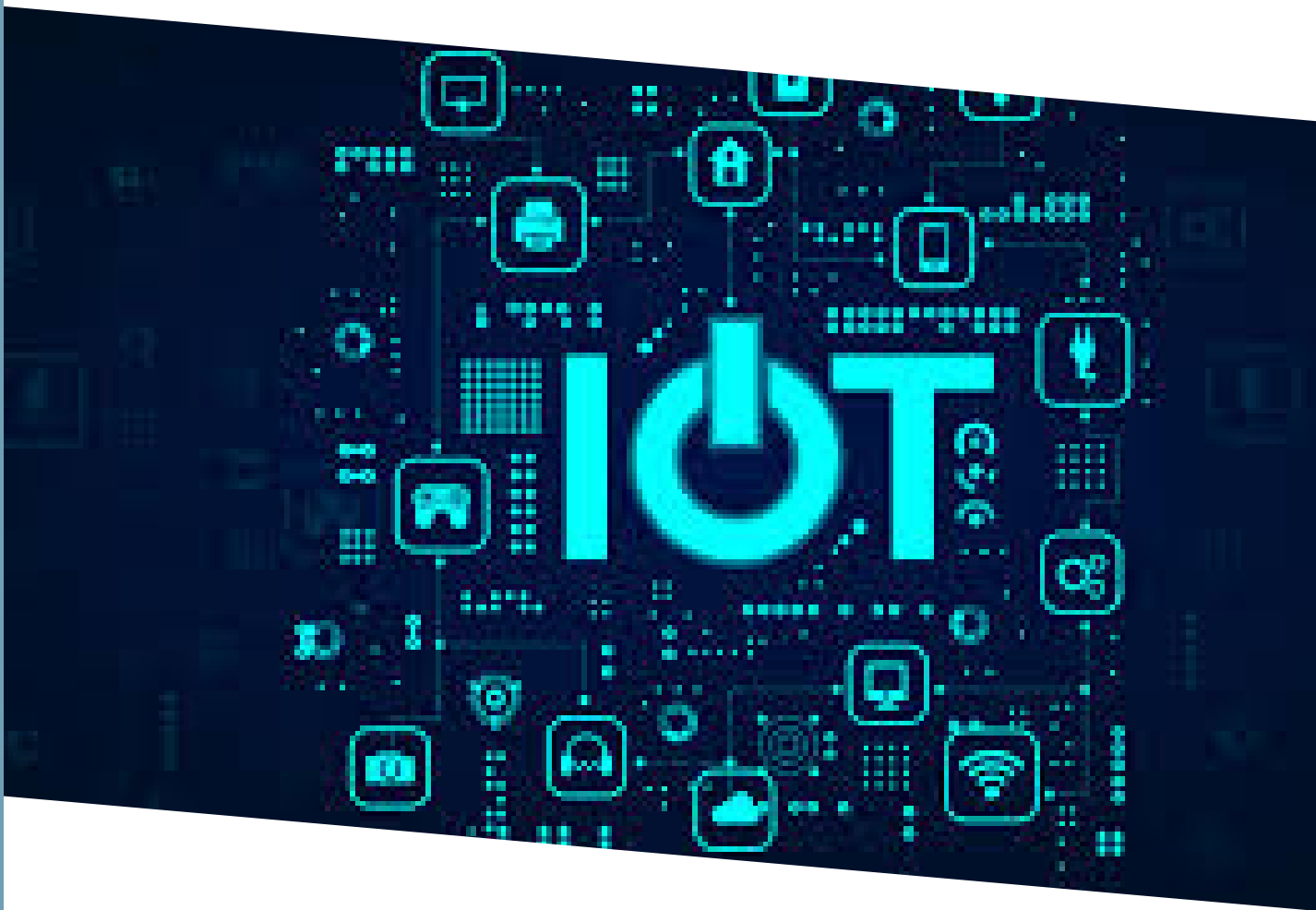




ECOLE NATIONALE SUPERIEURE POLYTECHNIQUE DE YAOUNDE
DEPARTEMENT DU GENIE INFORMATIQUE

UE: Electronique et Interfaçage

EVOLUTION DU PROJET: SIGHT EYES



Phase: Prise en main des modules

Sous la supervision de:

Dr. CHANA Anne Marie

Dr NGOUNOU Guy Merin

Année Académique **2024-2025**



UE: Electronique et Interfaçage

MEMBRES DU GROUPE

- ATABONG Stéphane
- DJOUNKENG Eléonor (Chef)
- FOMEKONG Jonathan
- MBOCK Jean Daniem
- NGAH NDONGO Estelle (Vice-chef)
- NGHOGUE Franck
- NGO BASSOM Anne Rosalie
- NGOUPAYE DJIO Thierry
- NOMO Gabriel
- WANDJI Emmanuel

1. Introduction.....	1
2. Résultats attendus.....	2
3. Processus et méthodologie.....	2
4. Prise en main de chaque module.....	3
4.1. Module alimentation, signalisation et alerte.....	3
4.2. Module central.....	4
4.2.1. Description du module.....	4
4.2.2. Tâches réalisées.....	4
4.2.3. Composants utilisés.....	4
4.2.4. Ressources exploitées.....	5
4.2.5. Résultats obtenus.....	5
4.2.6. Problèmes rencontrés et solutions.....	5
4.2.7. Organisation et contributions.....	6
4.2.8. Conclusion et prochaines étapes.....	6
4.3. Module logiciel.....	6
4.4. Module châssis.....	10

1. Introduction

Le projet en cours consiste à développer un robot capable de reconnaître et de localiser des objets de manière autonome. Ce robot sera constitué de plusieurs modules essentiels qui travailleront de concert pour accomplir cette tâche. Avant de procéder à l'intégration finale des modules dans le robot, il est crucial de bien comprendre le fonctionnement de chacun de ces composants.

Cette phase de validation préliminaire permet d'assurer que chaque module est opérationnel et conforme aux spécifications requises avant leur assemblage. Dans tout projet robotique, l'intégration réussie des différents modules dépend de la compréhension approfondie de chaque composant pris individuellement. Tester et valider chaque module avant leur intégration permet de détecter d'éventuelles erreurs ou incohérences, garantissant ainsi la fiabilité du système global une fois monté.

Dans notre projet, nous avons identifié quatre modules principaux qui nécessitent une exploration détaillée : l'Alimentation, le Module Central, le Logiciel et le Châssis. Chaque module joue un rôle crucial et interagit étroitement avec les autres, ce qui rend indispensable leur validation indépendante pour éviter tout dysfonctionnement à l'étape suivante. L'objectif principal de cette phase est d'explorer et de tester chacun des quatre modules afin de s'assurer de leur bon fonctionnement et de leur adéquation avec les besoins du robot. Concrètement, nous visons à explorer les quatre modules pour comprendre en profondeur le rôle et le fonctionnement de chaque module : l'Alimentation, le Module Central, le Logiciel et le Châssis. Des tests détaillés seront réalisés pour évaluer les performances, l'efficacité et la compatibilité de chaque module. La documentation de ces tests fournira un retour d'information précieux pour l'intégration et la mise en production du robot.

2. Résultats attendus

Au terme de cette phase, les membres de chaque module devaient être en mesure de comprendre le fonctionnement global de leur module conformément aux objectifs qu'ils se sont fixés, manipuler le matériel et tester différents scénarios afin de faciliter la fusion et l'assemblage final du robot.

Pour permettre à toutes l'équipe de comprendre le fonctionnement de tout le matériel utilisé, chaque sous module devra fournir au groupe un petit rapport avec sources de l'utilisation du matériel relatif à leur module.

3. Processus et méthodologie

Afin d'être plus productif, notre système a été subdivisé en 4 modules principaux à savoir:

- **Module Alimentation** : Ce module avait la charge de définir clairement le processus d'alimentation entier de tous les composants constituant le robot.
- **Module Central** : Ce module avait la tâche d'étudier les algorithmes et les formules mathématiques nécessaires pour le repérage des objets dans l'espace.
- **Module Logiciel** : Ce module devait concevoir l'interface permettant la communication entre l'utilisateur final et le robot .
- **Module Châssis** : Ce module avait la charge de comprendre le fonctionnement du kit motorisé et de le monter.

A cet effet, chaque module devait:

- Prendre en main la documentation du matériel concerné par leur module

- Tester les différentes fonctionnalités offertes par le matériel de leur module

4. Prise en main de chaque module

4.1. Module alimentation, signalisation et alerte

- **Tâches**

Dans ce module, il était question de:

- penser la structure de l'alimentation des modules;
- concevoir un système d'alerte sur commande.

- **Ressources utilisées**

Les ressources utilisées sont:

- Comme composants: ESP32-CAM, un buzzer, un boutons, des fils, une LED.
- Comme technologies: Arduino IDE (Environnement de développement), C++ (Langage de programmation).

- **Résultats obtenus**

Après cette période de travail, nous avons pu:

- Obtenir le flux vidéo de l'ESP32-CAM à partir d'un ordinateur;
- Connecter l'ESP32-CAM à un WiFi distant pour l'accès à Internet;
- Réaliser un circuit d'alerte sur commande (bouton pour le test) dont le programme pourra être utilisé dans d'autres modules;
- Concevoir un plan d'alimentation des modules.

- **Difficultés et solutions**

Au cours de cette période de prise en main, la principale difficulté rencontrée a été le téléversement d'un programme dans l'ESP32-CAM.

Cette difficulté a été résolue en appuyant régulièrement lors du téléversement sur le bouton RST de l'ESP32-CAM.

- **Organisation et contribution des membres**

Dans ce module, les membres assignés étaient:

- MBOCK Jean Daniel
- NGOUPAYE DJIO Thierry
- NOMO Gabriel

Les tâches d'alimentation, de montage de circuit buzzer et de prise en main de l'ESP32-CAM ont été faites en présence des 3 membres. L'utilisation de l'ESP32-CAM (téléversement de codes) a été effectué par MBOCK (détenteur de l'ESP32-CAM).

4.2. Module central

4.2.1. Description du module

Le *Module Central* joue un rôle clé dans notre projet, car il constitue le cerveau du robot. Il est responsable de la centralisation des traitements, de la communication entre les autres modules, et de l'exécution des algorithmes d'analyse d'images pour la reconnaissance des objets. Nous avons utilisé un Raspberry Pi pour répondre à ces exigences grâce à ses performances adaptées aux projets d'IoT.

4.2.2. Tâches réalisées

Les travaux réalisés sur ce module se sont déroulés en plusieurs étapes, décrites comme suit :

- **Installation et configuration du Raspberry Pi :**
 - Installation du système d'exploitation Raspberry OS sur une carte microSD.
 - Configuration de la connexion au réseau Wi-Fi pour assurer l'intégration avec les autres modules.
 - Connexion en SSH pour permettre l'accès distant via le terminal.
 - Émulation de l'interface utilisateur du Raspberry Pi sur une autre machine à l'aide de l'application RealVNC Viewer.
- **Exploration des technologies d'intelligence artificielle :**
 - Apprentissage des bases de l'utilisation de Tensor Flow pour l'analyse d'images.
 - Test d'un modèle pré entraîné pour la reconnaissance des objets communs à partir d'une webcam.

4.2.3. Composants utilisés

- **Matériel :**
 - Raspberry Pi 3 Model B (4 Go de RAM).
 - Carte microSD (32 Go).
 - Câble d'alimentation USB-C.
 - Clavier et souris USB (pour la configuration initiale).
 - Moniteur HDMI pour l'installation.
- **Logiciels :**
 - Raspberry OS pour l'installation du système sur la carte raspberry pi
 - Bibliothèques d'apprentissage automatique: Tensor Flow, numpy, openCV
 - RealVNC Viewer (pour l'accès à distance).
 - Python 3 (pour l'exécution des scripts d'apprentissage automatique).

4.2.4. Ressources exploitées

- Documentation officielle de Raspberry Pi pour l'installation et la configuration.
- Tutoriels en ligne pour la mise en place de Tensor Flow sur Raspberry Pi.
- Modèles Tensor Flow pré entraînés téléchargés depuis Tensor Flow Model.
- Guides techniques pour l'utilisation de RealVNC Viewer.

4.2.5. Résultats obtenus

- Installation et configuration complètes du Raspberry Pi avec Raspberry OS.
- Connexion réseau stable assurée via Wi-Fi, accessible en SSH et émulée via RealVNC Viewer.
- Le modèle pré entraîné utilisé a correctement identifié des objets communs comme des bouteilles, des chaises, et des livres à partir des images fournies par la webcam.
- Documentation de toutes les étapes pour faciliter la reproduction des configurations.

4.2.6. Problèmes rencontrés et solutions

- **Problème 1** : Lenteur lors de l'installation de TensorFlow sur Raspberry Pi dû à une connexion Internet instable et lente.
 - **Solution** : Nous avons dû nous déplacer et sortir du campus pour trouver un point d'accès internet stable.
- **Problème 2** : Difficultés lors de l'installation du système sur la carte du raspberry pi.
 - **Solution** : Après de nombreux échecs sur la carte que nous avions payée, nous avons dû nous en procurer une autre.
- **Problème 3** : Erreurs dans l'installation des librairies et exécution des modèles pré entraînés détection d'objets avec le modèle pré entraîné.
 - **Solution** : Ajustement des paramètres du modèle et changement des versions des librairies et bibliothèques utilisées pour assurer les compatibilités.

4.2.7. Organisation et contributions

Les membres responsables de ce module étaient:

- DJOUNKENG NGUEFACK ELéonor
- FOMEKONG TAMDI Jonathan
- NGAH NDONGO Estelle

Pour une meilleure collaboration et organisation, nous nous sommes tous autant impliqués dans chacune des tâches du module qui sont:

- Configuration matérielle et logicielle initiale du Raspberry Pi (Raspberry OS, Wifi, SSH).
- Installation et tests des algorithmes Tensor Flow pour la reconnaissance d'objets.
- Configuration et tests de l'émulation avec RealVNC Viewer.
- Documentation des étapes et solutions rencontrées.

4.2.8. Conclusion et prochaines étapes

Cette phase de prise en main du module central a permis de maîtriser les bases nécessaires à l'exploitation du Raspberry Pi, à la fois comme plateforme de contrôle et comme outil d'analyse d'images. La prochaine étape consistera à intégrer ce module avec les autres (alimentation, châssis et logiciel) pour tester l'ensemble du système en conditions réelles et entraîner notre propre algorithme de détection des objets à partir d'images et flux vidéos envoyés.

4.3. Module logiciel

Ici, il est question de mettre sur pied une interface de contrôle du robot. Concrètement, il s'agit de mettre en place une interface (un bot Telegram ou une Application Web) qui permet d'influencer le comportement du bot. L'interaction avec le bot implique :

- La gestion des flux d'images utilisateur : L'utilisateur doit pouvoir uploader des images dans la base de données pour pouvoir ensuite consulter les images disponibles
- La communication et le transfert d'images au robot : L'utilisateur doit pouvoir envoyer des signaux au robot et transférer au robot une image depuis la base de données

- **Tâches :**

- Créer un bot Telegram
- Ecrire le code Python du bot
- Créer l'application web qui a les mêmes fonctions

- **Ressources utilisées :**

- Un ordinateur
- L'application Telegram
- Node JS
- Navigateur

- **Les résultats obtenus :**
 - Bot Telegram :





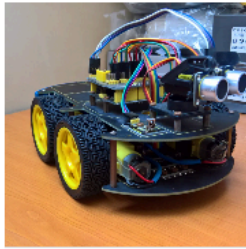
- Application Web :

Gestion des images

Afficher les images

Uploader une image

[Retour au menu](#)



2024-11-09T22:53:19.000Z

[Transférer au bot](#)

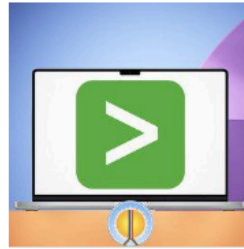
[Voir](#)



2024-11-09T23:09:29.000Z

[Transférer au bot](#)

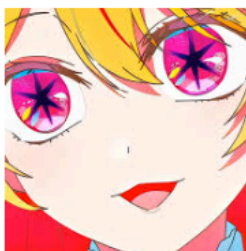
[Voir](#)



2024-11-09T23:17:44.000Z

[Transférer au bot](#)

[Voir](#)



[Retour au menu](#)

Uploader une nouvelle image

[Choisir un fichier](#) Aucun fichier n'a été sélectionné [Uploader](#)

4.4. Module châssis

1.Objectifs

- Monter le kit motorisé du robot Keyestudio avec précision.
- Installer un servomoteur permettant la rotation à 360° d'une caméra ESP32-CAM et d'un capteur ultrasonique.
- Mettre en place un circuit d'alimentation fiable pour l'ensemble des composants à l'aide de piles.

2.composants utilisés

Conformément aux besoins de notre robot et à notre maquette, le module châssis inclut les éléments suivants :

- **Châssis en aluminium** : Structure robuste pour supporter les moteurs et les modules.
- **Roues Mecanum** : Permettant un mouvement omnidirectionnel.
- **Carte Arduino UNO** : Pour gérer les capteurs et modules.
- **Carte de commande moteur L298N** : Utilisée pour contrôler les moteurs du châssis.
- **LED RGB WS2812** : Pour des effets lumineux personnalisés.
- **Capteur ultrasonique** : Détection d'obstacles et suivi.
- **Servomoteur** : Permettant la rotation de la caméra ESP32-CAM et du capteur ultrasonique.
- **Module Wi-Fi ESP32-CAM** : Fournissant la capacité de streaming vidéo et de communication sans fil.
- **Support batterie 18650** : Alimentation du robot (batteries non incluses).
- **Câbles et connecteurs** : Pour les connexions entre modules.
- **Visserie et outils** : Pour assembler le kit.

3.Ressources exploitées

- Documentation officielle de Keyestudio.
- Tutoriels vidéos sur YouTube pour le montage et la programmation.

4.Technologies et logiciels utilisés

- **Langage** : C++
- **Environnement** : Arduino IDE

5. Résultats obtenus

- **Montage du kit motorisé personnalisé**

Intégration réussie de la caméra ESP32-CAM avec un capteur ultrasonique fixé sur un servomoteur rotatif.

- **Alimentation électrique**

Circuit d'alimentation optimisé pour fournir une énergie stable à chaque composant.

- **Tests unitaires**

Validation individuelle du fonctionnement de chaque composant (capteurs, moteurs, ESP32-CAM, etc.).

- **Programmation des fonctionnalités de base**

Commandes Arduino pour avancer, s'arrêter, changer de direction, et faire tourner la caméra.

6. Problèmes rencontrés et solutions

- **Contrôle des roues**

La carte Arduino UNO seule ne peut pas gérer directement les moteurs. De plus, la carte de commande L298N ne peut piloter que deux roues.

Solution : Liaison logique de deux roues diamétralement opposées pour garantir un mouvement fluide et omnidirectionnel sans perturber la stabilité du robot.

7. Organisation et contribution de l'équipe

Dans ce module, les membres assignés étaient:

- ATABONG EFON Stephane Fritz
- WANDJI Emmanuel
- NGO BASSOM ANNE ROSALIE

l'équipe étant réduite à trois membres, chacun a dû s'impliquer dans toutes les étapes :

- Recherche et achat des matériaux.
- Montage physique du kit.
- Implémentation des fonctionnalités.
- Rédaction des rapports techniques.

8. Conclusion et prochaines étapes

Au terme de cette phase , nous pouvons constater que chaque module a pu effectuer entièrement ou en partie les tâches qui lui avaient été assignées. Malgré les bugs et les erreurs auxquels nous avons fait face, nous avons pu d'une manière toucher du doigt le matériel et comprendre son fonctionnement.

La prochaine étape consiste maintenant à définir une architecture de communication optimale des composants et débiter la fusion des modules.