

Projet Transversal

Hugues Farthouat – Pierre Gosson – Reda Laborieux

Capucine Jumelle – Antoine Kleitz – Alexandre Vernet

Rapport intermédiaire du jalon 1 et 2



**NERVO
BOT**

SOMMAIRE

Chapitre 1 - Présentation de l'équipe _____ 2

Chapitre 2 - Présentation de l'objectif global retenu _____ 3

JALON 1

Chapitre 3 - Alexandre Vernet - Gestion microcontrôleur _____ 5

Chapitre 4 - Reda Laborieux - PC/Smartphone vers Internet/site web via liaison Wifi _____ 5

Chapitre 5 - Antoine Kleitz - Mise en place du retour vidéo sur le site web _____ 6

Chapitre 6 - Hugues Farthouat - Conception du site web _____ 6

Chapitre 7 - Pierre Gosson - Mise en place Raspberry et environnement de travail commun
Github _____ 6

Chapitre 8 - Capucine Jumelle- Détection et suivi de point rouge avec OpenCV dans une
vidéo en temps réel + Détection du son _____ 7

JALON 2

Chapitre 9 - Alexandre Vernet - Codage stm32 et communication _____ 8

Chapitre 10 - Reda Laborieux - Réalisation d'un pare-feu pour sécuriser le site web du
robot _____ 8

Chapitre 11 - Antoine Kleitz - Configuration du mode 3 du robot _____ 8

Chapitre 12 - Hugues Farthouat - Architecture server _____ 9

Chapitre 13 - Pierre Gosson - Mise en commun raspberry stm32 _____ 9

Chapitre 14 - Capucine Jumelle- Amélioration du suivi de point rouge et détection d'un son
“anormal” dans un bruit ambiant _____ 9

Chapitre 15 - Intégration des sous-ensembles - Fonctionnalités _____ 10

CHAPITRE 1 - PRÉSENTATION DE L'ÉQUIPE

Notre équipe est composée de 6 membres, issus de différentes majeures. Chaque membre à une responsabilité bien définie dans le projet.

Hugues Farthouat – Responsable Communication

Il sera responsable de la bonne communication entre tous les membres du groupe. Il fait directement le lien entre le tuteur et l'équipe.

Pierre Gosson – Responsable Planificateur

Il sera chargé de planifier les différentes tâches à effectuer dans l'équipe. Il s'assure que chaque membre effectue les tâches qui lui sont assignées dans le temps qui lui est donné.

Capucine Jumelle – Responsable Logistique

Elle sera chargée de s'assurer du respect du matériel. Elle veillera à faire l'inventaire du matériel mis à notre disposition à chaque séance pour éviter toute perte ou dégradation. De plus, elle s'assurera tout au long de notre projet que les consignes et le cahier des charges soient bien respectés.

Antoine Kleitz – Gestionnaire des rendus

Il s'assurera que les rendus soient rendus dans le temps donné. De plus, il veillera à la bonne qualité de chaque rendu (fin de séance, rapports, etc...) et des différents documents produits.

Reda Laborieux – Animateur du projet

Ce rôle est très important, il sera chargé de veiller au bien-être des membres au sein du groupe de travail. Il devra s'assurer que chaque membre trouve sa place dans le groupe tout au long du projet. De plus, il devra coordonner les différentes tâches pour que chaque tâche fasse avancer les autres.

Alexandre Vernet – Responsable Qualité-Test

Il sera chargé de valider le bon fonctionnement des différentes fonctions ajoutées au robot avant la fin de chaque jalon.

Hugues FARTHOUAT	Le responsable Communication
Pierre GOSSON	Le responsable Planificateur
Capucine JUMELLE	Le responsable Logistique & Le responsable "Respects des Consignes"
Antoine KLEITZ	Le responsable – Gestionnaire des rendus.
Réda LABORIEUX	L'animateur du projet
Alexandre VERNET	Le responsable Qualité – Tests – Aspects contrôle technique

Pour assurer une bonne gestion de notre projet, nous avons mis en place deux outils collaboratifs : GitHub et Google Drive. Nous utilisons Google Drive comme espace de stockage commun pour les comptes rendus de réunion et l'élaboration de nos rapports. Tous les membres de l'équipe peuvent y contribuer et travailler en collaboration.

Quant à GitHub, c'est là que la majeure partie de notre projet est organisée. Tous les membres de l'équipe y ont accès et peuvent y contribuer en proposant des solutions à des problèmes spécifiques, comme par exemple la détection de points rouges sur une caméra, qui a été attribuée à Capucine pour le Jalon 1.

Grâce à GitHub, nous pouvons facilement suivre l'avancement de chaque tâche, assigner des délais et des responsabilités à chaque membre de l'équipe, et collaborer efficacement sur le code source du projet. En utilisant ces deux outils de manière complémentaire, nous sommes en mesure de travailler en collaboration pour atteindre nos objectifs dans les délais impartis.

CHAPITRE 2 - PRÉSENTATION DE L'OBJECTIF GLOBAL RETENU

L'objectif global de ce projet est de concevoir un robot mobile « semi-autonome » qui réalise certaines fonctionnalités. Le robot sera capable de se déplacer dans un espace défini. Notre robot proposera plusieurs modes de fonctionnement différents.

Mode 1

L'intérêt de ce mode est de détecter une couleur précise et de la suivre si elle est en mouvement. Si la zone en couleur s'arrête, le robot doit s'arrêter. De même, si elle tourne et change de direction, le robot devra faire de même.

Mode 2

Le mode 2 proposera l'utilisation de ses microphones pour détecter l'intensité d'un bruit bien précis. Quand le robot détecte ce son, il le localise et s'éloigne de son point d'émission.

Mode 3

Le mode 3 est le mode perroquet, le robot détecte un son via son microphone et répète en boucle la phrase enregistrée.

Axes d'amélioration :

Pour le mode 2, on pourrait mettre en place un mode 'Courageux' et un mode 'Peureux'. Dans le premier mode, au moment de détecter le bruit, il s'en approcherait. A l'inverse, dans le mode 'Peureux', il s'éloignerait du bruit. Pour le site, on pourrait l'adapter à l'utilisation sur smartphone. Enfin, on pourrait utiliser un capteur de luminosité pour passer d'un mode 'Jour' à un mode 'Nuit'. Le mode 'Jour' étant le mode « de base », le mode 'Nuit' permettrait l'allumage d'une LED, et le retour automatique du robot à la position initiale.

On peut résumer le système dans le schéma suivant :

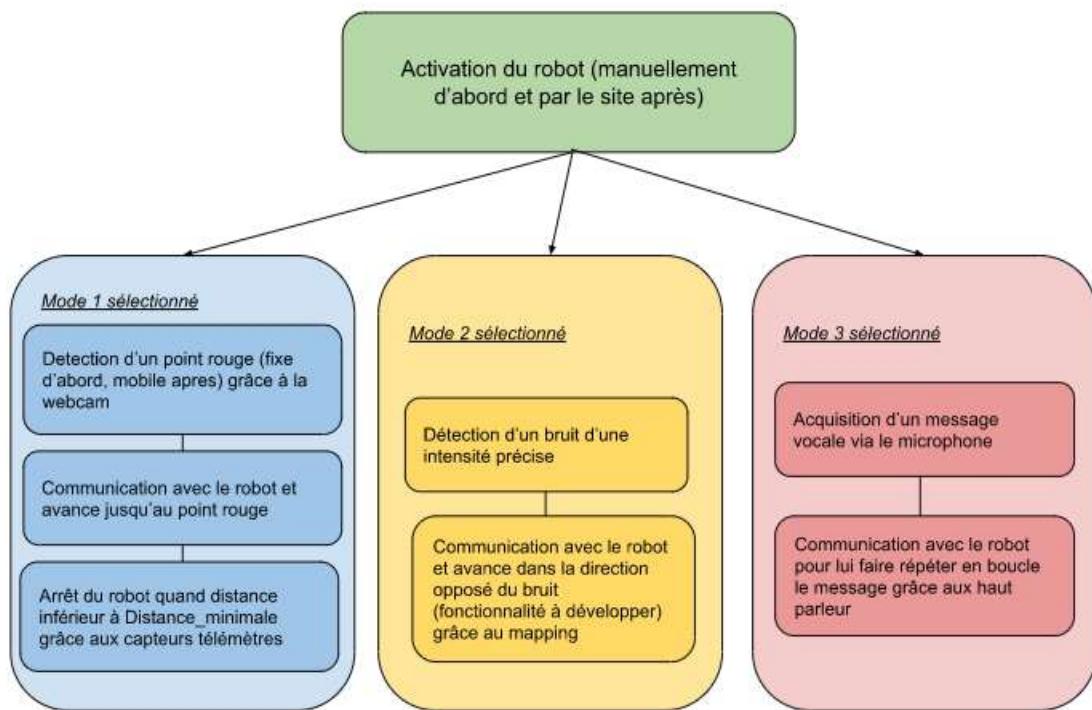


Schéma de synthèse du système complet

Tâches techniques du Jalon 2 :

- Antoine s'occupera de la détection du message vocal et de la communication avec le robot pour le faire répéter le son souhaité.
- Hugues s'occupera d'implémenter les fonctions manquantes au site: sélection du mode, activation/désactivation en temps réel de la détection de point rouge.
- Reda s'occupera d'optimiser la sécurité du site internet et son accessibilité
- Alexandre s'occupera du pilotage des servomoteurs, du calibrage des périphériques sonores et de l'intégration générale.
- Pierre s'occupera de finir l'hébergement du site et la bonne mise en place et communication des différents code des robots.
- Capucine s'occupera de la détection de son afin que le robot puisse détecter d'où vient le bruit et de le fuire.

JALON 1

CHAPITRE 3 - ALEXANDRE VERNET - GESTION MICROCONTRÔLEUR

Lors de la première séance nous avons eu un cours de prise en main du microcontrôleur qui a duré une bonne partie de la séance. Ensuite lors de la séance qui a suivi, j'ai commencé à m'intéresser à la carte Serializer pour les roues et j'ai élaboré un script destiné au 8051 pour permettre d'actionner la carte Serializer pour mettre en marche et arrêter les roues. La séance suivante j'ai continué à travailler sur le script du microcontrôleur pour atteindre l'objectif qui était de piloter le robot (marche/arrêt) à l'aide d'un bouton poussoir. On a découvert que les deux roues ne tournent pas à la même vitesse à commande égale. Le calibrage des roues/du moteur reste à faire pour obtenir une vitesse optimale. J'ai donc attaqué la prise en main d'une carte STM32 pour établir une communication entre cette dernière et le 8051 à l'aide d'une liaison UART. Cette carte servira de cerveau pour les microcontrôleurs et tout devrait passer par elle. Lors de la dernière séance la communication a été établie et le 8051 et le STM32 peuvent être reliés et des données peuvent être envoyées sur le STM32 pour qu'il puisse communiquer avec le 8051.

CHAPITRE 4 - REDA LABORIEUX - PC/SMARTPHONE VERS INTERNET/SITE WEB VIA LIAISON WIFI

Le PC (équipé d'une carte réseau Wi-Fi) ou le smartphone saisira l'adresse IP du Raspberry Pi dans sa barre de recherche. Préalablement, le Raspberry Pi aura enregistré dans sa base de données les différentes adresses IP autorisées à accéder au site via le framework Flask et la bibliothèque Flask-HTTPAuth.

J'ai aussi travaillé sur le design du site, la gestion des sorties du Raspberry Pi et la sécurisation de la connexion. Mes contributions incluent :

- Création du design du site, en sélectionnant les images, agençant le contenu, et en utilisant une IA pour concevoir le logo. J'ai également aidé au codage du site.
- Gestion des sorties du Raspberry Pi en établissant une communication avec le site. J'ai d'abord utilisé les ports GPIO, mais en raison d'une erreur, j'ai dû utiliser le port série.
- Sécurisation de la connexion entre le PC et le site hébergé sur le Raspberry Pi. J'ai choisi la méthode de création d'un réseau local ad-hoc pour sa simplicité et pertinence.
- Apprentissage de Flask et implémentation du code pour le projet, y compris la création d'une liste d'adresses IP autorisées pour assurer la sécurité du site contrôlant le robot. Je rencontre actuellement un problème qui m'empêche de faire fonctionner mon code sur le Raspberry Pi.

CHAPITRE 5 - ANTOINE KLEITZ - MISE EN PLACE DU RETOUR VIDÉO SUR LE SITE WEB

Au début du projet, mon premier objectif était de participer à la schématisation de l'architecture globale du projet, et notamment réfléchir aux solutions techniques envisagées pour répondre à notre cahier des charges.

Ensuite, j'ai participé à la prise en main du LIDAR. Cependant, nous nous sommes rendu compte que la mise en place du LIDAR était difficile et n'était pas primordiale, nous voulions d'abord configurer les fonctions de base du robot (déplacements, interface WEB).

Je me suis donc chargé de configurer la fonction d'affichage vidéo sur le site Web, on souhaite mettre en place la récupération du flux vidéo de la caméra du robot et l'afficher en direct dans un onglet du site web. Pour cela, j'ai utilisé Python et notamment Flask, un framework open-source qui m'a permis de récupérer le flux vidéo. Il y avait une petite partie de langage HTML pour relier Flask au site web.

Une fois que cette fonctionnalité était fonctionnelle, j'ai commencé à travailler sur la configuration de la commande PWM des servo-moteurs sur la STM32.

CHAPITRE 6 - HUGUES FARTHOUAT - CONCEPTION DU SITE WEB

Nous avons réalisé un site web, hébergé sur la Raspberry Pi. Pour le moment il est accessible depuis n'importe quel ordinateur sur le même réseau privé. Pour le moment le site permet de récupérer les entrées utilisateur (manette) et est capable de retransmettre au stm32 ou directement au robot les informations traitées.

CHAPITRE 7 - PIERRE GOSSON - MISE EN PLACE RASPBERRY ET ENVIRONNEMENT DE TRAVAIL COMMUN GITHUB

La Raspberry Pi est un micro-ordinateur qui héberge notre site internet communiquant avec le robot, il stocke également les codes complexes tel que la détection de point rouge, par la suite il permettra la gestion des différents programmes et leur communication.

Lors de la mise en place du système d'exploitation de la Raspberry Pi 3 B, mon choix s'est tourné vers une interface en ligne de commande pour s'assurer de ne pas perdre de performance lors du fonctionnement du robot.

J'ai également mis en place un projet Github pour centraliser nos différents codes et documents, Github permet également de mettre à jour en temps réel nos codes lorsque différentes personnes travaillent sur le même code. Le projet permet également d'avoir un point de vue global sur l'avancement du projet, mais aussi d'avoir précisément les problèmes et travaux en cours de traitement, à traiter ou encore déjà finis.

Github permet également de pouvoir facilement ajouter sur le Raspberry Pi 3 B les codes que nous avons maniés sur nos machines personnelles.

CHAPITRE 8 - CAPUCINE JUMELLE- DÉTECTION ET SUIVI DE POINT ROUGE AVEC OPENCV DANS UNE VIDÉO EN TEMPS RÉEL + DÉTECTION DU SON

Dans le cadre du jalon 1, j'ai développé un algorithme de détection de point rouge pour permettre au robot de suivre une personne portant une pastille rouge. J'ai utilisé la bibliothèque OpenCV de Python pour effectuer une détection d'objet en temps réel à partir d'une vidéo. Mon code détecte les pixels rouges, trouve les contours des objets et calcule les mouvements nécessaires pour suivre l'objet. Les coordonnées de ces mouvements sont affichées dans le terminal et indiquent au robot dans quelle direction se tourner pour avoir la pastille rouge au centre de la caméra. Voici un exemple de ce que mon code permet d'afficher.



De plus, ayant fini cet algorithme, j'ai revu mes objectifs et ai décidé d'avancer sur la détection de son. Mon code réalisé une analyse spectrale d'un signal audio enregistré à l'aide d'un

microphone en utilisant les bibliothèques Python "sounddevice" et "scipy.signal". Il enregistre cinq signaux audio de durée prédéfinie et calcule le spectre moyen de chaque signal en ne gardant que les fréquences spécifiées. Il détecte le bruit en comparant la valeur maximale du spectre moyen à un seuil défini et stocke la valeur maximale de chaque enregistrement dans une liste. Si le bruit augmente entre le premier et le deuxième enregistrement, un message est affiché. Le code permet ainsi de détecter la présence de bruit dans le signal enregistré et de mesurer son niveau. Cet algorithme permet d'avoir une ébauche de la détection de son auquel je me consacrerai plus spécifiquement au jalon 2.

JALON 2

CHAPITRE 9 - ALEXANDRE VERNET - CODAGE STM32 ET COMMUNICATION

Dans le cadre du jalon 2, j'ai codé entièrement le stm32. Ce stm32 sera utilisé pour communiquer avec la carte serializer et avec les servomoteurs. Ce code a été réécrit plusieurs fois pour satisfaire les besoins du projet mais il reste à parfaire la communication entre la Raspberry et le stm32 ainsi que la communication avec les servomoteurs.

CHAPITRE 10 - REDA LABORIEUX - RÉALISATION D'UN PARE-FEU POUR SÉCURISER LE SITE WEB DU ROBOT

Dans le cadre du projet de robot, j'ai travaillé sur la sécurité du site web qui permet de contrôler le robot à distance. Pour ce faire, j'ai mis en place un pare-feu avec plusieurs fonctionnalités. Tout d'abord, seules les adresses IP présentes dans la liste blanche peuvent accéder au site web. Ensuite, si un utilisateur tente d'accéder à des fonctionnalités spécifiques, telles que la caméra, une requête de connexion avec un nom d'utilisateur et un mot de passe est requise. Si un utilisateur échoue à se connecter plus de sept fois de suite, son adresse IP est bloquée, ce qui empêche les attaques par force brute. Enfin, si une personne tente de se connecter en moins de cinq secondes, il est considéré comme un bot et son adresse IP est bloquée et supprimée de la liste blanche. Grâce à ces mesures, le site web du robot est bien protégé contre les accès non autorisés.

CHAPITRE 11 - ANTOINE KLEITZ - CONFIGURATION DU MODE 3 DU ROBOT

Au début du jalon 2, mon objectif principal est de configurer le mode 3 du robot, à savoir Enregistrement d'un audio et le répéter en boucle (mode Perroquet). Dans un premier temps, je me suis tourné vers la configuration du micro WM62A fourni. Grâce à la fiche technique de mise en œuvre du microphone sur l'e-campus, j'ai câblé un montage amplificateur. J'ai rencontré de nombreux problèmes pour pouvoir mesurer un signal sonore vu par le micro. Par manque de temps, j'ai donc décidé d'utiliser le micro de la caméra plutôt que le WM62A.

Grâce aux librairies Python, je peux en quelques lignes utiliser le micro de la caméra pour enregistrer un signal sonore et le répéter en boucle. Dans un second temps, j'ai dû configurer l'amplificateur du haut-parleur du robot. Pour cela, j'ai réalisé le câblage d'un circuit amplificateur en tension et en courant (le haut-parleur a besoin de beaucoup de puissance). Le circuit fonctionne

bien, mais au moment de l'intégration avec le robot nous avons rencontré des problèmes de code. En fonction de l'OS, les librairies python peuvent varier et nous avons codé sous Windows certains de nos scripts, or la Raspberry utilise Linux. Après avoir réglé ces problèmes d'adaptation, nous pourrons envisager assez rapidement de finir l'intégration du mode 3.

CHAPITRE 12 - HUGUES FARTHOUAT - ARCHITECTURE SERVER

Durant le jalon 2 j'ai essentiellement restructuré le code python sur le serveur-Raspeberry. On a maintenant 3 process tournant indépendamment: flask (web server), robot controller (chargé de communiquer avec le robot, simule ou estime sa position), camera (traitement de la caméra). Ces process discutent à travers des variables partagées. Le site a été amélioré, notamment avec la possibilité de modifier certaines config du programme depuis le site, ajout des modes fonctionnels. J'ai également travaillé sur le lidar, continent à tester et me documenter sur les algorithmes SLAM (Simultaneous Localization And Mapping)

CHAPITRE 13 - PIERRE GOSSON - MISE EN COMMUN RASPBERRY STM32

Pour le jalon 2, la mise en place de la Raspberry Pi 3 B est finie, l'interface finale est en ligne de commande : Raspberry Pi OS Lite (32-bit).

Pendant l'évolution du projet, j'ai géré les librairies et la bonne implémentation des codes sur la Raspberry, en passant par un repository Github. Je me suis également occupé de mettre à jour notre projet Github avec la mise en place des issues (recommandé par notre tuteur).

J'ai participé à l'hébergement du site web sous flask.

J'ai débuté la mise en place d'une communication entre la Raspberry et le STM32, de mon côté j'implémente le script Python à travers la librairie Serial pour communiquer par UART.

CHAPITRE 14 - CAPUCINE JUMELLE- AMÉLIORATION DU SUIVI DE POINT ROUGE ET DÉTECTION D'UN SON “ANORMAL” DANS UN BRUIT AMBIANT

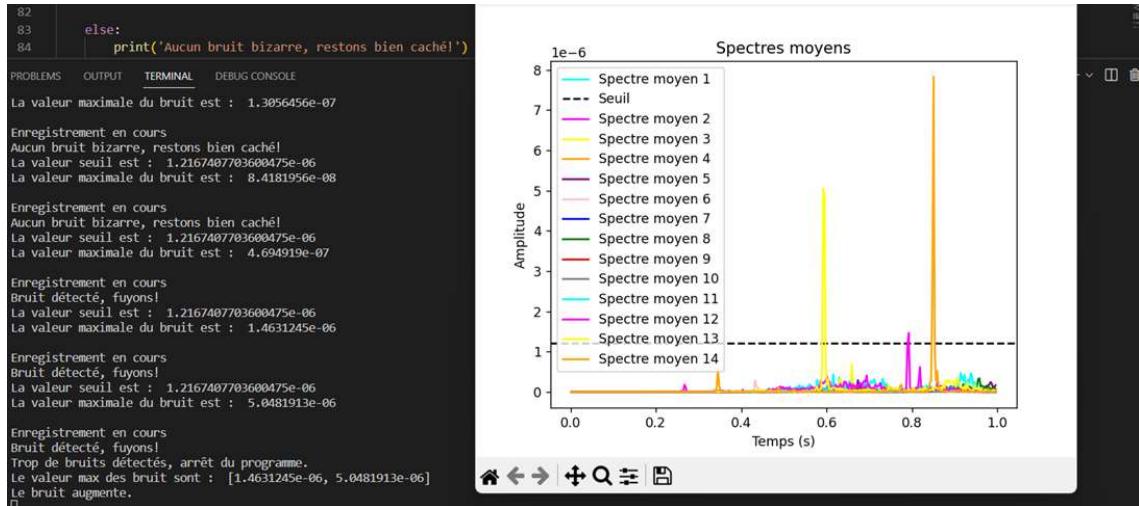
Pendant la période entre les jalons 1 et 2 du projet de tronc commun, j'ai travaillé sur deux projets en parallèle : la détection de points rouges pour le mode 1 du robot joueur, et la détection de sons pour le mode 2.

J'ai consacré 2 heures à chaque projet à chaque séance.

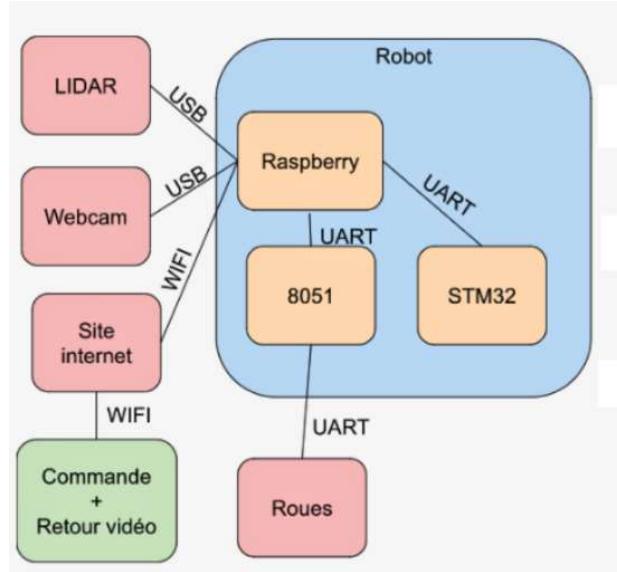
Pour la détection de points rouges, j'ai amélioré le code en ajoutant une fonction pour choisir la couleur à suivre. Je me suis aussi penchée sur la détection d'une forme précise en plus de la couleur mais cette dernière mise en œuvre est assez complexe et ne fonctionne pas pour le moment.

Pour la détection de sons, j'ai réalisé un code qui analyse le bruit ambiant de la salle et détecte tout bruit considéré comme "anormal" grâce au micro de la caméra. Le code utilise une

transformée de Fourier à court terme sur plusieurs enregistrements, et tourne en boucle tant qu'il n'y a pas deux bruits "anormaux" consécutifs.



CHAPITRE 15 - INTÉGRATION DES SOUS-ENSEMBLES - FONCTIONNALITÉS



Le test de l'intégration des sous-ensemble se fait tout d'abord sur machine personnelle pour vérifier le fonctionnement du programme. Par la suite, le programme est testé sur le système d'exploitation correspondant : Raspberry Pi 3 B, STM32, etc ...

L'ensemble des test et validation et leur avancement sont présent sur notre projet Github : <https://github.com/users/PierreCPE/projects/3>