
Prise en main du projet

Projet de fin d'études : Remote Pi

Auteurs :

[Axel PAPE](#)
[Misha SETTA](#)
[Quentin SERENO](#)
[Alexandre SANTOS GOMES](#)
[Nassim SI-MOHAMMED](#)
[Maxime PERES](#)

Auteur du sujet :

[M. LARDEUX Geoffroy](#)

Tuteur :

[M. MOITEL Sébastien](#)

Nous certifions l'originalité de ce travail, affirmant qu'il résulte d'une collaboration au sein de l'équipe et qu'il a été rédigé de manière autonome.

Table des matières

1. INTRODUCTION	3
2. VUE D'ENSEMBLE DU PROJET	4
2.1 CREATION D'UN COMPTE	4
2.2 TELEVERSEMENT D'UN CODE VERS UN ROBOT	5
2.3 UTILISATION DU TEMPLATE.....	6
2.4 EXECUTION DU PROGRAMME	6
3. INSTALLATION DE L'ENVIRONNEMENT DE DEVELOPPEMENT	8
3.1 INSTALLATION DE L'ENVIRONNEMENT ARDUINO	8
3.1.1 <i>Installation des librairies</i>	8
3.2 INSTALLATION DE LA CARTE RASPBERRY	10
3.2.1 <i>Matériel nécessaire</i>	10
3.2.2 <i>Installation de Raspberry Pi Imager</i>	10
3.2.3 <i>Formatage de la carte Micro SD</i>	10
3.3 CONFIGURATION DE LA CARTE RASPBERRY.....	12
3.3.1 <i>Connexion à la Raspberry Pi via SSH :</i>	12
3.3.2 <i>Configuration initiale via SSH :</i>	13
3.3.3 <i>Installation de Arduino cli via SSH</i>	13
3.3.4 <i>Installation des outils ESP8266 pour Arduino CLI</i>	14
3.4 CONFIGURATION DU ROBOT	16
3.5 TEST DE TELEVERSEMENT	16
3.5.1 <i>Prérequis</i>	16
3.5.2 <i>Envoi d'un code compilé</i>	17
3.5.3 <i>Envoi d'un code non compilé</i>	19
3.6 MISE EN PLACE DU GIT.....	19
3.7 MISE EN PLACE DU BACKEND	20
3.7.1 <i>Téléchargement local</i>	20
3.7.2 <i>Backend</i>	21
3.7.3 <i>Interface web :</i>	21
3.7.4 <i>Configuration de l'interface web :</i>	22
3.7.5 <i>Tests (optionnel) :</i>	22
3.7.6 <i>Documentation :</i>	22
4. STRUCTURE DU CODE SOURCE	23
4.1 STRUCTURE DU CODE DE L'INTERFACE.....	23
4.2 MODELE DE CONCEPTION :	23
4.2.1 <i>Rendering Côté Serveur (SSR):</i>	23
4.2.2 <i>Génération de Site Statique (SSG) :</i>	24
4.2.3 <i>File System Based Routing :</i>	24
4.2.4 <i>API Routes :</i>	24
5. BONNES PRATIQUES DE DEVELOPPEMENT.....	25
6. REFERENCES UTILES	26
7. CONTACTS ET SUPPORT	27
8. CONCLUSION	28

1. Introduction

Le projet RemotePi représente une réponse innovante et pertinente à l'évolution constante du domaine de la robotique. Il s'agit d'une plateforme web spécialement conçue pour vous permettre le contrôle à distance de robots à l'aide d'un Raspberry Pi. En se positionnant comme un outil facilitant l'apprentissage de la programmation et de la manipulation robotique, RemotePi vise à rendre ces compétences accessibles à un large public, en particulier aux étudiants. Son interface conviviale et sécurisée vise à démocratiser l'accès à ces technologies en vous proposant une expérience d'apprentissage intuitive et engageante.

L'idée de RemotePi a émergé en réponse à une demande spécifique émanant d'un enseignant de l'ECE Paris, qui avait identifié un besoin concret lors de la compétition annuelle ECE CUP. Cette compétition met en compétition des robots à différents niveaux de contrôle, allant de la télécommande à l'autonomie complète. L'objectif principal de RemotePi est de vous fournir une solution pratique, en vous permettant de programmer et de contrôler à distance vos robots de manière intuitive et sécurisée.

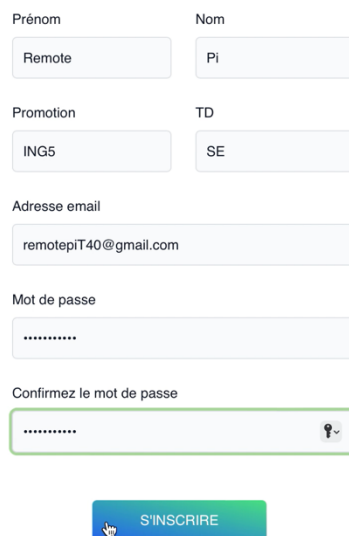
Ce document représente le guide de prise en main du projet remotePi. Il vous expliquera en détail comment RemotePi fonctionne, ainsi que toutes les installations nécessaires à son bon fonctionnement.

Le lien du projet sur github : https://github.com/ElmoAlreadyTaken/PFE_RemotePi.git

2. Vue d'ensemble du projet

2.1 Création d'un compte

Pour utiliser RemotePi, vous devez simplement vous rendre sur l'interface accessible via le lien : <https://pfe-remote-pi.vercel.app/>. Vous pouvez alors créer un compte, dont la validation se fait par le biais d'un lien envoyé par e-mail.



Registration form for RemotePi. The form includes fields for Prénom (First Name), Nom (Last Name), Promotion, TD (Tutorial), Adresse email (Email), Mot de passe (Password), and Confirmez le mot de passe (Confirm password). The Prénom field contains 'Remote' and the Nom field contains 'Pi'. The Promotion field contains 'ING5' and the TD field contains 'SE'. The Adresse email field contains 'remotepi740@gmail.com'. The Mot de passe field is masked with dots. The Confirmez le mot de passe field is also masked with dots and has a toggle icon. A green 'S'INSCRIRE' button is at the bottom.

Figure 1. Inscription sur l'interface remotePi

Une fois votre compte validé, vous devez attendre l'approbation de vos accès par un administrateur. Dès lors cette étape franchie, vous pouvez accéder à l'interface de programmation des robots par l'onglet « TELEVERSER ».

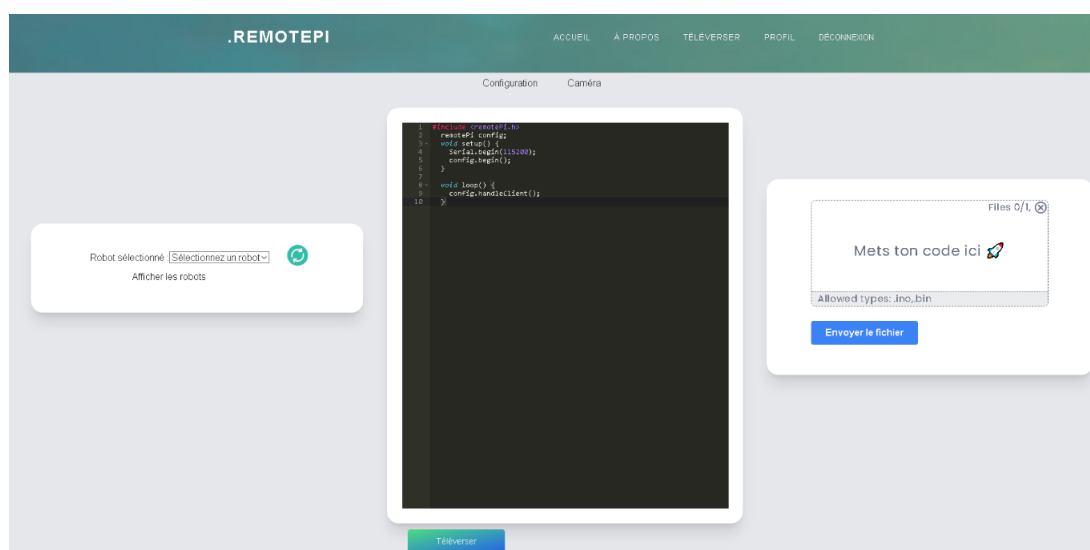


Figure 2. Page de téléversement remotePi

2.2 Téléversement d'un code vers un robot

Lorsque vous vous trouvez sur la page « TELEVERSER », il vous est possible de choisir le robot que vous souhaitez contrôler parmi la liste des robots disponible. Pour envoyer du code au robot sélectionné, deux options s'offrent à vous : soit envoyer un code non compilé via l'IDE intégré, soit compiler le code Arduino localement et envoyer le fichier binaire résultant. L'envoi du code se fait simplement en faisant glisser le fichier à droite de la fenêtre.



Figure 3. Sélection du robot souhaité

Si vous souhaitez envoyer un code non compilé (cela peut arriver si vous n'avez pas un environnement de développement Arduino adéquat), Il vous sera obligatoire d'utiliser le template fourni avec le projet pour garantir la compatibilité et le bon fonctionnement du code. En cas d'erreur de code envoyer, un message d'erreur est affiché pour vous guider dans le processus de débogage.

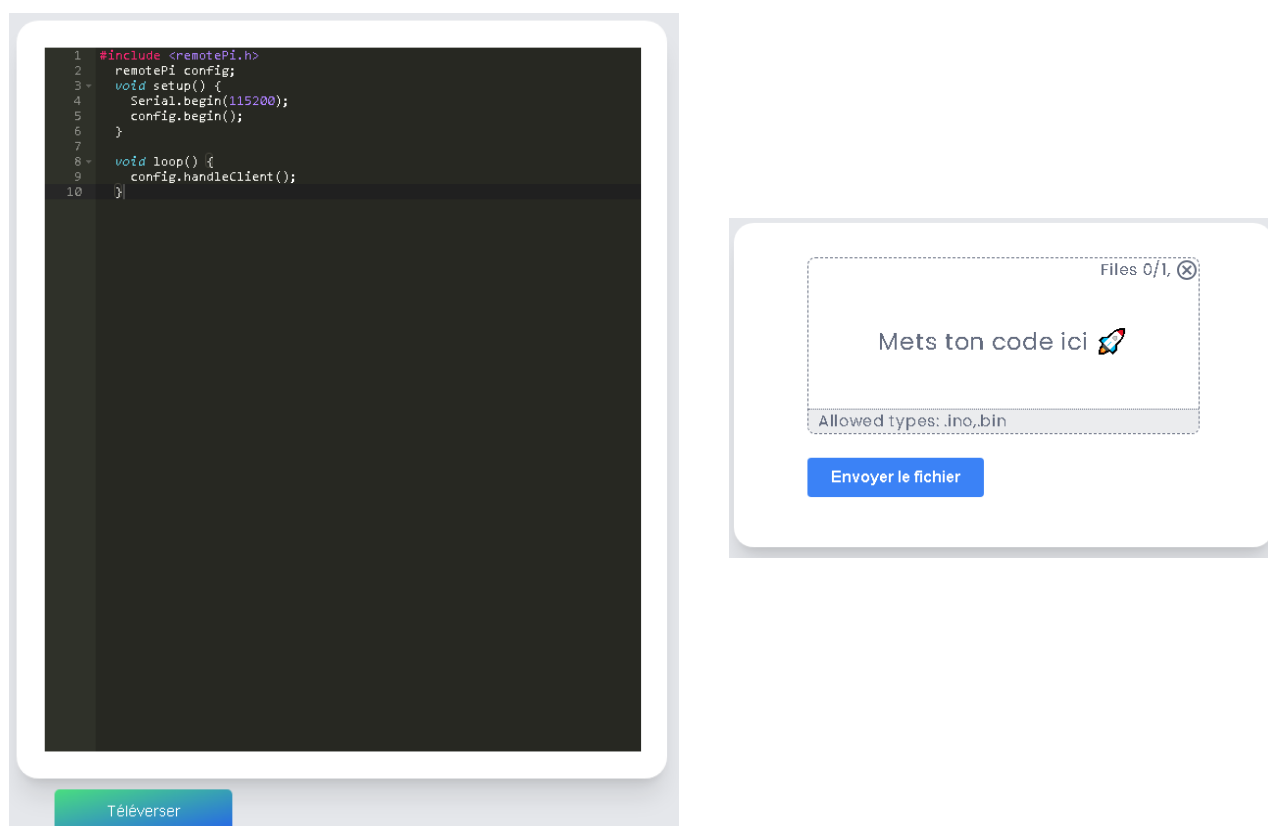


Figure 4. Possibilités d'envoi de code aux robots

2.3 Utilisation du template

Le template "remotePi.ino" est important pour le fonctionnement du projet RemotePi. Il s'agit d'un modèle de code préconfiguré qui établit les bases nécessaires pour que votre robot communique avec la plateforme RemotePi. En gros, ce modèle comprend deux parties principales.

Premièrement, il inclut une bibliothèque appelée "remotePi.h", ce qui permet à votre code d'utiliser les fonctionnalités spéciales de RemotePi. Ensuite, il initialise une connexion avec le serveur RemotePi et configure les paramètres de base.

Dans la partie "setup()" du modèle, nous initialisons la communication série et démarrons la configuration du robot. Puis, dans la partie "loop()", nous maintenons une communication continue avec le serveur RemotePi pour gérer les instructions entrantes.

Il est crucial de garder ce modèle car il garantit une configuration uniforme et stable de votre code sur le robot. En gardant une structure cohérente, nous nous assurons que le robot fonctionne bien avec RemotePi sans rencontrer de problèmes de compatibilité.

En ce qui concerne l'écriture des logs de retour, la librairie RemotePi fournit une méthode simple appelée "sendLog(String message)" pour envoyer des messages de log au serveur RemotePi. Ces messages peuvent contenir des informations sur le fonctionnement du robot ou des erreurs rencontrées. Ces logs sont ensuite accessibles via l'interface de RemotePi pour faciliter le suivi et le débogage.

```
1. #include <remotePi.h>
2. remotePi config;
3. void setup() {
4.     Serial.begin(115200);
5.     config.begin();
6. }
7.
8. void loop() {
9.     config.handleClient();
10. }
11.
```

2.4 Exécution du programme

Une fois le code envoyé au robot, vous pouvez suivre l'exécution du programme dans la fenêtre dédiée, où sont affichés les retours de la table où se trouve le robot ainsi que les logs éventuellement ajoutés dans le code. À la fin de l'exécution, le robot passe à l'état libre et vous pouvez observer le résultat de votre programme dans l'onglet dédié.

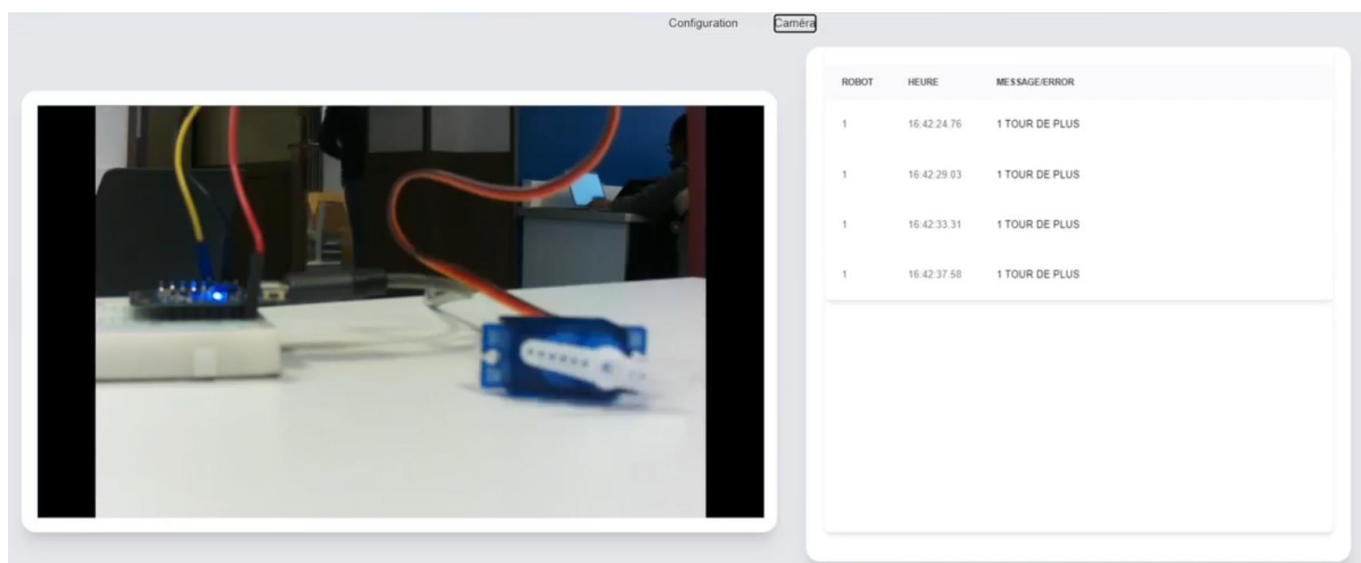


Figure 5. Retour vidéo et des logs du robots

3. Installation de l'environnement de développement

3.1 Installation de l'environnement Arduino

Arduino est une plateforme open-source de prototypage électronique basée sur du matériel et un logiciel flexible et facile à utiliser. Elle est utilisée pour créer des objets interactifs ou des environnements physiques pouvant détecter et contrôler des objets du monde réel.

Cette plateforme comprend à la fois du matériel comme une carte électronique avec un microcontrôleur (ici une carte ESP3266) et un environnement de développement (IDE) permettant aux utilisateurs de programmer la carte pour réaliser diverses tâches.

Afin de pouvoir programmer les robots, le téléchargement puis l'installation de l'environnement de développement d'Arduino est nécessaire. Pour ce faire, rendez-vous sur le site officiel Arduino : <https://www.arduino.cc/en/software> puis télécharger la dernière version d'Arduino selon votre configuration.

Downloads

Figure 6. Environnement Arduino à installer

3.1.1 Installation des librairies

L'utilisation de librairies lors de l'installation d'Arduino est importante pour étendre les fonctionnalités de base de la plateforme. Ces librairies fournissent une variété de fonctions prêtes à l'emploi, allant de la communication série à la gestion des capteurs et des actionneurs, ce qui permet aux développeurs de gagner du temps en évitant de réécrire des routines pour des tâches courantes. Elles facilitent la compatibilité entre différents projets et modèles de cartes Arduino, tout en permettant de standardiser les méthodes et protocoles de communication.

Afin de télécharger puis installer des librairies plusieurs procédures sont possibles.

a) Installation des librairies manuellement

1. **Téléchargez la bibliothèque:** Commencez par télécharger la bibliothèque depuis sa source. Il peut s'agir d'un fichier compressé (zip) ou d'un dossier contenant les fichiers de la bibliothèque.
2. **Ouvrez l'IDE Arduino:** Lancez l'IDE Arduino sur votre ordinateur.
3. **Importez la bibliothèque:** Allez dans le menu "Croquis" (Sketch), puis sélectionnez "Inclure une bibliothèque" (Include Library) > "Ajouter une bibliothèque .ZIP" (Add .ZIP Library) si vous avez téléchargé la bibliothèque au format zip, ou "Gérer les bibliothèques" (Manage Libraries) si vous avez le dossier contenant la bibliothèque.
4. **Sélectionnez le fichier ou le dossier:** Dans la fenêtre qui s'ouvre, naviguez jusqu'à l'emplacement où se trouve le fichier zip ou le dossier de la bibliothèque, puis sélectionnez-le.
5. **Validez l'installation:** Cliquez sur "Ouvrir" (Open) pour importer la bibliothèque dans l'IDE Arduino.
6. **Vérifiez l'installation:** Après l'importation, la bibliothèque apparaîtra dans le menu "Croquis" > "Inclure une bibliothèque" (Sketch > Include Library) et vous pourrez l'utiliser dans vos projets Arduino.

b) Installation des librairies via l'environnement Arduino

L'installation des librairies manuellement peut s'avérer chronophage. Il existe une méthode plus simple et plus rapide afin d'installer ces librairies, c'est à dire les installer depuis l'environnement Arduino.

1. **Ouvrez l'IDE Arduino :** Démarrez l'IDE Arduino sur votre ordinateur.
2. **Accédez à la Gestion des bibliothèques :** Dans le menu, cliquez sur "Croquis" (Sketch), puis sélectionnez "Inclure une bibliothèque" (Include Library) > "Gérer les bibliothèques" (Manage Libraries).
3. **Recherchez la bibliothèque :** Dans la fenêtre de Gestion des bibliothèques, vous pouvez rechercher la bibliothèque que vous souhaitez installer en utilisant la barre de recherche en haut de la fenêtre.
4. **Sélectionnez la bibliothèque à installer :** Une fois que vous avez trouvé la bibliothèque souhaitée, cliquez sur son nom pour afficher plus d'informations à son sujet.
5. **Installez la bibliothèque :** Pour installer la bibliothèque, cliquez sur le bouton "Installer" (Install) à côté de son nom. L'IDE Arduino téléchargera et installera automatiquement la bibliothèque pour vous.
6. **Vérifiez l'installation :** Une fois l'installation terminée, la bibliothèque apparaîtra dans le menu "Croquis" > "Inclure une bibliothèque" (Sketch > Include Library) et vous pourrez l'utiliser dans vos projets Arduino.

c) Liste des librairies nécessaire

Afin de pouvoir correctement utiliser le projet remotePi, un ensemble de librairies sont nécessaire.

- ESP32httpUpdate
- WebServer_ESP32_ENC
- ESP8266httpUpdate

- ArduinoJson
- remotePi

Remarques : Toutes les bibliothèques peuvent être téléchargées puis installées depuis l'environnement de développement sauf la bibliothèque remotePi (fournit avec le projet) qui doit être installée manuellement. De plus, les bibliothèques doivent être installées sur la machine de l'utilisateur ainsi que sur la carte Raspberry. Enfin, la configuration du réseau internet est nécessaire dans la bibliothèque remotePi afin que le robot puisse se connecter à un réseau internet. Pour ce faire, rendez-vous dans le fichier remotePi.h (dans le dossier source de la bibliothèque remotePi) puis insérez votre réseau internet ainsi que l'adresse du serveur http de la Raspberry au niveau de la ligne suivante :

```
remotePi(const char* ssid = "", const char* password = "", const char* serverAddress = "");
```

3.2 Installation de la carte Raspberry

3.2.1 Matériel nécessaire

- Une carte Raspberry Pi (modèle 3B+ ou 4B recommandé)
- Une carte MicroSD d'au moins 8 Go
- Un adaptateur USB pour carte MicroSD
- Un câble HDMI
- Un clavier et une souris USB
- Une alimentation micro USB (5V, 2.5A)
- Un écran ou un téléviseur avec port HDMI

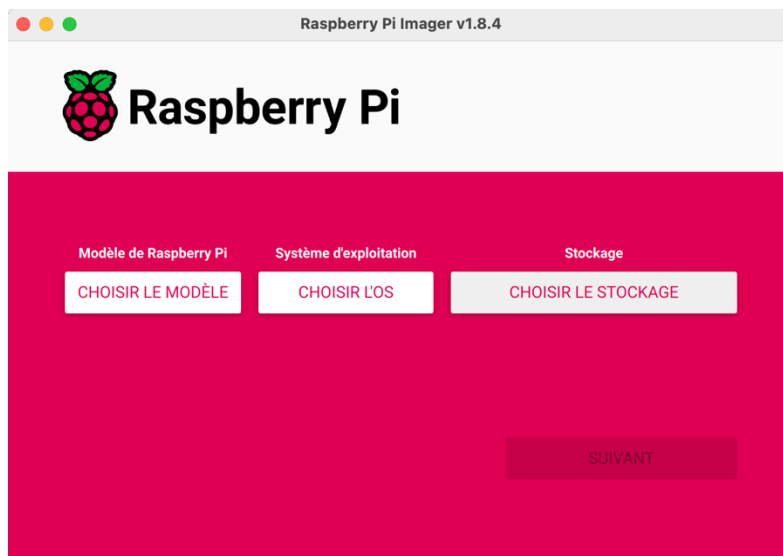
3.2.2 Installation de Raspberry Pi Imager

La première étape consiste à télécharger la dernière version de Raspberry Pi Imager depuis le site officiel. Cet outil est nécessaire afin de copier le système d'exploitation sur la carte MicroSD.

Pour ce faire, il faut se rendre sur le site <https://www.raspberrypi.org/software/>, puis télécharger la version compatible avec son ordinateur.

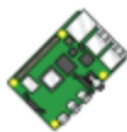
3.2.3 Formatage de la carte Micro SD

Une fois l'outil Raspberry Pi Imager téléchargé, insérez la carte MicroSD dans l'adaptateur USB puis lancez Raspberry Pi Imager.



a) Étape 1 : Modèle de Raspberry Pi

Choisissez le modèle qui correspond à la Raspberry Pi que vous souhaitez configurer. Dans le cadre du projet remotePi, il faut sélectionner la Raspberry Pi 4.



Raspberry Pi 4

Models B, 400, and Compute Modules 4, 4S

Figure 7. Modèle de Raspberry Pi à choisir

b) Étape 2 : Système d'exploitation

Choisissez le système d'exploitation adapté au modèle de la Raspberry Pi, puis lancez le téléchargement de celui-ci. Il est très important de sélectionner la bonne version pour assurer la compatibilité et les performances optimales.



Raspberry Pi OS (64-bit)

A port of Debian Bookworm with the Raspberry Pi Desktop (Recommended)

Publié le : 2023-12-05

En ligne - 1.1 GO à télécharger

Figure 8. Système d'exploitation Raspberry Pi à choisir

c) Étape 3 : Stockage

Sélectionnez le stockage sur lequel sera installer le système d'exploitation. Il s'agit de la carte MicroSD.

d) Étape 4 : Options avancées

Si vous souhaitez connecter la carte Raspberry au réseau WiFi, accédez à l'icône en bas à droite de la fenêtre. Sélectionnez le paramètre « Configurer WiFi », puis ajoutez vos paramètres WiFi.



Figure 9. Insertion des paramètres WiFi

e) Étape 5 : Écriture du système d'exploitation sur la carte MicroSD

Une fois le téléchargement terminé, Raspberry Pi Imager écrira le système d'exploitation sur la carte MicroSD. Cette opération peut prendre quelques minutes, selon la vitesse de lecture et d'écriture de la carte et de l'ordinateur.

3.3 Configuration de la carte Raspberry

3.3.1 Connexion à la Raspberry Pi via SSH :

Branchez la Raspberry Pi à l'alimentation et laissez-la démarrer. Après quelques instants, celle-ci devrait être prête à être configurée via SSH. Une fois la Raspberry Pi démarrée, ouvrez un terminal sur votre ordinateur puis insérez la commande suivante pour vous connecter à la Raspberry Pi via SSH :

```
ssh pi@raspberrypi.local
```

Si cela ne fonctionne pas, vous pouvez utiliser l'adresse IP de la Raspberry Pi (qui peut être obtenue à partir de votre routeur ou en utilisant un outil de balayage du réseau local) à la place de "raspberrypi.local".

3.3.2 Configuration initiale via SSH :

Une fois que vous êtes connecté à la Raspberry Pi via SSH, il est désormais possible d'effectuer les premières configurations. Afin de mettre à jour le système d'exploitation, insérez la commande :

```
sudo apt update && sudo apt upgrade
```

3.3.3 Installation de Arduino cli via SSH

Arduino CLI offre plusieurs fonctionnalités utiles pour les développeurs Arduino. Il permet la compilation de codes Arduino en lignes de commande, le téléversement de programmes directement sur des cartes Arduino, l'installation de bibliothèques Arduino, la gestion de multiples versions de cartes Arduino et de bibliothèques, l'automatisation des tâches de développement Arduino via des scripts et des outils de CI/CD, l'intégration avec des éditeurs de texte et des environnements de développement tiers, et la facilitation de l'intégration d'Arduino dans des projets plus vastes en permettant l'automatisation des processus de compilation et de téléversement.

a) Téléchargement d'Arduino CLI :

Sur la Raspberry Pi, téléchargez la dernière version de Arduino CLI depuis le site officiel en utilisant la commande suivante :

- wget https://downloads.arduino.cc/arduino-cli/arduino-cli_latest_Linux_64bit.tar.gz

Cette commande télécharge l'archive compressée contenant Arduino CLI.

b) Extraction de l'archive :

Une fois le téléchargement terminé, extrayez le contenu de l'archive via la commande suivante :

```
tar -xzf arduino-cli_latest_Linux_64bit.tar.gz
```

c) Déplacement du fichier binaire :

Une fois que vous avez extrait l'archive contenant Arduino CLI, vous devez vous assurer que vous êtes dans le bon répertoire pour accéder au fichier binaire arduino-cli. Cette étape est nécessaire car vous devez être en mesure d'exécuter le programme à partir de n'importe quel emplacement sur votre système. Le fichier binaire arduino-cli est l'exécutable principal qui vous permettra d'interagir avec Arduino CLI à partir de la ligne de commande. En déplaçant le fichier binaire vers un emplacement approprié, vous facilitez son utilisation ultérieure sans avoir à spécifier le chemin complet à chaque fois que vous souhaitez l'exécuter.

Accédez au répertoire où se trouve le fichier binaire Arduino CLI extrait avec la commande :

```
cd ~/arduino-cli_folder/
```

Une fois dans le bon répertoire, vous pouvez déplacer le fichier binaire vers un emplacement approprié. Pour cela, utilisez la commande **mv** (move). Déplacez le fichier binaire vers le répertoire **/usr/local/bin/**, qui est souvent inclus dans la variable **PATH** pour que les exécutables soient accessibles globalement. Utilisez la commande suivante :

```
sudo mv arduino-cli /usr/local/bin/
```

d) Configuration du chemin d'accès :

Dès que le fichier binaire déplacé, vous devez modifier le fichier **.bashrc** pour ajouter le chemin d'accès au dossier contenant Arduino CLI à la variable **PATH**. Ouvrez le fichier **.bashrc** avec un éditeur de texte. Vous pouvez utiliser la commande **nano** pour cela :

```
nano ~/.bashrc
```

Ajoutez ensuite la ligne suivante à la fin du fichier **.bashrc**, en remplaçant **/chemin_vers_le_dossier_arduino-cli/** par le chemin absolu du dossier où se trouve le fichier binaire Arduino CLI :

```
export PATH=$PATH:/chemin_vers_le_dossier_arduino-cli/
```

Si vous avez déplacé le fichier binaire vers **/usr/local/bin/**, vous n'avez pas besoin d'ajouter manuellement ce chemin car il est déjà inclus dans la variable **PATH** par défaut sur de nombreux systèmes. Cependant, si vous avez choisi un autre emplacement, assurez-vous d'ajouter le chemin correct.

Enregistrez les modifications apportées au fichier **.bashrc**, puis rechargez le fichier de configuration en utilisant la commande suivante :

```
source ~/.bashrc
```

Une fois ces étapes terminées, vous devriez être en mesure d'utiliser Arduino CLI de manière globale sur votre système Raspberry Pi en tapant simplement **arduino-cli** dans n'importe quel terminal.

e) Vérification de l'installation

Vérifiez que Arduino CLI est correctement installé en exécutant la commande suivante :

```
arduino-cli version
```

3.3.4 Installation des outils ESP8266 pour Arduino CLI

a) Création du répertoire Arduino15 et du fichier de configuration arduino-cli.yaml

La première étape consiste à créer le répertoire **Arduino15** dans votre répertoire utilisateur (s'il n'existe pas déjà) et à créer le fichier de configuration **arduino-cli.yaml** à l'intérieur. Cette étape permet de configurer Arduino CLI pour votre environnement. Insérez les commandes suivantes dans le terminal de la Raspberry Pi :

```
mkdir -p ~/.arduino15  
~/.arduino15/arduino-cli.yaml
```

b) Configuration du fichier arduino-cli.yaml

Après avoir créé le fichier **arduino-cli.yaml**, vous devez y ajouter la configuration nécessaire pour indiquer à Arduino CLI où trouver les informations sur les cartes ESP8266. Pour ce faire, Ouvrez le fichier **arduino-cli.yaml** avec un éditeur de texte pour y ajouter la configuration nécessaire. Utilisez la commande **nano** pour ouvrir le fichier :

```
nano ~/.arduino15/arduino-cli.yaml
```

Ajoutez les lignes suivantes dans le fichier pour spécifier l'URL du gestionnaire de cartes ESP8266 :

```
1. board_manager:  
2.   additional_urls:  
3.     - http://arduino.esp8266.com/stable/package_esp8266com_index.json  
4.
```

Cela indique à Arduino CLI d'ajouter l'URL spécifiée à la liste des sources pour rechercher les informations sur les cartes ESP8266. Enfin, enregistrez les modifications et quittez l'éditeur de texte (dans **nano**, utilisez **Ctrl + X** pour quitter, suivi de **Y** pour confirmer et **Enter** pour sauvegarder).

c) Mise à jour de l'index des cœurs Arduino

Avant d'installer le cœur ESP8266, assurez-vous que l'index des cœurs Arduino est à jour en exécutant la commande suivante :

```
arduino-cli core update-index
```

d) Installation du cœur ESP8266

Une fois l'index des cœurs Arduino mis à jour, il est désormais possible d'installer le cœur ESP8266 sur le système de la Raspberry Pi. Pour ce faire, insérez la commande suivante :

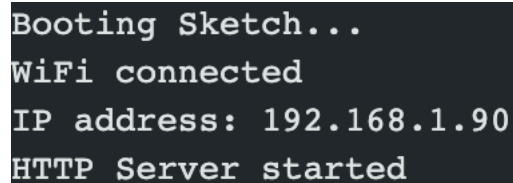
```
arduino-cli core install esp8266:esp8266
```

Cette commande télécharge puis installe le package ESP8266 pour Arduino CLI. Elle spécifie le cœur ESP8266 avec l'identifiant **esp8266:esp8266**.

3.4 Configuration du Robot

Une fois que la carte Raspberry Pi est correctement initialisée, vous pouvez procéder à la configuration des robots utilisés dans le cadre du projet RemotePi. Ces robots sont basés sur une carte ESP3266, offrant plusieurs avantages significatifs. Tout d'abord, cette carte est programmable directement depuis l'interface Arduino, simplifiant ainsi le processus de programmation. De plus, elle est équipée d'un module WiFi intégré, facilitant la mise en place d'un serveur HTTP pour la communication entre le robot et la Raspberry Pi.

La première étape consiste à initialiser le robot pour le rendre capable de recevoir du code. Pour ce faire, vous devez connecter la carte ESP3266 à votre ordinateur via un câble USB. Ensuite, vous devez récupérer le code "initESP.ino" (fournit avec le rendu du projet) et le compiler. Une fois la compilation terminée, vous devez le téléverser sur la carte ESP3266. Dès que le téléversement accompli, vous devez accéder à l'onglet "Serial Monitor" dans l'interface Arduino. Après quelques secondes, des messages spécifiques devraient apparaître dans la console dont l'adresse IP du robot qui vous sera utile. À ce stade, le robot est prêt à être utilisé, c'est-à-dire qu'il est prêt à recevoir du code pour fonctionner.



```
Booting Sketch...  
WiFi connected  
IP address: 192.168.1.90  
HTTP Server started
```

Figure 10. Retour de connexion du robot au réseau

3.5 Test de téléversement

A cette étape, la Raspberry ainsi que les robots sont configurés et donc prêts à l'emploi. Afin de s'assurer que tout fonctionne comme souhaité, il est possible de réaliser le premier téléversement à distance. A la fin de la mise en place de projet remotePi, vous devriez être capable d'envoyer un code à un robot souhaité, qu'il soit compilé ou non.

3.5.1 Prérequis

Les essais de fonctionnement dans le cadre de remotePi sont effectués avec deux robots qui ont un fonctionnement simple. Un robot munit d'un capteur à ultrason, puis un robot munit d'un servomoteur. Voici le schéma de câblage effectué.

a) Robots avec capteur à ultrason

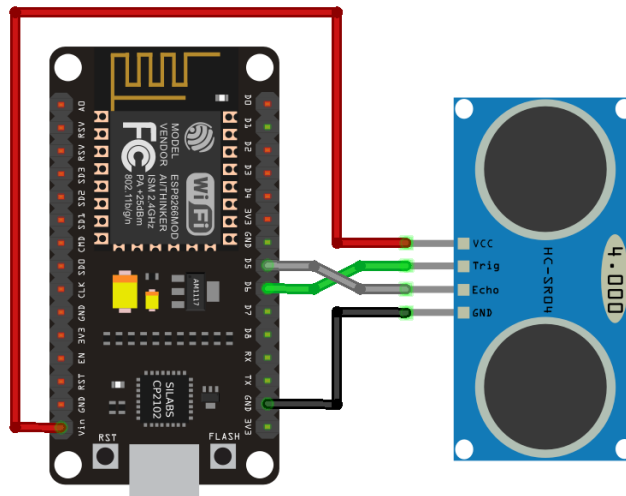


Figure 11. Câblage du robot avec le module ultrason

b) Robots avec un servomoteur

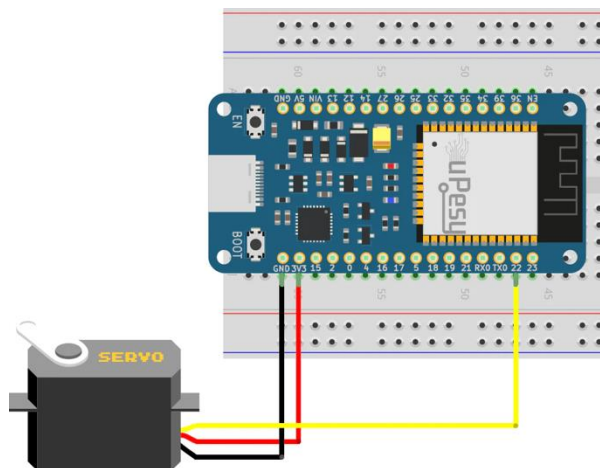


Figure 12. Câblage du robot avec le servomoteur

3.5.2 Envoi d'un code compilé

La première étape de cette tâche consiste à compiler le programme appelé « ultraSonicSensorCode.ino » depuis l'environnement de développement Arduino. Pour ce faire, rendez-vous dans **Croquis**, puis **Exporter les binaires compilés**. Cette procédure vous permettra de compiler le code puis de vous exporter le fichier binaire que vous allez envoyer au robot.

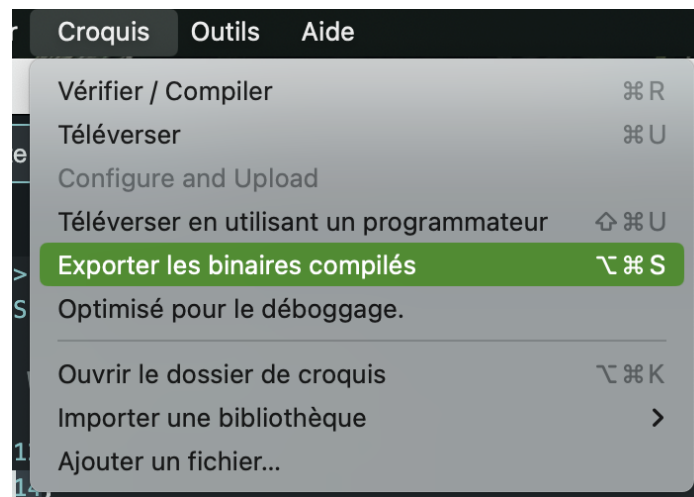


Figure 13. Extraction du fichier binaire

Une fois le fichier binaire exporté, il est possible d'envoyer celui-ci vers la Raspberry. Ouvrez le terminal puis insérez la commande suivante :

```
Scp
/Chemin/Du/Fichier/Binaire/ultraSonicSensorCode.ino.bin pi@AdresseIP_Raspberry:/home/pi/Chemin/DeDestination
```

Exemple de commande testé au cours des essais :

```
scp /Users/xago/Desktop/ECE/ING5/PFE/binTest/ultraSonicSensorCode.ino
pi@192.168.1.91:/home/pi/Documents/PFE_Project
```

Une fois le fichier binaire envoyé vers la Raspberry, il est possible de l'envoyer vers le robot souhaité. Accédez au dossier où se trouve le fichier binaire que vous avez envoyé au préalable. Enfin, envoyez la commande suivante.

```
curl -F "image=@filename.bin" http://adresse_IP_robot/update
```

Une fois le téléversement effectué, l'ESP redémarrera puis s'allumera sur le bon code.

```
remote: Filename: esp8266_Servo.ino.bin X IP à curl : http://172.20.10.3/update
remote:  % Total      % Received % Xferd  Average Speed   Time    Time     Time  Current
remote:                                 Dload  Upload  Total  Spent    Left  Speed
remote: 100 345k 100   74 100 344k    4 23753  0:00:18  0:00:14  0:00:04 24281
remote: <META http-equiv="refresh" content="15;URL=/">Update Success! Rebooting...
```

Figure 14. Retour du robot succès téléversement

3.5.3 Envoi d'un code non compilé

Le projet remotePi offre également la possibilité de pouvoir envoyer un code non compilé. Pour ce faire, rendez-vous dans le dossier où se trouve votre code Arduino (fichier avec l'extension .ino). Puis envoyez ce code vers la carte Raspberry avec la commande **scp** suivante :

```
Scp /Chemin/Du/Fichier/Binaire/ultraSonicSensorCode.ino pi@AdresseIP_Raspberry:/home/pi/Chemin/De/Destination
```

Une fois le fichier envoyé vers la Raspberry, il faut procéder à la compilation du code directement sur la carte Raspberry. Rendez-vous dans le dossier où se trouve le fichier envoyé puis exécutez la commande suivante :

e) Compilation du code pour la carte ESP8266 (NodeMCU)

Enfin, pour compiler un code source .ino pour la carte ESP8266 (NodeMCU), utilisez la commande suivante :

```
arduino-cli compile -b esp8266:esp8266:nodemcu chemin_vers_le_fichier.ino
```

Assurez-vous de remplacer **chemin_vers_le_fichier.ino** par le chemin absolu vers votre fichier source Arduino. Cette commande utilise Arduino CLI pour compiler le code source pour la carte ESP8266 spécifiée. De plus, elle spécifie le matériel et les paramètres de compilation à utiliser.

Enfin, lorsque le fichier compilé, rendez-vous dans le fichier où se trouve le fichier binaire généré par la compilation, puis, exécutez la commande suivante :

```
curl -F "image=@filename.bin" http://adresse_IP_robot/update
```

Comme pour l'étape précédente, une fois le téléversement effectué, le message « Update Success ! » s'affiche dans la console.

A ce stade, vous êtes capable de d'envoyer du code compilé ou non compilé à un robot souhaité, depuis le terminal de votre ordinateur.

3.6 Mise en place du GIT

Depuis la dernière version de RemotePi, la mise en place du git n'a plus besoin d'être réalisée manuellement. Lors de l'upload d'un code, le serveur backend exécutera les commandes permettant le déploiement d'un projet git vide si nécessaire. Si un projet git est déjà existant, alors le serveur passera directement à la phase de commit, push et gestion de branches, sans aucune interaction requise des développeurs.

3.7 Mise en place du backend

Un backend peut être considéré comme le cerveau caché derrière une application. Il gère toutes les informations importantes et les actions nécessaires pour que l'application fonctionne correctement. Cela inclut la gestion des utilisateurs, le stockage des données, et la prise en charge de toutes les fonctionnalités que vous voyez à l'écran. En bref, le backend est essentiel pour faire fonctionner une application de manière fluide et fiable.

3.7.1 Téléchargement local

Pour commencer, vous devez cloner le projet RemotePi depuis GitHub. Ouvrez votre terminal et exécutez les commandes suivantes :

```
1. git clone https://github.com/ElmoAlreadyTaken/PFE_RemotePi
2. cd PFE_RemotePi
```

3.7.2 Backend

Accédez au dossier du serveur en utilisant la commande suivante :

```
cd server
```

Ensuite, installez les dépendances nécessaires via APT en exécutant :

```
sudo apt update && sudo apt install -y libcamera-apps libcamera0 && \ curl -s https://ngrok-agent.s3.amazonaws.com/ngrok.asc | sudo tee /etc/apt/trusted.gpg.d/ngrok.asc >/dev/null && echo "deb https://ngrok-agent.s3.amazonaws.com buster main" | sudo tee /etc/apt/sources.list.d/ngrok.list && sudo apt install ngrok
```

Après cela, installez les modules Python requis en utilisant pip :

```
pip install -r requirements.txt
```

Si vous le souhaitez, vous pouvez créer un compte Ngrok et modifier le token d'authentification dans ngrok/ngrok.conf.

Enfin, démarrez le serveur Flask et les tunnels Ngrok pour le serveur et la caméra en exécutant le script :

```
./startServers.sh
```

3.7.3 Interface web :

Accédez au dossier de l'interface en exécutant la commande :

```
cd ../interface
```

Installez les dépendances npm avec :

```
npm install
```

Ensuite, déployez l'interface web en exécutant :

```
npm run dev
```

Vous pourrez accéder à l'interface web via <http://localhost:3000>.

3.7.4 Configuration de l'interface web :

Pour ajouter manuellement les adresses IP publiques et les ports attribués à la backend et au serveur Flask par Ngrok, rendez-vous dans le panneau Admin de l'interface Web. Entrez les IP et ports générés par Ngrok lors de l'exécution du script startServers.sh.

3.7.5 Tests (optionnel) :

RemotePi fournit un module de tests pour s'assurer du bon déploiement et du fonctionnement des différents services. Pour l'exécuter, accédez au dossier server et exécutez la commande :

```
pytest -v
```

3.7.6 Documentation :

La documentation des routes API est accessible sur la page /doc, au format SwaggerUI. Vous pouvez tester les différents paramètres des requêtes, observer les réponses de la backend, etc. Cette page facilite le développement et le débogage pendant les phases de tests.

4. Structure du code source

4.1 Structure du code de l'interface

```
interface/
├── .env.local
├── package.json
├── README.md
├── app/
│   ├── config.js
│   ├── favicon.ico
│   ├── globals.css
│   ├── assets/
│   │   ├── css/
│   │   ├── fonts/
│   │   ├── components/
│   │   │   ├── Accueil.js
│   │   │   ├── AllRobots.js
│   │   │   ├── FileUpload.js
│   │   │   ├── Footer.js
│   │   │   ├── FreeRobots.js
│   │   │   ├── Header.js
│   │   │   ├── HomePage.js
│   │   │   └── layout.js
│   │   └── lib/
│   │       ├── supabase.js
│   │       └── supabaseUtils.js
│   └── pages/
│       ├── admin.js
│       ├── camera.js
│       ├── configuration.js
│       ├── index.js
│       ├── informations.js
│       ├── inscription.js
│       ├── login.js
│       ├── profil.js
│       └── update-password.js
└── _app.js
```

4.2 Modèle de conception :

4.2.1 Rendering Côté Serveur (SSR):

Next.js utilise le SSR pour rendre les composants React côté serveur, ce qui améliore les performances de chargement initial des pages et est bénéfique pour le SEO (Search Engine Optimization).

4.2.2 Génération de Site Statique (SSG) :

Next.js permet également la Génération de Site Statique, où les pages sont rendues lors du build time, ce qui réduit la charge sur le serveur au runtime.

4.2.3 File System Based Routing :

Next.js propose un système de routage basé sur le système de fichiers. Les fichiers placés dans le dossier pages correspondent à des routes dans l'application, ce qui rend la création de nouvelles pages aussi simple que d'ajouter un nouveau fichier.

4.2.4 API Routes :

Next.js permet de créer des API directement dans le projet en utilisant les fichiers dans pages/api. Cela simplifie la création d'endpoints API sans nécessiter de serveur séparé.

Intégration de frameworks ou de bibliothèques tierces :

next.js -> framework react

API Supabase -> Gestion de base de données

AceEditor -> IDE intégrer dans l'interface web

files-ui/react -> Bibliothèque permettant un file dropzone pour importer / drag and drop un fichier

5. Bonnes pratiques de développement

Lors du développement avec RemotePi, il est important de suivre certaines bonnes pratiques pour garantir la fiabilité, la sécurité et la maintenabilité de votre code. Voici quelques recommandations essentielles :

1. **Utilisation du template de base :** L'utilisation du template fourni avec la librairie RemotePi est impérative. Ce template établit une structure de base cohérente pour le code Arduino, garantissant une configuration uniforme et stable du robot. En utilisant ce template, vous évitez les erreurs de configuration et assurez une compatibilité maximale avec la plateforme RemotePi.
2. **Ajout de logs de retours :** Pour assurer le suivi et le débogage efficace de vos programmes, il est recommandé d'ajouter des logs de retour à l'aide des méthodes fournies par la librairie RemotePi, telles que `config.sendLog()` et `config.sendErrorLog()`. Ces logs permettent de capturer des informations cruciales sur le fonctionnement du robot, les erreurs éventuelles rencontrées et les événements importants. Par exemple :

```
// Envoi d'un message de log
config.sendLog("Le robot a démarré avec succès.");
```

```
// Gestion d'une erreur
config.sendErrorLog("Erreur lors de l'initialisation du capteur de distance.");
```

3. **Utilisation des fonctionnalités de la librairie :** Explorez et utilisez toutes les fonctionnalités fournies par la librairie RemotePi pour optimiser le contrôle et la gestion des robots à distance. Que ce soit pour la gestion de la caméra, la communication avec le serveur ou la manipulation des données, l'utilisation des fonctionnalités intégrées facilite le développement et améliore l'expérience utilisateur.
4. **Tests et vérifications régulières :** Effectuez des tests réguliers de votre code pour vérifier son bon fonctionnement et détecter les éventuels problèmes. Utilisez les outils de test fournis par RemotePi, comme le module de tests, pour garantir la qualité et la robustesse de votre application.

En suivant ces bonnes pratiques de développement, vous pouvez optimiser l'efficacité de votre travail avec RemotePi, garantir la stabilité de vos programmes et offrir une expérience utilisateur optimale aux utilisateurs de votre application.

6. Références utiles

Mediamtx: <https://github.com/bluenviron/mediamtx>

Arduino-cli : <https://github.com/arduino/arduino-cli>

7. Contacts et support

Ce projet a été réalisé par six étudiants :

1. Majeure Systèmes d'Information et Cybersécurité

- **Misha Setta**: Implémentation de la plateforme web
- **Nassim Si Mohammed**: Implémentation de la plateforme web
- **Axel Pape**: Implémentation du serveur et de l'API pour la communication plateforme web <--> serveur
- **Quentin Sereno**: Implémentation du serveur et de l'API pour la communication plateforme web <--> serveur

2. Majeure Systèmes embarqués

- **Alexandre Santos Gomes**: Gestion des robots et transfert de code
- **Maxime Peres**: Gestion des robots et transfert de code

Si vous rencontrez des problèmes au niveau de la mise en place du projet, il vous est possible de contacter notre groupe de projet à l'adresse : remotepiT40@gmail.com

8. Conclusion

En conclusion, le projet RemotePi représente une réponse novatrice et pratique aux défis posés par le contrôle à distance des robots à l'aide d'un Raspberry Pi. En mettant l'accent sur la facilité d'utilisation, la sécurité et l'accessibilité, RemotePi vise à démocratiser l'apprentissage de la programmation et de la manipulation robotique, en particulier pour les étudiants et les passionnés de technologie.

À travers ce guide de prise en main, nous avons exploré en détail toutes les facettes de RemotePi, depuis la création d'un compte jusqu'à l'installation de l'environnement de développement et la configuration des robots. En suivant les bonnes pratiques de développement recommandées, telles que l'utilisation du template de base, l'ajout de logs de retours et l'exploration des fonctionnalités de la librairie, les utilisateurs peuvent maximiser l'efficacité et la fiabilité de leurs projets.

En somme, RemotePi ouvre de nouvelles perspectives dans le domaine de la robotique éducative et expérimentale, offrant une plateforme puissante et conviviale pour explorer les possibilités infinies de la programmation et de la manipulation robotique. Nous espérons que ce guide vous aidera à tirer pleinement parti de cette expérience unique et enrichissante.

9. Table des illustrations

Figure 1. Inscription sur l'interface remotePi	4
Figure 2. Page de téléversement remotePi	4
Figure 3. Sélection du robot souhaité	5
Figure 4. Possibilités d'envoi de code aux robots	5
Figure 5. Retour vidéo et des logs du robots	7
Figure 6. Environnement Arduino à installer	8
Figure 7. Modèle de Raspberry Pi à choisir	11
Figure 8. Système d'exploitation Raspberry Pi à choisir	11
Figure 9. Insertion des paramètres WiFi	12
Figure 10. Retour de connexion du robot au réseau	16
Figure 11. Câblage du robot avec le module ultrason	17
Figure 12. Câblage du robot avec le servomoteur	17
Figure 13. Extraction du fichier binaire.....	18
Figure 14. Retour du robot succès téléversement.....	18