaître et communiquer à tout moment (en temps réel) la consommation de chaque équipement,
- recevoir et exécuter des instructions du serveur.

### **I.4.2. Résultats attendus**

Après la réalisation de ce projet, nous aurons un système dont :

- l'état d'un l'appareil (allumé ou éteint), est connu ;
- l'état de la consommation actuelle d'un appareil, est connu ;
- l'état d'une pièce est connu.

## **I.5. METHODOLOGIE DE RECHERCHE DE SOLUTION**

---

Afin de proposer une meilleure solution qui répondra au mieux aux attentes de DD-VISION nous ferons :

- une recherche documentaire basée sur les généralités des différentes solutions en termes de gestion d'habitat libre et gratuite (Home Assistant, Domoticz) ;
- une recherche documentaire sur les différents microcontrôleurs sur carte ;
- une recherche documentaire sur les Protocoles de communications filaires et non filaires pour utilisation de microcontrôleurs sur carte : une recherche documentaire sur les différents capteurs adaptés,
- une recherche documentaire sur les différents moyens possibles pour la sécurisation de tout le système ;
- une synthèse des solutions retenues pour la mise en place de la sécurisation du système.

## **CHAPITRE 2: GENERALITES ET DOCUMENTATION**

Dans ce deuxième chapitre, il sera question de faire une étude des différents technologies et outils qui peuvent nous aider à atteindre nos objectifs, notamment allumer et éteindre les appareils grâce au réseau, connaître leurs consommations et savoir s'ils sont allumés ou éteints.

## 2.I. SOLUTION DE GESTION D'HABITAT

---

Une solution de gestion d'habitat est un logiciel qui permet de programmer, de contrôler, ou configurer des appareils. Les logiciels de gestion peuvent aussi être des applications Android qui communiquent avec votre Box domotique via Bluetooth, Wifi, Z-wave ou autres protocoles sans fils.

### 2.I.I. Home Assistant

Home Assistant <sup>1</sup> est un logiciel gratuit et open-source conçue pour être un système de contrôle central pour les appareils domestiques intelligents avec un accent sur le contrôle local et la confidentialité. Il est accessible via une interface utilisateur basée sur le Web à l'aide d'applications complémentaires pour Android et iOS, ou par des commandes vocales via un assistant virtuel pris en charge tel que Google Assistant, Amazon Alexa ou même "Génie" (anciennement connu sous le nom de "Almond") par Stanford Open Virtual Assistant Lab. (OVAL.)

Les informations de tous les appareils et de leurs attributs (entités) qu'ils voient peuvent être utilisées et contrôlées à partir de scripts qui déclenchent des automatisations à l'aide de sous - programmes de planification et de « plan », par exemple pour contrôler l'éclairage, le climat, les systèmes de divertissement et les appareils ménagers.

---

<sup>1</sup> <https://www.home-assistant.io/>

- Avantages
  - Facilité d'implémentation,
  - Découverte automatique des appareils,
  - Facilité d'ajout de nouvelles fonctionnalités.
- Inconvénients : Home Assistant n'a aucun inconvénient.

## 2.1.2. Domoticz

Domoticz<sup>2</sup> est gratuit et open source. Il peut être utilisé sur Raspberry Pi, Windows, Linux, Mac OS X et les appareils embarqués. Le système est conçu pour fonctionner dans divers systèmes d'exploitation. L'interface utilisateur est une interface Web HTML5 évolutive et est automatiquement adaptée aux ordinateurs de bureau et aux appareils mobiles. Domoticz est compatible avec tous les navigateurs.

Remarque pour les utilisateurs d'Internet Explorer : Vous avez besoin de la version 10+.

La première version de Domoticz date de décembre 2012.

- Avantages
  - Léger et consomme peu de ressource système,
  - Offrent un nombre conséquent de fonctionnalités,
  - Interface web disponible pour la configuration.
- Inconvénients
  - Documentation moins exhaustive,
  - Nombre d'appareils pris en charge non détaillé.
  - Le langage pour les automatisations est rigide et peu connus

## 2.1.3. Ago AHome

---

<sup>2</sup> <https://www.domoticz.com/>

Ago AHome est la solution développée par Digital Deep Vision. Elle est pour le moment une application mobile qui tourne sur Android, le but est d'aider les utilisateurs à contrôler et à suivre la consommation énergétique de chacun des appareils de son habitat dans un optique de pouvoir optimiser cette consommation.

- Avantages
  - cible particulièrement la consommation énergétique des appareils.
  - facile d'utilisation,
  - adapté aux réalités togolaises.
- Inconvénients
  - disponible que sur Android
  - pas gratuit

## 2.1.4. Choix de la solution de gestion d'habitat

Au vu de ce qui précède, et pour une synthèse optimale des services étudiés, nous allons faire un comparatif sur cinq (05) grands critères, notamment la flexibilité, la communauté et la vitesse de développement, l'automatisation des tâches puis du coût d'acquisition.

Home Assistant étant très flexible, il pourra couvrir la plupart des besoins de ces utilisateurs. En revanche, Domoticz, avec une interface peu évolutive, l'est moins.

La communauté de Home assistant est probablement la plus grande ; cependant, Domoticz a moins d'utilisateur.

La vitesse de développement de Home assistant étant trop rapide. ils font une mise à jour toutes les deux semaines. Domoticz, lui, est à la traine en terme mise à jour, donc peu de protocole et d'appareil pris en charge.

Le coût des deux premières solutions pour la version premium est sur demande, pour Ago AHome, étant en développement, les tarifications n'ont pas encore été fixées.

Avec Home Assistant, vous pouvez configurer les automatisations depuis l'interface web, un fichier yaml ou avec python, Domoticz lui, utilise le Scripting LUA très propre et puissant. Le tableau suivant nous donne une comparaison entre les différentes solutions de gestion de l'habitat :

Tableau 2: synthèse des services de gestion.

N°	Critères	Home Assistant	Domoticz	Ago AHome
1	Flexibilité	Très	Moyens	Très
2	Communauté	Très grande	faible	Faible
3	Développement	Trop rapide	Lente	Moyen
4	Automatisation	Très avancé	Très avancé	simple
5	Coût	Sous demande	Sous demande	Non déterminé

Home Assistant et Domoticz sont des solutions de gestion d'habitat remplissant correctement leurs rôles. De cette comparaison, home Assistant a plus d'avantage que Domoticz, mais tous deux dans la gestion de la consommation sont assez limités et leur écosystème est moins adapté à nos réalités. C'est pour cela DIGITAL DEEP VISION a décidé de développer son propre système plus axé sur la gestion de la consommation énergétique d'un habitat, et notre rôle sera de préparer l'environnement de déploiement, de concevoir l'architecture réseau et système, puis de déployer la solution développée par DIGITAL DEEP VISION.

## 2.2. LE SERVEUR (LA BOX)

---

L'objectif du système est d'aider l'utilisateur à optimiser sa consommation énergétique. Or le serveur (la box) doit tourner 24h/24 et 7j/7, et la plupart des matériels serveur, sont énergivores, donc il nous faut trouver une solution économique, et assez puissante pour faire tourner aisément tous les services déployés.

Un ordinateur monocarpe « Single Board Computer » (SBC), comme son nom l'indique, est un ordinateur qui se compose d'une seule carte. Un SBC est un ordinateur à part entière avec de la RAM, du stockage, des entrées/sorties et des périphériques comme les USB, tous sur une seule carte mère (Smartphones, Raspberry pi, Asus Thinkerboard...).

### 2.2.1. ASUS Thinker Board

L'ASUS Tinker Board est un ordinateur monocarte lancé par ASUS au début de 2017. Sa taille physique et son brochage GPIO sont conçus pour être compatibles avec les modèles Raspberry Pi de deuxième et troisième génération. La première carte publiée comprend 2 Go de RAM intégrée, Gigabit Ethernet et un processeur Rockchip RK3288 fonctionnant à 1,8 GHz.



*Figure 6: logo de l'Asus Thinkerboard*

- Avantage
  - Un port PCI-E de libre
  - Antenne wifi m2
- Inconvénient
  - Architecture du processeur
  - Chère par rapport au pi.

## 2.2.2. Raspberry pi

Raspberry Pi est une série de petits ordinateurs monocarte (SBC) développés au Royaume-Uni par la Raspberry Pi Foundation en association avec Broadcom. Le modèle original est devenu plus populaire que prévu, vendu en dehors de son marché cible pour des utilisations telles que la robotique. Il est largement utilisé dans de nombreux domaines, tels que la surveillance météorologique en raison de son faible coût, de sa modularité et de sa conception ouverte. Il est généralement utilisé par les amateurs d'informatique et d'électronique, en raison de son adoption des normes HDMI et USB.



*Figure 7: logo de la Raspberry pi*

- Avantage
  - Compact ;
  - Consomme très peu ;
  - Forte communauté.



- Inconvénient
  - Architecture du processeur ;
  - Surchauffe.

### 2.2.3. Choix de l'ordinateurs monocarte

Le Pi et le Tinker Board comportent des mêmes composants, mais ceux-ci diffèrent par leurs spécifications, leurs types et leurs nombres. À des fins de comparaison, nous prendrons la version la plus récente de ces deux fabricants, Raspberry Pi 4B et Asus Tinker Board 2S. Vous trouverez ci-dessous une comparaison de leurs composants matériels.

*Tableau 3: une comparaison des ordinateur monocarte (Raspberry pi, Asus Thinkerboard)*

N°	Critères	Raspberry pi	ASUS Thinkerboard 2
1	CPU	Quad Core 1,5 GHz Cortex-A72	double cœur 2,0 GHz A-72
2	RAM	1/2/4/8 Go LPDDR4	2/4 Go LPDDR4
3	Stockage	- Fente pour carte micro SD	- 16 Go eMMC - Fente pour carte micro SD
4	Connectivité	- gigabit Ethernet - Sans fil 802.11b/g/n/ac - Bluetooth 5.0	- Gigabit Ethernet - M.2 802.11 a/b/g/n/ac sans fil - Bluetooth 5.0
5	USB	- 2 ports USB 3.0 de type A - 2 ports USB 2.0	- 3 ports USB 3.2 Gen1 Type-A - 1 port USB 3.2 Gen1 Type-C OTG

<b>6</b>	Alimentation	<ul style="list-style-type: none"> <li>- 5V CC via le connecteur USB-C</li> <li>- Alimentation par Ethernet (PoE HAT)<sup>3</sup></li> </ul>	<ul style="list-style-type: none"> <li>- Prise d'entrée d'alimentation 12 ~ 19 V CC (5,5/2,5 mm)</li> <li>- 5V DC via l'en-tête GPIO<sup>4</sup></li> </ul>
<b>7</b>	Coût	130 000 <sup>5</sup>	95 000 <sup>6</sup>

Au vu de cette comparaison nous remarquons que les différents critères soumis à étude rempliront parfaitement nos attentes avec les puissances nécessaires. La tinker Board avec son PCI-E de libre et une antenne M2 serait le parfait serveur, mais compte tenu de sa consommation, et que le Raspberry pi remplit tout aussi les exigences tout en consommant moins, nous avons opté pour un pi 4, 4Gb de ram et 32GB de carte SD.

## 2.3. LES MICROCONTROLEURS SUR CARTE

Un microcontrôleur a beaucoup en commun avec un SBC. Les deux ont un processeur, un stockage et des périphériques d'entrée/sortie. Mais c'est là où s'arrêtent les similitudes.

<sup>3</sup> <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/>

<sup>4</sup> <https://www.asus.com/fr/Networking-IoT-Servers/AIoT-Industrial-Solutions/All-series/Tinker-Board/>

<sup>5</sup> <https://www.amazon.fr/Raspberry-ARM-Cortex-A72-WLAN-AC-Bluetooth-Micro-HDMI>

<sup>6</sup> <https://www.es-france.com/17431-carte-mere-de-developpement-avec-dual-core-arm-tinker-board>

Les microcontrôleurs ont beaucoup moins de ressources que les SBC. Par exemple, un microcontrôleur peut avoir des Ko de stockage au lieu de Mo ou de Go de SBC. De même, leurs capacités de traitement sont également assez limitées.

Dans la plupart des cas, un microcontrôleur a également besoin de cartes de dérivation pour faciliter sa programmation. Un Arduino UNO est un bon exemple de microcontrôleur avec une carte de dérivation. Il nous permet de connecter le microcontrôleur, dans ce cas, un Atmel ATmega328P, avec périphériques d'entrée/sortie, alimentation et périphériques.

Les microcontrôleurs ne peuvent pas non plus exécuter plus d'un programme à la fois, car ils sont destinés à boucler une routine.

## 2.3.1. Les ESP

### 2.3.1.1. ESP8266

L'ESP8266 est un circuit intégré à microcontrôleur avec connexion Wi-Fi développé par le fabricant chinois Espressif.

La version ESP-01 de ce circuit intégré est produite par une entreprise tierce, Ai-Thinker. Celui-ci, de taille réduite permet de connecter un microcontrôleur à un réseau Wi-Fi et d'établir des connexions TCP/IP avec des commandes Hayes

Plus tard, Espressif propose un kit de développement logiciel (SDK) permettant de programmer le circuit sans recourir à l'utilisation d'un microcontrôleur additionnel. Depuis, Espressif a sorti une quantité de nouvelles versions du SDK et ce sous deux variantes : une basée sur FreeRTOS3 et une autre basée sur les fonctions de rappel (callbacks).

En novembre 2020, Espressif annonce la sortie du ESP32-C3, Wi-Fi & BLE, un processeur dans la lignée de l'ESP32, mais compatible broche à broche avec l'ESP8266, et utilisant l'architecture RISC-V 32 bits<sup>5</sup>. La figure suivante nous montre une ESP8266 NodeMCU:



*Figure 8: ESP8266 NodeMCU*

- Avantages
  - Interface sans fil (Wi-Fi),
  - un Single-core 32-bit cadencé à 80Mhz,
  - 17 Port E/S (GPIO),
  - programmable via l'environnement Arduino.
- Inconvénients : n'a aucun inconvénient majeur.

### 2.3.1.2. ESP23

ESP32 est une série de microcontrôleurs de type système sur une puce (SoC) d'Espressif Systems, basé sur l'architecture Xtensa LX6 de Tensilica, intégrant la gestion du Wi-Fi et du Bluetooth (jusqu'à LE 5.0 et 5.11) en mode double, et un DSP. C'est une évolution d'ESP8266. Le support du développement via la plateforme et l'IDE d'Arduino est disponible depuis la bibliothèque ESP32 2.0.03. Son support Wi-Fi et Bluetooth, est en fait un système apprécié dans le domaine de l'internet des objets. Ce SoC rencontre un certain succès depuis quelques années à la fois pour son coût, ses capacités et son intégration dans un nombre croissant de systèmes. La figure suivante nous montre une ESP32 LOLIN32:



Figure 9: ESP32 LOLIN32

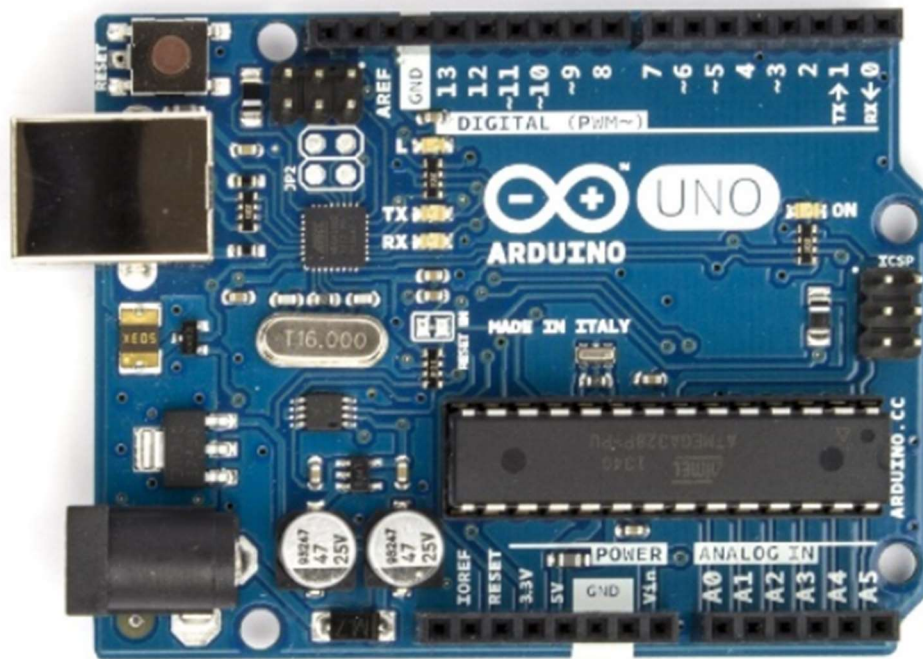
- Avantages
  - Interface sans fil (Wi-Fi et Bluetooth),
  - Un Dual-core 32-bit cadencé 160Mhz minium et 240Mhz,
  - 36 GPIO,
  - programmable via l'environnement Arduino.
- Inconvénients : n'a aucun inconvénient majeur.

## 2.3.2. Les Arduino

Arduino est essentiellement une plate-forme électronique open source. Comme il est open source, il existe de nombreuses cartes Arduino et elles sont proposées à des prix différents et avec des capacités différentes. La carte Arduino est essentiellement basée sur des puces ATmel.

### 2.3.2.1. Arduino Nano

Arduino Nano est particulièrement basé sur ATmega 328 ou 168. Arduino Nano est populaire en raison de sa petite taille, 8 broches d'entrée analogiques, 14 E/S numériques (dont 6 peuvent fournir une sortie PWM) et une vitesse d'horloge de 16 Mhz. La figure suivante nous montre un arduino UNO:



*Figure 10: Arduino UNO*

- Avantages
  - Forte communauté
  - un l'environnement très développé
- Inconvénients
  - Aucune interface sans fil.

### *2.3.2.2. Arduino Méga 2560*

L'une des cartes contrôleurs 8 bits les plus connues. cette carte présente de nombreuses variantes. Il dispose de 54 ports E / S, dont 15 fournissent PWM. La vitesse d'horloge à 16MHZ est certainement une table polyvalente. La figure suivante nous montre deux arduino mega 2560:



*Figure 11: Arduino Méga 2560*

- Avantages
  - Forte communauté
  - un l'environnement très développé
  - 54 ports E / S
- Inconvénients
  - Aucune interface sans fil.

### 2.3.3. Choix du microcontrôleur

Les Arduino Mega et UNO ont le même processeur ; la seule spécificité qu'a le Mega est d'avoir plus de port E/S que la UNO, L'ESP32 étant une évolution du ESP8266, notre synthèse va se concentrer sur le esp32 et l'Arduino UNO.

Nous allons plus nous intéresser dans cette synthèse à la puissance de calcul de chaque carte, leur connectique et leur prix. Le tableau 3 nous donnent une comparaison des différents microcontrôleurs :



Tableau 4: synthèse sur les microcontrôleurs sur carte

N°	Critères		ESP32 <sup>7</sup>	UNO <sup>8</sup>
1	Puissance	Processeur	Microprocesseur Xtensa Dual Core LX6 32 bits	ATMega328P
2		Mémoire flash	4Mo	32Ko
3		Vitesse de l'horloge	Jusqu'à 240Mhz	16Mhz
4	Connectivité	Broches d'E/S	36	14
5	Et Caractéristiques supplémentaires	USB	Micro USB	USB A / USB B
6		Wifi	802.11 b/g/n	-
7		Bluetooth	v4.2 BR/EDR et BLE	-
8	Prix	-	10 000 FCFA <sup>9</sup>	9 000 <sup>10</sup> FCFA

Dans cette comparaison, nous noterons que pour son prix, la ESP32 de Expressif est bien plus puissante et a plus de connectique que l'Arduino, son plus gros argument, ce sont ces interfaces sans fil.

<sup>7</sup> <https://www.espressif.com/>

<sup>8</sup> <http://arduino.cc/>

<sup>9</sup> Boutique d'électronique a carrefour bodjona

<sup>10</sup> Boutique d'électronique a carrefour bodjona



## 2.4. LES PROTOCOLES DE COMMUNICATION

### 2.4.1. Filaire

#### 2.4.1.1. KNX

KNX, aussi appelé Konnex, est un bus de terrain et un protocole d'automatismes pour le bâtiment. KNX est né de la fusion des standards EHS, EIB et Bâtibus à l'initiative de constructeurs. Cette initiative est due au besoin d'un standard permettant l'interopérabilité entre plusieurs constructeurs. Le protocole KNX est un protocole à logique répartie. Contrairement à d'autres protocoles d'automatismes, celui-ci ne fonctionne pas en mode maître/esclave, chaque automate étant indépendant des autres.



*Figure 12: KNX*

- Avantages
  - Raccordement en étoile des modules.
  - Pas d'ondes radios émises dans le logement (plus sain).
- Inconvénients
  - Coût des modules.
  - Nécessite un câble à paires torsadées.
  - Logiciel ETS onéreux pour le protocole KNX.

#### 2.4.1.1. CPL

Les CPL, c'est la possibilité de faire passer de l'information numérique (voix, donnée, image) sur le réseau électrique ordinaire.

L'intérêt de cette technologie porte sur l'utilisation d'un réseau filaire structuré déjà existant et parfaitement distribué dans toute la maison ou le bâtiment : le réseau électrique et ses points d'accès constitués par les prises électriques.

- Avantages
  - Utilisation des conducteurs déjà en place (réseau existant), ce qui facilite les rénovations.
  - Pas d'ondes radios émises dans le logement (plus sain).
- Inconvénients
  - Dans le cas d'installation triphasée, les modules branchés sur des phases différentes ne peuvent pas communiquer.
  - Le fonctionnement d'un appareil électrique peut parasiter le réseau et perturber la communication par un courant porteur.
  - Fonctionnement moins rapide.
  - La sécurité, possibilité de commander depuis l'extérieur si des trames de commande sont envoyées.

## 2.4.2. Sans Fil

### 2.4.2.1. Z-Wave

Le Z-Wave est un protocole radio de fréquence 868 MHz en Europe qui permet aux périphériques de communiquer entre eux à faible puissance soit sur pile ou sur l'alimentation réseau.

Sa portée est de l'ordre de 20 à 30 mètres en intérieur.



Figure 13: Z-Wave

- Avantages
  - Réseau maillé
  - Un périphérique Z-Wave est très simple à installer

- le Z-Wave fonctionne en 868 MHz et a donc une résistance naturelle aux interférences
- Inconvénients
  - l'installation d'un Z-Wave nécessite des connaissances en électricité,
  - License pour exploiter le z-wave.

#### 2.4.2.2. Wi-Fi

Le (ou la) Wi-Fi, aussi orthographié wifi, est un ensemble de protocoles de communication sans fil régis par les normes du groupe IEEE 802.11 (ISO/CEI 8802-11). Un réseau Wi-Fi permet de relier par ondes radio plusieurs appareils informatiques (ordinateur, routeur, smartphone, modem Internet, etc.) au sein d'un réseau informatique, afin de permettre la transmission de données entre eux.



Figure 14: Le Wi-Fi

- Avantages
  - Très répandus
  - Exploitation gratuite
- Inconvénients
  - Sujet à perturbation

#### 2.4.3. choix du protocole de communication (I)

Parmi les protocoles filaires, la seule pouvant avantager notre conception serait le CPL, car le KNX a des modules spécifiques, qui doivent supporter ce protocole donc implique des coûts, et cela nous empêcherait d'utiliser des produits standards, de même que le Z-wave au-delà de sa licence pour la production. Donc, nous allons comparer le CPL et le Wi-Fi, afin de pouvoir choisir un protocole pouvant répondre efficacement à nos besoins, dans le tableau suivant :

Tableau 5: comparatif Wi-Fi et CPL

N°	Critères	Wi-Fi	CPL
1	Cout de déploiement	Faible	Élevé
2	Mobilité	Plus avantageux	Pas très mobile
3	Stabilité du signale	Sujet a perturbation	Très stable
4	Sante	Les ondes radio	Aucun impact note

Au vu de nos comparatifs, le CPL est un candidat sérieux, mais présente un coût un peu trop élevé, mais vu la démocratisation du Wi-Fi et son faible coût d'implémentation, nous allons le choisir comme protocole de communication.

## 2.5. SYNTHÈSE ET CHOIX DES ÉLÉMENTS DE RÉALISATION

Nous avons présenté pour chaque technologie ou outils de travail deux ou plusieurs solutions qui permettent de mettre en place un système répondant aux besoins de notre projet. Après analyse, nous avons proposé une solution pour chaque technologie ou outils, Chacun d'eux a ses avantages et ses inconvénients. Le choix s'est fait en fonction des besoins et du risque acceptable. Le tableau ci-après fait un récapitulatif des solutions retenues pour la mise en place du système :

Tableau 6: récapitulatif des technologies et outil retenus.

N°	Besoin	Solution / outil retenus
1	Gestion d'habitat	AGO-AHome (développé par DIGITAL DEEP VISION)
2	Serveur de déploiement	Raspberry pi
3	Microcontrôleur	ESP32 (LOLIN 32)
4	Protocole de Communication	Wi-Fi

Dans ce chapitre, nous avons choisi les technologies qui nous permettront de mettre en place la solution adaptée aux besoins de DIGITAL DEEP VISION. Dans le prochain chapitre, nous allons déployer les solutions retenues sur les outils retenues.

## **CHAPITRE 3: MISE EN ŒUVRE ET PERSPECTIVES**

Dans ce chapitre, nous allons implémenter les différents composants qui nous permettront de déployer la solution retenue. Il sera question de configurer les outils et technologies retenues, de fiabiliser le système en assurant sa sécurité. Nous ferons ensuite les tests de fonctionnement. Cette solution n'étant pas gratuite, nous terminerons ce chapitre par son évaluation financière de celle-ci.

## 3.1. LES ELEMENTS DE REALISATION

---

Pour déployer la solution de DIGITAL DEEP VISION, après analyse, nous allons :

- installer du système d'exploitation,
- configurer du système,
- configurer des interfaces,
- installer des conditions nécessaires au déploiement,
- déployer d'un serveur web socket pour relier les microcontrôleurs au serveur,
- déployer de l'API développé par DIGITAL DEEP VISION,
- programmer des microcontrôleurs.
- concevoir de la maquette.

### 3.1.1. Ubuntu server 20.04 LTS ARM

Ubuntu est une distribution Linux basée sur Debian et composée majoritairement de logiciels libres et open-source. Toutes les éditions peuvent fonctionner sur l'ordinateur seul, ou dans une machine virtuelle. Ubuntu est publié tous les six mois, avec des versions de support à long terme (LTS) tous les deux ans. Avec la sortie du Raspberry Pi 2 Model B et de son processeur BCM2709 basé sur ARMv7, il est désormais possible d'exécuter Ubuntu directement sur le Raspberry Pi. Nous allons l'utiliser dans sa version serveur. La figure suivante illustre le logo de Ubuntu.



*Figure 15: logo de Ubuntu*

### 3.1.2. modprobe (2) (3)

modprobe est un programme Linux utilisé pour ajouter un module de noyau chargeable au noyau Linux ou pour supprimer un module de noyau chargeable du noyau. Il est couramment utilisé indirectement : udev s'appuie sur modprobe pour charger les pilotes du matériel détecté automatiquement.

### 3.1.3. RFkill (2)

RFkill est un sous-système du noyau Linux qui prend en charge les commutateurs d'arrêt pour les appareils de communication sans fil.

D'une part, il se compose d'une partie pilote et d'une interface de programmation générique (API), ainsi que d'un fichier de périphérique orienté caractères (situé sous) et du programme auxiliaire basé sur celui-ci pour commuter et interroger l'état de blocage à partir du mode utilisateur. /dev/rfkill

### 3.1.4. HostAPD (3)

Hostapd s'appuie sur les protocoles IEEE 802.11 AP et IEEE 802.1X/WPA/WPA2/EAP/RADIUS authentificateur.

Hostapd permet la création d'un point d'accès Wi-Fi, technologie sans fil utilisée pour se connecter à un réseau informatique. Dans les réseaux informatiques, un point d'accès sans fil (spot ou AP) est un dispositif qui relie les appareils de communication



sans fil pour former un réseau sans fil. Le spot Wi-Fi se connecte généralement à un réseau câblé, et peut transmettre des données entre les appareils sans fil et les périphériques câblés.

### 3.1.5. Dnsmasq

Dnsmasq est un serveur léger conçu pour fournir les services DNS, DHCP, Bootstrap Protocol et TFTP pour un petit réseau, voire pour un poste de travail. Il permet d'offrir un service de nommage des machines du réseau interne non intégrées au service de nommage global (i.e. le service DNS d'Internet). Le service de nommage est associé au service d'adressage de telle manière que les machines dont le bail DHCP est fourni par Dnsmasq peuvent avoir automatiquement un nom DNS sur le réseau interne. Le logiciel offre un service DHCP statique ou dynamique.



*Figure 16: logo Dnsmasq*

### 3.1.6. Iptables (4)

Iptables est une interface en ligne de commande permettant de configurer Netfilter. En plus de Iptables, depuis la version 8.04, Ubuntu est installé avec la surcouche UFW qui permet de contrôler simplement Netfilter, UFW est toutefois moins complet que iptables.

### 3.1.7. netfilter-persistent (4)

netfilter-persistent utilise un ensemble de plugins pour charger, vider et enregistrer les règles netfilter au démarrage et temps d'arrêt. Les plugins peuvent être écrits dans n'importe quel langage approprié et stockés dans `/usr/share/netfilter-persistent/plugins.d`

### 3.1.8. NodeJS (4)

Node.js est une plateforme logicielle libre en JavaScript, orientée vers les applications réseau.

Parmi les modules natifs de Node.js, on retrouve http qui permet le développement de serveur HTTP. Ce qui autorise, lors du déploiement de sites internet et d'applications web développés avec Node.js, de ne pas installer et utiliser des serveurs webs tels que Nginx ou Apache.

Concrètement, Node.js est un environnement bas niveau permettant l'exécution de JavaScript côté serveur.



*Figure 17:logo NodeJS*

### 3.1.9. npm (4)

npm est le gestionnaire de paquets par défaut pour l'environnement d'exécution JavaScript Node.js de Node.js.

npm se compose d'un client en ligne de commande, également appelé npm, et d'une base de données en ligne de paquets publics et privés payants, appelée le registre npm. Le registre est accessible via le client, et les paquets disponibles peuvent être parcourus et recherchés via le site Web de npm. Le gestionnaire de paquets et le registre sont gérés par npm, Inc.



*Figure 18: logo npm*

### 3.1.10. Python

Python est un langage de programmation interprété, multiparadigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions ; il est ainsi similaire à Perl, Ruby, Scheme, Smalltalk et Tcl.

Le langage Python fonctionne sur la plupart des plateformes informatiques, des smartphones aux ordinateurs centraux, de Windows à Unix avec notamment GNU/Linux en passant par macOS, ou encore Android, iOS, et peut aussi être traduit en Java ou .NET. Il est conçu pour optimiser la productivité des programmeurs en offrant des outils de haut niveau et une syntaxe simple à utiliser.



*Figure 19: logo python*

### 3.1.11. Flask (5)

Flask est un micro Framework open-source de développement web en Python. Il est classé comme micro Framework car il est très léger.<sup>2</sup> Flask a pour objectif de garder un noyau simple mais extensible. Il n'intègre pas de système d'authentification, pas de couche d'abstraction de base de données, ni d'outil de validation de formulaires. Cependant, de nombreuses extensions permettent d'ajouter facilement des fonctionnalités.



*Figure 20: logo Flask*

### 3.1.12. MongoDB (5)

MongoDB (de l'anglais humongous qui peut être traduit par « énorme ») est un système de gestion de base de données orienté documents, répartissable sur un nombre quelconque d'ordinateurs et ne nécessitant pas de schéma prédéfini des données. Il est écrit en C++. Il fait partie de la mouvance NoSQL.



*Figure 21: logo MongoDB*

### 3.1.13. les différents capteurs du système

#### 3.1.13.1. Capteur de courant ACS712 20A de AZDelivery

Un capteur de courant permet de mesurer le champ magnétique créé par un courant qui traverse un fil électrique. Il s'agit d'un composant électrique qui convertit le courant électrique détecté en une autre sorte de signal électrique, qui est proportionnel à la valeur du courant. Le courant est transformé selon divers procédés physiques, afin de

devenir une tension mesurable. Ce signal peut alors être une tension analogique, une intensité ou alors une sortie numérique et donne la mesure d'un courant dans un appareil appelé ampèremètre.

Le capteur de courant ACS712 de AZDelivery permet de mesurer le courant dans les circuits électriques et électroniques avec une puissance allant jusqu'à 20 Ampères et la sortie est de 185 mV/A. Sa taille compacte est pratique et facilite son montage.

Le dispositif comporte une puce de capteur et est tout à fait compatible avec un Arduino ou tout autre circuit.



*Figure 22: Capteur de courant ACS712 de AZDelivery*

### 3.1.13.2. Présence HC-SR501

Un capteur infrarouge passif (capteur PIR) est un capteur électronique qui mesure la lumière infrarouge (IR) émise par des objets dans son champ de vision. Ils sont le plus souvent utilisés dans les détecteurs de mouvement basés sur PIR. Les capteurs PIR sont couramment utilisés dans les alarmes de sécurité et les applications d'éclairage automatique.

Les capteurs PIR détectent un mouvement général, mais ne donnent pas d'informations sur qui ou quoi a bougé. Pour cela, un capteur IR d'imagerie est nécessaire.

Les capteurs PIR sont communément appelés simplement "PIR", ou parfois "PID", pour "détecteur infrarouge passif". Le terme passif fait référence au fait que les

dispositifs PIR n'émettent pas d'énergie à des fins de détection. Ils fonctionnent entièrement en détectant le rayonnement infrarouge (chaleur rayonnante) émis ou réfléchi par des objets.

Le HC-SR501 est un module de contrôle automatique basé sur la technologie infrarouge qui a une sensibilité et une sécurité élevées et fonctionne en mode très basse tension. Il est courant que le module soit appliqué dans une gamme d'équipements électroniques à détection automatique, en particulier dans les produits de contrôle automatique alimentés par une batterie sèche.



*Figure 23: capteur de présence HC-SR501*

### 3.1.13.3. Température DHT22

Le capteur DHT22 (aussi connu sous la référence AM2302) est des capteurs de température et d'humidité "deux en un".

Le capteur DHT22 / AM2302 est capable de mesurer des températures de -40 à +125°C avec une précision de +/- 0.5°C et des taux d'humidité relative de 0 à 100% avec une précision de +/- 2% (+/- 5% aux extrêmes, à 10% et 90%). Une mesure peut être réalisée toutes les 500 millisecondes (soit deux fois par seconde).



*Figure 24: DHT22*

## 3.2. MISE EN ŒUVRE

### 3.2.1. Installation du système d'exploitation

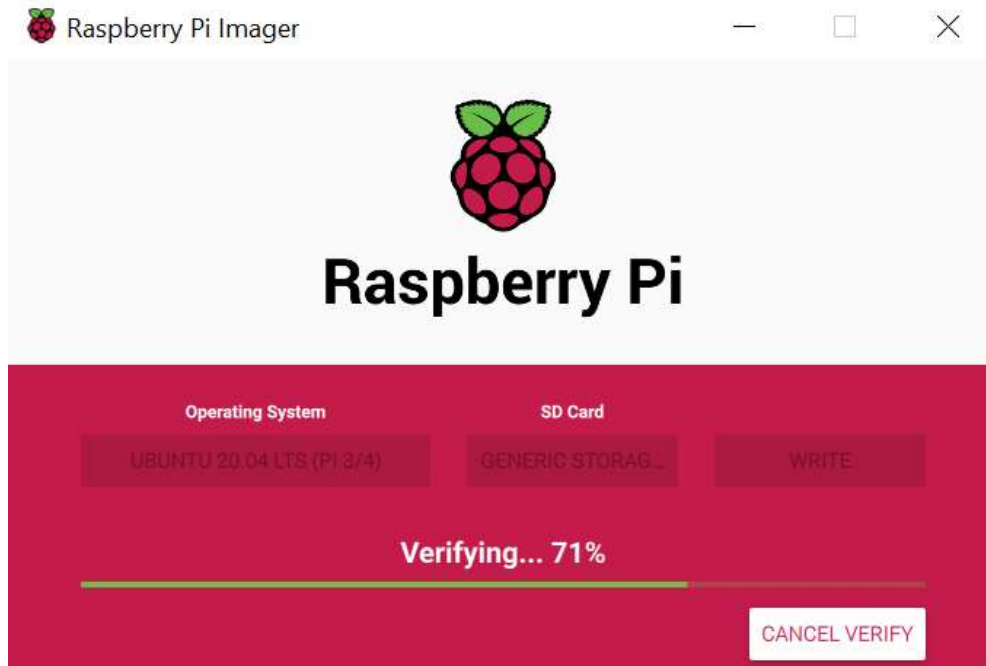
la version 20.04 LTS (Long Time Support) Focal Fossa de son système d'exploitation. Ceci garantit un support pour les 5 prochaines années sur la version serveur et la version Desktop (soit 2025). Nous allons l'installer et mettre en service un Ubuntu 20.04 sur une Raspberry pi 4 de 4Gb de ram.

#### 3.2.1.1. Préparez la carte SD

Tout d'abord, insérez la carte microSD dans votre ordinateur.

- Vous devez maintenant installer Raspberry Pi Imager pour votre système d'exploitation.
- Une fois cela fait, démarrez l'imageur et ouvrez le menu "CHOOSE OS".
- Faites défiler le menu, cliquez sur "Autre système d'exploitation à usage général".
- Ici, vous pouvez sélectionner Ubuntu et voir une liste d'options de téléchargement, donc il faut que le PC ou le Raspberry Pi soit connecté à Internet. Choisissez le serveur Ubuntu 20.04 LTS 64bits pour (Pi 3/4),
- Sélectionnez l'image et ouvrez le menu "Carte SD". Sélectionnez la carte microSD que vous avez insérée.

- Modifiez le fichier de configuration en cliquant sur l'icône en forme de roue crante ; activant le ssh, ajoutez un utilisateur et un nom a la machine, enregistrez vos modifications.
- Enfin, cliquez sur "WRITE" et attendez que la magie se produise... (Cette magie peut prendre quelques minutes)



*Figure 25: Raspberry Pi Imager terminant l'écriture du microSD*

### 3.2.1.2. Configuration réseau (7)

Une fois la vérification terminer,

- connectez votre Pi à votre routeur avec un câble Ethernet
- Avec la carte SD toujours insérée dans votre ordinateur portable, ouvrez un gestionnaire de fichiers et localisez la partition "system-boot" sur la carte. Il contient les fichiers de configuration initiaux qui se chargent lors du premier processus de démarrage.
- Modifiez le fichier network-config pour ajouter vos informations
- ligne dans le fichier network-config avec des lignes qui spécifient l'adresse IP prévue ainsi que sa passerelle par défaut et son serveur DNS.



```

ethernets:
  eth0:
    addresses:
      - 192.168.1.20/24
    gateway4: 192.168.1.254
    nameservers:
      addresses: [8.8.8.8]
    optional: true

```

*Figure 26: configuration de l'interface eth0*

### 3.2.1.3. Démarrage du serveur et connexion

Insérer votre carte microSD dans la PI, mettez la sous tension et attendez. Après environ une minute, Ubuntu sur votre Raspberry Pi sera entièrement démarré et connecté au réseau. Sur Ubuntu et Mac OS, un client SSH est déjà installé. Certaines versions de Windows 10 incluent également un client SSH, mais si le vôtre ne le fait pas, vous pouvez installer PuTTY qui est un excellent client ssh.

- Ouvrez un terminal et exécutez la commande suivante :
  - `ssh ahome@192.168.1.20`
- Il vous sera demandé de confirmer la connexion
- Tapez "oui" pour confirmer.
- mettez le mot de passe défini auparavant.

Votre système d'exploitation est parfaitement installé et vous êtes connecté par ssh.

## 3.2.2. Configuration de l'interface Wlan0

### 3.2.2.1. Configuration IP adresse statique.

Netplan est un utilitaire de ligne de commande utilisé pour configurer la mise en réseau sur les distributions Ubuntu modernes. Netplan fait usage de YAMLS pour configurer les interfaces réseau. Vous pouvez configurer une interface pour acquérir une adresse

IP dynamiquement en utilisant DHCP ou définissez une adresse IP statique. Nous utiliserons Netplan pour fixer l'IP sur l'interface wlan0.

- Ouvrez le fichier de configuration à l'emplacement « /etc/netplan/01-network-manager-all.yaml » et modifiez-le comme suit :

```
network:

  version: 2

  ethernets:

    eth0:

      dhcp4: false

      addresses: [192.168.1.20/24]

      gateway4: 192.168.1.254

      nameservers:

        addresses: [192.168.1.53]

  optional: true

  wifis:

    wlan0:

      dhcp4: false

      addresses: [10.20.1.1/24]
```

*Figure 27: fichier de configuration 01-network-manager-all.yaml*

### 3.2.2.2. Configuration en AP (Access Point) (3)

Nous allons, grâce à Hostapd, configurer l'interface wlan0 en AP, afin d'y connecter les MCUs.

- Nous allons dans un premier temps installer les packages nécessaires pour cela notamment : modprobe, pour ajouter le pilote au noyau ; RFkill (2) pour débloquer l'interface sans fil, hostapd pour configurer l'AP.
  - `sudo apt-get install -y net-tools`
  - `sudo apt-get install -y modprobe`

- `sudo apt-get install -y rfkill`
- `sudo apt-get install -y hostapd`
- **activons l'interface**
  - `sudo ifconfig wlan0 up`
- **créons le fichier de configuration et configurons l'AP,**
  - `sudo touch /etc/hostapd/hostapd.conf`
  - `sudo echo -e "country_code=TG\ninterface=wlan0\nssid=AgoAHome\nhw_mode=g\nchannel=7\nmacaddr_acl=0\nauth_algs=1\nignore_broadcast_ssid=1\nwpa=2\nwpa_passphrase=Sister00@\nwpa_key_mgmt=WPA-PSK\nwpa_pairwise=TKIP\nrsn_pairwise=CCMP\n">>/etc/hostapd/hostapd.conf`
- **débloquons l'interface et activons le daemon**
  - `sudo rfkill unblock wlan`
  - `sudo systemctl unmask hostapd.service`
  - `sudo systemctl enable hostapd.service`
  - `sudo hostapd /etc/hostapd/hostapd.conf &`

Nous venons de configurer un wifi caché avec hostapd, mais il nous manque un serveur DHCP pour desservir l'IP à nos clients.

### 3.2.3. Configuration du nom de domaine et du DHCP

Graca a Dnsmasq nous allons configurer un serveur DNS et DHCP

- Ubuntu est livré avec systemd-resolve que vous devez désactiver car il se lie au port 53 qui entrera en conflit avec le port Dnsmasq.
  - `sudo systemctl disable systemd-resolved`
  - `sudo systemctl stop systemd-resolved`
- **Supprimez également le resolv.conf fichier lié symboliquement**
  - `ls -lh /etc/resolv.conf`
  - `sudo unlink /etc/resolv.conf`
- **Créez ensuite un nouveau fichier resolv.conf**
  - `echo nameserver 8.8.8.8 | sudo tee /etc/resolv.conf`
- **Installation de Dnsmasq**
  - `sudo apt-get install -y dnsmasq`
- **Création de sauvegarde du fichier de configure**
  - `sudo mv /etc/dnsmasq.conf /etc/dnsmasq.conf.ald`
- **Configuration de Dnsmasq (3)**

- `sudo echo -e "interface=wlan0\ndhcp-range=10.20.1.5,10.20.1.100,255.255.255.0,36d\ndomain=ago\naddress=mcu.ago/10.20.1.1\naddress=/ahome.ago/192.192.1.20">>/etc/dnsmasq.conf`
- Ajoutez des enregistrements DNS dans le fichier `/etc/hosts`. Dnsmasq répondra aux requêtes des clients utilisant ces enregistrements.
  - `Sudo vim /etc/hosts`
  - `10.20.1.1 mcu.ago`
  - `192.168.1.20 ahome.ago`
- Vous devez redémarrer le service dnsmasq
  - `sudo systemctl restart dnsmasq`

### 3.2.4. Installation et programmation du serveur socket

#### 3.2.4.1. installation de NodeJS

Ubuntu 20.04 contient une version de Node.js dans ses référentiels par défaut qui peut être utilisée pour fournir une expérience cohérente sur plusieurs systèmes.

- Installons NodeJS
  - `sudo apt install nodejs`

Si le package dans les référentiels répond à vos besoins, c'est tout ce que vous devez faire pour installer Node.js. Dans la plupart des cas, vous voudrez également installer npm, le gestionnaire de package Node.js. Vous pouvez le faire en installant le paquet npm avec apt :

- `sudo apt install npm`
- installons les dépendances de notre code,
  - `npm install fs`
  - `npm install ws`

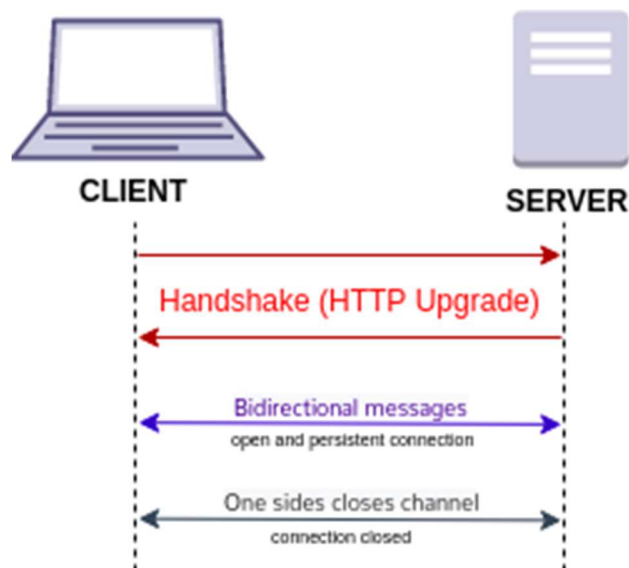
### 3.2.5. Le programme

Le serveur que nous avons programmé a pour rôle principal de connecter les MCU, recevoir les messages à l'intention du serveur applicative, et le transférer au MCU destinataire, et inversement ; cela à tout moment (en temps réel).

Pour cela nous avons, utilisé du Web Socket et nous avons défini un datagramme pour les échanges entre le SVR-APP, SVR-WS et les MCU.

### 3.2.6. WebSocket (8)

WebSocket est un standard du Web désignant un protocole réseau de la couche application et une interface de programmation du World Wide Web visant à créer des canaux de communication full-duplex par-dessus une connexion TCP. Le protocole a été normalisé par l'IETF dans la RFC 64552 en 2011.



*Figure 28: schéma illustrant les échanges webSocket*

### 3.2.7. Le datagramme

Pour les échanges de données, nous avons dû établir un datagramme sous un format Json :

```
Datagram = {
  "IDSender": "",
  "IDDestiner": "",
  "State" : [0,0,0]
}
```

*Figure 29: datagramme d'échanges de donnée.*

- IDSender désigne l'id de l'émetteur
- IDDestiner désigne l'id du destinataire
- State désigne les états de l'appareil
  - Le premier élément pour allumer ou éteinte (0 ou 1).
  - Le deuxième la consommation de l'appareil

- Le troisième désigne le type de l'appareil (1 [une lampe], 2 [une prise], 3 [un capteur])

Pour les capteurs de mouvement et de température, leurs valeurs sont renseignées dans state respectivement à la première et seconde position.

### 3.2.8. Installation du SGBD MongoDB (5)

Pour installer MongoDB Community sur Ubuntu, nous utiliserons le mongodb-org package officiel, qui est maintenu et pris en charge par MongoDB Inc. Le mongodb-org package officiel contient toujours la dernière version de MongoDB et est disponible à partir de son propre référentiel dédié.

- Importez la clé publique utilisée par le système de gestion des packages, lancez la commande suivante pour importer la clé GPG publique MongoDB à partir de <https://www.mongodb.org/static/pgp/server-6.0.asc>:
  - `wget -qO - https://www.mongodb.org/static/pgp/server-6.0.asc | sudo apt-key add -`
  - L'opération doit répondre par un OK.
- Créez un fichier de liste pour MongoDB `/etc/apt/sources.list.d/mongodb-org-6.0.list`.
  - `echo "deb [arch=amd64, arm64] https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/6.0 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-6.0.list`
- Exécutez la commande suivante pour recharger la base de données de packages locale :
  - `sudo apt-get update`
- Vous pouvez installer soit la dernière version stable de MongoDB, soit une version spécifique de MongoDB.
  - `sudo apt-get install -y mongodb-org`
- Vous pouvez démarrer le processus mongod en exécutant la commande suivante :
  - `sudo systemctl daemon-reload`
  - `sudo systemctl start mongod`
- Vous pouvez éventuellement vous assurer que MongoDB démarrera après un redémarrage du système en exécutant la commande suivante :
  - `sudo systemctl enable mongod`

- Démarrez une mongosh session sur la même machine hôte que le mongod. Vous pouvez exécuter mongosh sans aucune option de ligne de commande pour vous connecter à un mongod qui s'exécute sur votre hôte local avec le port par défaut 27017.

Voilà MongoDB est installé et prêt à l'emploi.

### 3.2.9. Installation du serveur applicatif (5)

Les Packages Flask sont inclus dans les dépôts officiels d'Ubuntu et peuvent être installés à l'aide du « apt » gestionnaire de packages.

Ubuntu 20.04 est livré avec Python 3.8. Vous pouvez vérifier que Python est installé sur votre système en tapant :

- `python3 -V`
- installez pip3 et d'autres outils Python :
  - `sudo apt -y install build-essential python3-pip libffi-dev python3-dev python3-setuptools libssl-dev`
- utilisez le gestionnaire de packages Python « pip » pour installer Flask :
  - `pip3 install flask`

#### 3.2.9.1. Installation de l'API

Nous allons télécharger et installer l'API développé par DIGITAL DEEP VISION sur un dépôt privée

- Installation de git hub
  - `sudo apt-get install git-all`
- Maintenant il nous faut cloner l'API depuis le depot
  - `git clone https://github.com/nickAP02/api_ahome.git`
- Il vous faut vous identifier et c'est bon.
- Pour pouvoir lancer le Serveur, il nous faut installer les prérequis de ce dernier en faisant :
  - `Sudo pip install api_ahome/requiment.txt`
- Maintenant, nous pouvons la lancer en faisant :
  - `Sudo Python api_ahome/main.py`

## 3.2.10. Programmation des MCUs

### 3.2.10.1. Environnement Arduino

L'environnement de développement intégré Arduino, ou logiciel Arduino (IDE), contient un éditeur de texte pour écrire du code, une zone de message, une console de texte, une barre d'outils avec des boutons pour les fonctions communes et une série de menus. Il se connecte au matériel Arduino pour télécharger des programmes et communiquer avec eux.



*Figure 30: logo d'Arduino*

Pour télécharger l'environnement de développement Arduino, il faut vous rendre sur le site officiel du projet Arduino, puis cliquer sur l'onglet "Download" ("téléchargement" en français).

Elle nous permettra de coder les MCU avec du C.

### 3.2.10.2. Structure du code des MCU

Comme annoncé plus haut, un microcontrôleur a une et une seule routine qu'il exécute en boucle, compte tenu du fait que dans notre conception un MCU est appelé à gérer 10 appareils et un couple de capteur de température et de présence, cela nous limiterait énormément. En programmation système, base niveau ; nous avons la possibilité de créer des threads pour chaque processus, donc des tâches parallèles.

Donc nous avons trois tâches :

- La première chargée d'allumer et éteindre un appareil parmi ceux configurés sur le MCU
- La second récolte les donne des capteurs de courant par rapport à chaque appareil, afin de déterminer sa consommation, et s'il est allumé ou éteint.



- Le troisième, l'état des capteurs de présences et de température.

Chacun des différents états récoltés dans les trois tâches est envoyé au serveur de MCU puis au serveur applicatif pour sauvegarde et historisation.

### 3.2.11. Création des daemons

Ubuntu 20.04 est basé sur Systemd, donc le moyen recommandé pour exécuter un script au démarrage est de créer un fichier de service Systemd, via ce service pendant le système démarrage.

Les étapes ci-dessous nous montreront comment exécuter le Programme au démarrage

- nous allons créer un fichier de configuration (alias un fichier d'unité) qui indique à systemd ce que nous voulons qu'il fasse. Créons le fichier unite :
  - `sudo nano /lib/systemd/system/api-ahome.service`
- Ajoutez dans le texte suivant :

```
[Unit]
Description=Api Ago AHome
After=multi-user.target

[Service]
Type=idle
ExecStart=/usr/bin/python3 /home/ahome/api_ahome/main.py >
/home/ahome/api_ahome/api-ahome.log 2>&1
WorkingDirectory=/home/ahome/api_ahome
User=ahome

[Install]
WantedBy=multi-user.target
```

*Figure 31: configuration du fichier unité*

- L'autorisation sur le fichier d'unité doit être définie sur 644 :
  - `sudo chmod 644 /lib/systemd/system/api-ahome.service`
- Maintenant que le fichier unité a été défini, nous pouvons dire à systemd de le démarrer pendant la séquence de démarrage :
  - `sudo systemctl daemon-reload`
  - `sudo systemctl enable api-ahome.service`
  - `sudo systemctl start api-ahome.service`
  - `sudo systemctl status api-ahome.service`
- faisons de même pour le Serveur websocket.

### 3.2.12. Configuration du pare-feu

La règle de pare-feu ici simplement c'est son simple, nous devons nous assurer grâce au pare-feu nous avons seulement le strict minimum autorisé qui sont :

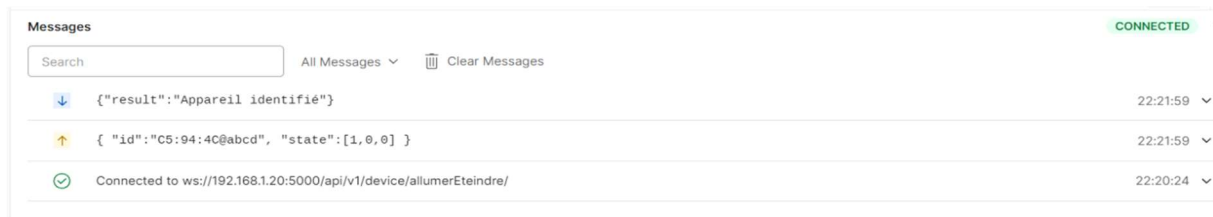
- autoriser le SSH pour les 2 sous réseau.
  - `iptables -t filter -A input --dport ssh -j ACCEPT`
- Autoriser le HTTP sur le port 5000 pour le sous-réseau 192.168.0.1/24
  - `iptables -t filter -A input -s 192.168.0.1/24 --dport 5000 -j ACCEPT`
- Autoriser le HTTP sur les ports 7000 pour le sous-réseau 10.20.0.1/24
  - `iptables -t filter -A input -s 10.20.1.0/24 --dport 7000 -j ACCEPT`
- Bloquer tout le reste.
  - `iptables -t filter -A input -j Drop`

## 3.3. TEST DE FONCTIONNEMENT

---

Pour effectuer les tests de fonctionnement de notre système, nous devons vérifier la transmission du message à chaque niveau, vérifier à la latence pour la transmission d'un message de l'application mobile vers le serveur applicatif en passant par le serveur websocket puis vers le microcontrôleur.

- Envoie du message du client du client web



*Figure 32: connexion web socket entre le client Postman et le serveur applicative*

- Réception sur le serveur applicatif et transmission au websocket

```
* Serving Flask app 'main'
* Debug mode: off
Address already in use
Port 5000 is in use by another program. Either identify and stop that program, or start the server with a different port

ahome@ahome:~$ sudo systemctl stop api-ahome.service
ahome@ahome:~$ python api_ahome/main.py
Connection établie avec le client
Connection établie avec la bd
Connection établie avec la collection users
Connection établie avec la collection roles
Connection établie avec la collection rooms
Connection établie avec la collection devices
Connection établie avec la collection historics
Connection établie avec la collection capteurs
Connection établie avec la collection plannings
Connection établie avec la collection params
Connection établie avec la collection previsions
* Serving Flask app 'main'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://192.168.1.20:5000
Press CTRL+C to quit

{"id": "C5:94:4C@abcd",
 "state": [1,0,0]}
```

*Figure 33: serveur applicative*

- Réception par le serveur web socket et transféret au microcontrôleur

```
ws.service - ws server
Loaded: loaded (/lib/systemd/system/ws.service; enabled; vendor preset: enabled)
Active: active (running) since Wed 2022-08-17 22:19:11 GMT; 3min 7s ago
Main PID: 2202 (node)
Tasks: 7 (limit: 4432)
CGroup: /system.slice/ws.service
└─2202 /usr/bin/node /home/ahome/Server_Socket_v2/Server.js > /home/ahome/Server_Socket_v2/ws.log 2>&1

Aug 17 22:19:11 ahome systemd[1]: Started ws server.
Aug 17 22:20:00 ahome node[2202]: server running
Aug 17 22:20:01 ahome node[2202]: 290-9c-9b-c0a4 ::ffff:127.0.0.1
Aug 17 22:20:01 ahome node[2202]: 97e-ac-8f-9d50 ::ffff:127.0.0.1
Aug 17 22:20:01 ahome node[2202]: c5d-9b-a2-ac81 ::ffff:127.0.0.1
Aug 17 22:20:01 ahome node[2202]: 7a5-b9-81-7fb4 ::ffff:127.0.0.1
Aug 17 22:20:01 ahome node[2202]: disconnected 7a5-b9-81-7fb4 ::ffff:127.0.0.1
Aug 17 22:20:01 ahome node[2202]: disconnected 97e-ac-8f-9d50 ::ffff:127.0.0.1
Aug 17 22:22:00 ahome node[2202]: Mise a jour de Id du serveur!
Aug 17 22:22:00 ahome node[2202]: Envoie a C5:94:4C@abcd
```

*Figure 34: Serveur web Socket*

- Réception sur le microcontrôleur et exécution de l'instruction.

```
l.199 -> {"IDSender":"C5:94:4C@abcd","IDDestiner":"R3:87:Ts@9613","State":[0,0,1]}
l.246 -> msg send
```

*Figure 35: console port série du MCU*

- Réception des états du capteur sur le microcontrôleur et envoi des informations du capteur du microcontrôleur au serveur websocket

```
420 -> 28.00
420 -> {"IDSender":"C5:94:4C@aaca","IDDestiner":"R3:87:Ts@9613","State":[0,27,3]}
```

*Figure 36: message des capteurs dans console port série du MCU*

Nous observons alors ici que toute la chaîne de transmission est respectée et s'effectue en un temps très faible quasi indescriptible. Donc nous avons pu répondre à nos attentes, notamment envoyer des informations au microcontrôleur et en recevoir, connaître les états des appareils (allumés ou éteints) ; connaître leur consommation et l'état des différents capteurs.

### 3.4. EVALUATION FINANCIERE

L'évaluation financière du projet a été réalisée en fonction de deux éléments : les ressources humaines et les ressources matérielles.

- les ressources humaines : il s'agit des coûts liés à la conception et au déploiement de la solution ;
- les ressources matérielles : il s'agit des coûts liés à l'acquisition des matérielles pour la réalisation de la maquette

Les tableaux donnent un résumé des coûts liés à la mise en place de la solution et des coûts liés à l'acquisition du matériel :

*Tableau 7: Tableau des coûts liés à la mise en place de la solution*

N°	Désignation	Unité	Prix Unitaire (CFA)	Quantité	Total (CFA)
1	Conception et déploiement de la solution	Heure/Homme	5000 <sup>11</sup>	400	2 000 000
2	Coût de la maintenance	06 Mois	35 000	2	70 000
<b>Sous-total 1</b>					<b>2 070 000</b>

*Tableau 8: Tableau des coûts liés à l'acquisition du matériels*

N°	Désignation	Unité	Prix Unitaire (CFA)	Quantité	Total (CFA)
1	Raspberry pi 4 Gb/32GB	1	130 000 <sup>12</sup>	1	130 000
2	fils électriques	m	800 <sup>13</sup>	15	12 000
3	capteur de courant	1	1 700	9	15 300

<sup>11</sup> Cours de maintenance deuxième niveaux, l'heure ingénieur.

<sup>12</sup> <https://www.fnac.com/>

<sup>13</sup> CTT Bâtiment

<b>4</b>	Microcontrôleur	1	9 000	1	9 000
<b>5</b>	capteur de Température	1	2 500	5	12 500
<b>6</b>	capteur de Présence	1	2 500	5	12 500
<b>7</b>	transformateur	1	5 000	1	5 000
<b>8</b>	relais 15A 230v	1	5000	8	40 000
<b>9</b>	interrupteur vas-et-viens	1	1500 <sup>14</sup>	2	3 000
<b>10</b>	Prise électrique	1	500	4	2 000
<b>11</b>	Par terre droit	1	800	4	3 200
<b>12</b>	les ampoules électriques	1	800	4	3 200
<b>13</b>	Boîte de dérivation	1	900	1	900
<b>14</b>	Boitier d'isolation	1	200	6	1 200

---

<sup>14</sup> CTT BATIMENT

<b>15</b>	Transport et livraison	1	45 000	-	45 000 <sup>15</sup>
<b>Sous-total 2</b>					295 000

*Tableau 9: coûts totaux de l'implémentation du système*

N°	Sous-Total	Montant (F CFA)
<b>1</b>	Sous-total 1 : coûts liés à la mise en place de la solution	2 070 000
<b>2</b>	Sous-total 2 : coûts liés à l'acquisition du matériels	295 000
<b>TOTAL</b>		2 365 000

Ce troisième chapitre nous a permis de présenter les différents matériels et logiciels utilisés dans la Conception et implémentation d'un système de contrôle informatique de la consommation d'énergie d'un habitat, mais aussi de présenter les différentes installations et configurations effectuées ainsi que son coût de mise en place. Nous aborderons dans la suite un dernier chapitre nommé « Guide d'utilisation ».

---

<sup>15</sup> CTT BATIMENT

## **PARTIE 3: GUIDE D'UTILISATION**



Dans la partie précédente de ce document, nous avons conçu et implémenté un système de contrôle informatique de la consommation d'énergie d'un habitat. La présente partie « Guide d'utilisation » servira à guider les utilisateurs du système sur les manipulations requises.

## 3.1. GUIDE D'UTILISATION POUR LES CLIENTS

Notre solution est fournie sous forme d'une box, donc l'utilisateur n'aura qu'à brancher la box au secteur puis à son routeur et télécharger l'application mobile sur le Store. Une fois le démarrage terminé, il peut utiliser la solution.

Après le lancement de l'application mobile, la première étape que nous devons passer est l'étape d'authentification. Pour ce faire, l'utilisateur renseigne l'email et le mot de passe associés à son compte comme suit.



*Figure 37: Page d'authentification*

Après l'authentification, nous avons accès à l'accueil où nous avons la barre de navigation comme ci-dessous :



*Figure 38: Barre de navigation de l'application*

Nous avons un bouton principal, qui nous permet de faire de nouveaux enregistrements, comme ci-dessous :



*Figure 39: Bouton d'ajout d'une pièce*

L'interface d'ajout d'une pièce apparaît lorsqu'on clique sur le bouton « ajout d'une pièce » comme suit :

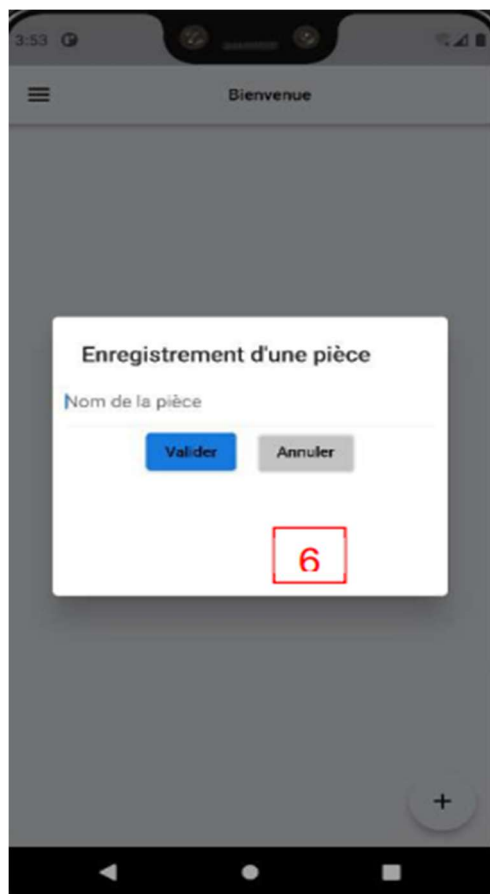


Figure 40: Page d'ajout d'une pièce

## CONCLUSION

---

Ce stage de trois (03) mois effectué au sein de la société DIGITAL DEEP VISION nous a permis de mettre en pratique toutes nos connaissances acquises durant nos trois (03) années de formation à l'IAI-TOGO. Nous avons été confronté à des difficultés réelles plus ou moins importantes du monde du travail, que nous avons pu surmonter. Ce document composé de trois (03) parties, que sont : la présentations, l'étude et réalisation du projet et du guide d'utilisation est le résultat du travail que nous avons effectué au sein de la société DIGITAL DEEP VISION durant notre période de stage. Le produit de notre travail, est un système qui permet au potentiel client de DIGITAL DEEP VISION d'avoir un système de gestion et de contrôle informatique de la consommation d'énergie de son habitat. Ce stage a été pour nous une occasion d'acquérir de nouvelles connaissances, notamment dans le domaine de la domotique, un domaine qui est en plein développement et qui nous intéresse particulièrement. Pour des raisons de performances, ne serait-il pas intéressant de déployer une solution d'authentification Cote matériels ? Rendre accessible le système à distance grâce à un reverse proxy ?

## BIBLIOGRAPHIE INDICATIVE

---

- Apprenez le fonctionnement des réseaux tcp IP, Par Romain Guichard (Caelifer), elalitte
  - Faire communiquer les machines entre elles. p29 – p37« tout au long du projet »
- L'électronique de zéro, Par Basil1402, olyte ,Polinichon et uknow
  - Application p27 – p37
  - Lois fondamentales p38 – p43« tout au long du projet »
- Arduino à l'école, Par Frédéric Genevey - Jean-Pierre Dulex
  - Découverte de la plateforme Arduino p13 – p16
  - Les bases de l'électronique p19 – p21« tout au long du projet »

## ANCIENS MEMOIRES CONSULTES

---

- « ETUDE ET MISE EN PLACE D'UN CLOUD HYBRIDE DE TYPE IAAS : CAS DE SOFTSOLUX » AFIVI Daniel (2020-2021) ;
  - Plan de localisation de l'IAI-TOGO p5« tout au long du projet »
- « SECURISATION D'UN SYSTEME DE STOCKAGE NAS : LES ENVIRONNEMENTS JAILS (TRUENAS CORE) » AKAKPO Amandine Laura (2020-2021)
  - Structure organisationnelle de l'IAI-TOGO p4.« tout au long du projet »

## NOTES DE COURS

---

- Programmation système 2021-2022 de Mr EKOUE-KOVI, les threads.

## WEBOGRAPHIE INDICATIVE

---

1. **Electrotoile.** presentation domotique. *Electrotoile.* [En ligne] [https://electrotoile.eu/presentation\\_domotique.php](https://electrotoile.eu/presentation_domotique.php).
2. **ubuntu.** le noyau Ubuntu. *ubuntu documentation.* [En ligne] <https://doc.ubuntu-fr.org/kernel>.
3. —. pilote manquant du noyau Ubuntu. *ubuntu forum.* [En ligne] <https://forum.ubuntu-fr.org/viewtopic.php?id=1439761>.
4. **debian-facile.** utilisation de RFkill. *debian-facile.* [En ligne] <https://debian-facile.org/doc:reseau:rkill>.
5. **doc.ubuntu-fr.** configuration de hostapd. *doc.ubuntu-fr.* [En ligne] <https://doc.ubuntu-fr.org/hostapd> .
6. **ubuntu.** configuration du parefeu Ubuntu. *ubuntu.* [En ligne] <https://ubuntu.com/server/docs/security-firewall>.
7. **digital ocean.** NodeJS et npm sur Ubuntu. *digital ocean.* [En ligne] <https://www.digitalocean.com/community/tutorials/how-to-install-node-js-on-ubuntu-20-04-fr> .
8. **tecmint.** installation de Flask sur Ubuntu. *tecmint.* [En ligne] <https://www.tecmint.com/install-flask-in-ubuntu/>.
9. **mongodb.** installation de MongoDB sur Ubuntu. *mongodb.* [En ligne] <https://www.mongodb.com/docs/v4.4/tutorial/install-mongodb-on-ubuntu/> .
10. **ubuntu.** configuration réseau de Ubuntu. *ubuntu.* [En ligne] <https://ubuntu.com/server/docs/network-configuration>.
11. **computingforgeeks.** configuration de dnsmasq. *computingforgeeks.* [En ligne] <https://computingforgeeks.com/install-and-configure-dnsmasq-on-ubuntu/> ,.
12. **ably.** Web Socket avec NodeJS. *ably.* [En ligne] <https://ably.com/blog/web-app-websockets-nodejs> .
13. **raspberrypi.** RRI 4 - Ubuntu + USB SSD ne démarre pas. *raspberrypi forums.* [En ligne] <https://forums.raspberrypi.com/viewtopic.php?t=295609>.



14. **ooreka.** la domotique. *domotique oorek.* [En ligne]  
<https://domotique.ooreka.fr/comprendre/>.
15. **arduino.** Multithread sur Arduino et ESP. *arduino create.* [En ligne]  
<https://create.arduino.cc/projecthub/reanimationxp/how-to-multithread-an-arduino-prototyping-tutorial-dd2c37> .

# TABLE DES MATIERES

<b>DEDICACES .....</b>	<b>I</b>
<b>REMERCIEMENTS .....</b>	<b>II</b>
<b>SOMMAIRE .....</b>	<b>III</b>
<b>RESUME .....</b>	<b>V</b>
<b>GLOSSAIRE .....</b>	<b>VI</b>
<b>LISTE DES FIGURES .....</b>	<b>VIII</b>
<b>LISTE DES TABLEAUX .....</b>	<b>X</b>
<b>LISTE DES PARTICIPANTS AU PROJET .....</b>	<b>XI</b>
<b>INTRODUCTION .....</b>	<b>1</b>
<b>PARTIE 1: PRESENTATION .....</b>	<b>2</b>
<b>1.1. BREVE PRESENTATION DE L'IAI-TOGO .....</b>	<b>3</b>
<b>1.2. PRESENTATION DU CADRE DE STAGE .....</b>	<b>6</b>
1.2.1. STATUT .....	6
1.2.2. MISSION .....	6
1.2.3. ORGANIGRAMME .....	6
1.2.4. SERVICE D'ACCUEIL .....	7
1.2.5. PLAN DE LOCALISATION .....	7
<b>PARTIE 2: ETUDE REALISATION DU PROJET .....</b>	<b>8</b>
<b>CHAPITRE 1: CONTEXTE DE TRAVAIL ET APPROCHE DE SOLUTIONS .....</b>	<b>9</b>
<b>1.1. ÉTATS DES LIEUX .....</b>	<b>10</b>
1.1.1. ARCHITECTURE SYSTEME .....	10
1.1.2. ARCHITECTURE RESEAU (TOPOLOGIE PHYSIQUE ET TOPOLOGIE LOGIQUE) .....	10
<b>1.2. CRITIQUE DE L'EXISTANT .....</b>	<b>11</b>
1.2.1. POINTS FORTS .....	11
1.2.2. POINTS FAIBLES .....	11
<b>1.3. PROBLEMATIQUE .....</b>	<b>12</b>
<b>1.4. INTERET DU SUJET .....</b>	<b>12</b>
1.4.1. OBJECTIFS .....	12
1.4.2. RESULTATS ATTENDUS .....	13

<b>1.5. METHODOLOGIE DE RECHERCHE DE SOLUTION .....</b>	<b>13</b>
<b>CHAPITRE 2: GENERALITES ET DOCUMENTATION .....</b>	<b>14</b>
<b>2.1. SOLUTION DE GESTION D'HABITAT .....</b>	<b>15</b>
2.1.1. HOME ASSISTANT .....	15
2.1.2. DOMOTICZ.....	16
2.1.3. AGO AHOME .....	16
2.1.4. CHOIX DE LA SOLUTION DE GESTION D'HABITAT .....	18
<b>2.2. LE SERVEUR (LA BOX).....</b>	<b>19</b>
2.2.1. ASUS THINKER BOARD .....	19
2.2.2. RASPBERRY PI.....	20
2.2.3. CHOIX DE ORDINATEURS MONOCARTE.....	21
<b>2.3. LES MICROCONTROLEURS SUR CARTE.....</b>	<b>22</b>
2.3.1. LES ESP.....	23
2.3.1.1. ESP8266 .....	23
2.3.1.2. ESP23 .....	24
2.3.2. LES ARDUINO .....	25
2.3.2.1. Arduino Nano .....	25
2.3.2.2. Arduino Méga 2560 .....	26
2.3.3. CHOIX DU MICROCONTROLEUR .....	27
<b>2.4. LES PROTOCOLES DE COMMUNICATION.....</b>	<b>29</b>
2.4.1. <i>Filaire</i> .....	29
2.4.1.1. KNX.....	29
2.4.1.1. CPL .....	29
2.4.2. <i>Sans Fil</i> .....	30
2.4.2.1. Z-Wave.....	30
2.4.2.2. Wi-Fi.....	31
2.4.3. <i>choix du protocole de communication</i> .....	31
<b>2.5. SYNTHESE ET CHOIX DES ELEMENTS DE REALISATION .....</b>	<b>32</b>
<b>CHAPITRE 3: MISE EN ŒUVRE ET PERSPECTIVES.....</b>	<b>34</b>
<b>3.1. LES ELEMENTS DE REALISATION .....</b>	<b>35</b>
3.1.1. <i>Ubuntu server 20.04 LTS ARM</i> .....	35
3.1.2. <i>modprobe</i> .....	36
3.1.3. <i>RFkill</i> .....	36
3.1.4. <i>HostAPD</i> .....	36
3.1.5. <i>Dnsmasq</i> .....	37
3.1.6. <i>Iptables</i> .....	37
3.1.7. <i>netfilter-persistent</i> .....	37

3.1.8.	<i>NodeJS</i>	38
3.1.9.	<i>npm</i>	38
3.1.10.	<i>Python</i>	39
3.1.11.	<i>Flask</i>	39
3.1.12.	<i>MongoDB</i>	40
3.1.13.	<i>les différents capteurs du système</i>	40
3.1.13.1.	Capteur de courant ACS712 20A de AZDelivery	40
3.1.13.2.	Présence HC-SR501	41
3.1.13.3.	Température DHT22	42
<b>3.2.</b>	<b>MISE EN ŒUVRE</b>	<b>43</b>
3.2.1.	<i>Installation du système d'exploitation</i>	43
3.2.1.1.	Préparez la carte SD	43
3.2.1.2.	Configuration réseau	44
3.2.1.3.	Démarrage du serveur et connexion	45
3.2.2.	<i>Configuration de l'interface Wlan0</i>	45
3.2.2.1.	Configuration IP adresse statique	45
3.2.2.2.	Configuration en AP (Access Point)	46
3.2.3.	<i>Configuration du nom de domaine et du DHCP</i>	47
3.2.4.	<i>installation et programmation du serveur socket</i>	48
3.2.4.1.	installation de NodeJS	48
3.2.5.	<i>Le programme</i>	48
3.2.6.	<i>WebSocket</i>	49
3.2.7.	<i>Le datagramme</i>	49
3.2.8.	<i>Installation du SGBD MongoDB</i>	50
3.2.9.	<i>Installation du serveur applicatif</i>	51
3.2.9.1.	Installation de l'API	51
3.2.10.	<i>Programmation des MCUs</i>	52
3.2.10.1.	Environnement Arduino	52
3.2.10.2.	Structure du code des MCU	52
3.2.11.	<i>Création des daemons</i>	53
3.2.12.	<i>Configuration du pare-feu</i>	54
<b>3.3.</b>	<b>TEST DE FONCTIONNEMENT</b>	<b>54</b>
<b>3.4.</b>	<b>EVALUATION FINANCIERE</b>	<b>56</b>
<b>PARTIE 3: GUIDE D'UTILISATION</b>		<b>60</b>
<b>3.1. GUIDE D'UTILISATION POUR LES CLIENTS</b>		<b>61</b>
<b>CONCLUSION</b>		<b>64</b>
<b>BIBLIOGRAPHIE INDICATIVE</b>		<b>65</b>
<b>ANCIENS MEMOIRES CONSULTES</b>		<b>66</b>

NOTES DE COURS .....	67
WEBOGRAPHIE INDICATIVE.....	ERREUR ! SIGNET NON DEFINI.
TABLE DES MATIERES.....	70