Projet Reading Eye - État d'Avancement

Phase 3 : Code GitHub, Assemblage et Tests Raspberry Pi

Équipe : Bouba Ahmed & Lkhalidi Mohamed
Superviseur : Pr. Ahmed Regragui

January 9, 2026

Master 2 - Systèmes Intelligents pour l'Éducation (SIE)
ENS Meknès

Plan de la présentation

Récapitulatif Phase 1

Phase 1 : Prototypage Desktop

Objectifs Phase 1 : ✔ COMPLÉTÉS

Applications Desktop :

- ✔ Interface GUI (CustomTkinter)
- ✔ Capture caméra web
- ✔ OCR Tesseract
- ✔ TTS (pyttsx3 + gTTS)
- ✔ Support 3 langues

Technologies :

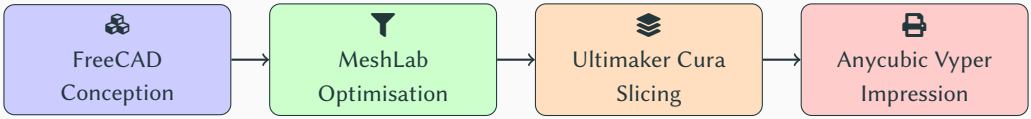
- ✔ Python 3.13+
- ✔ OpenCV
- ✔ Tesseract
- ✔ Tests validés
- ✔ Code documenté

Résultat : Code prêt à migrer sur Raspberry Pi

Design & Impression 3D

Conception et Impression - Résumé

Workflow complet réalisé :



Résultats :

- ✔ Boîtier design optimisé pour Raspberry Pi 5 + Caméra
- ✔ Impression réussie en PLA (14h 35min)
- ✔ Dimensions conformes : 180×120×95 mm
- ✔ Finitions appliquées (ponçage, nettoyage)

Développement Code

Repository GitHub - Organisation

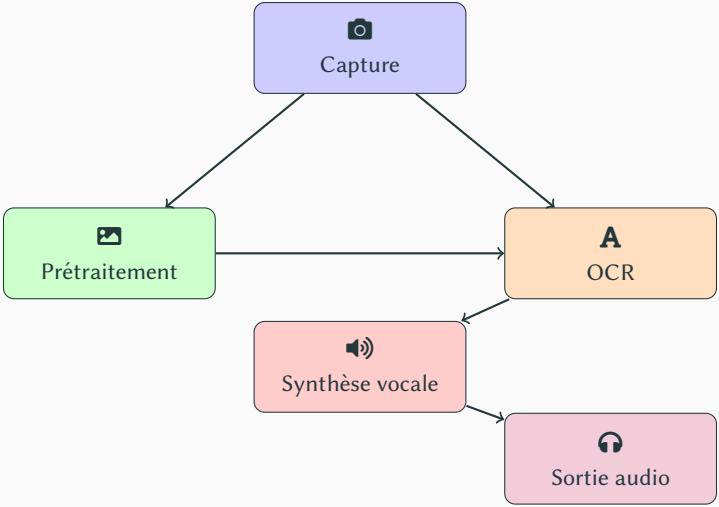
Dépôt public :

`https://github.com/BoubaAhmed/reading-eye-raspberry-pi`

Structure du projet :

- `scripts/` - Modules Python
  - `app_main.py` - Main application
  - `camera.py` - Capture Pi Camera
  - `ocr.py` - Tesseract OCR
  - `tts.py` - Synthèse vocale
- `config/` - Fichiers configuration
- `capture/` - Images capturées
- `logs/` - Logs d'exécution
- `requirements.txt` - Dépendances
- Scripts utilitaires :
  - `run.sh` - Lancement
  - `setup.sh` - Installation
  - `system_setup.sh` - Dépendances système

Architecture Logicielle



Intégration Matérielle

Assemblage Hardware - Boîtier Custom

Configuration matérielle :

Composants intégrés :

- Raspberry Pi 5 (4GB)
- Pi Camera Module 3
- Alimentation USB-C (5V/3A)
- Sortie audio 3.5mm
- Bouton d'alimentation
- Bouton de capture

Caractéristiques boîtier :

- Design ergonomique
- Accès aux ports (USB, HDMI)
- Ventilation passive
- Montage caméra stable
- Portabilité optimale

✔ Assemblage validé et fonctionnel

Vérifications Hardware

Tests d'intégration réussis :

Composant	Statut	Détails
Raspberry Pi 5	✔ OK	Boot, réseau, USB
Caméra Pi Module 3	✔ OK	Détectée, capture
Sortie Audio	✔ OK	Jack 3.5mm actif
Alimentation	✔ OK	Stable 5V/3A
Boutons GPIO	✔ OK	Interruptions
Température	✔ OK	35-45°C

Hardware prêt pour déploiement logiciel

Déploiement Raspberry Pi

Configuration SSH et Clonage

Étapes de déploiement :

1. Connexion SSH :

```
ssh raspberryens@192.168.43.197
# ou
ssh raspberryens@raspberryens
```

2. Clonage GitHub :

```
mkdir -p /Projet7Readingeye
cd /Projet7Readingeye
git clone https://github.com/BoubaAhmed/reading-eye-raspberry-pi.git .
```

3. Environnement virtuel :

```
python3 -m venv envprojet7
source envprojet7/bin/activate
```

4. Installation :

```
# Dpendances syst me
sudo bash systemsetup.sh

# Dpendances Python
pip install -r requirements.txt
```

Configuration Logicielle RPi

Dépendances système installées :

Système :

- ✔ Raspberry Pi OS 64-bit
- ✔ Tesseract OCR + langues
- ✔ Picamera2 (libcamera)
- ✔ ALSA audio tools
- ✔ Build essentials

Python (env_projet_7) :

- ✔ opencv-python-headless
- ✔ pytesseract
- ✔ pyttsx3 + gTTS
- ✔ numpy
- ✔ RPi.GPIO
- ✔ python-dotenv

Configuration caméra :

```
sudo raspi-config
# Interface Options     Camera     Enable
# Reboot
libcamera-hello --list-cameras
```

Configuration Application

Fichier de configuration : `config/reading_eye_config.json`

```
-
  "ocrlanguage": "fra+eng",
  "ttslanguage": "fr",
  "ttsrate": 150,
  "ttsvolume": 0.9,
  "cameraresolution": [1280, 720],
  "captureinterval": 5.0,
  "savecapturedimages": true,
  "enablelogging": true,
  "loglevel": "INFO"
```

Utilisation :

```
# Capture unique
bash run.sh --single --lang fra+eng --save-image

# Mode continu
bash run.sh --loop --interval 5.0 --lang fra+eng

# Mode verbose
bash run.sh --single --verbose
```

Tests et Validation

Tests Fonctionnels Prévus

Séquence de validation :

1. **Test SSH et réseau**
   - Connexion SSH stable
   - Clonage GitHub réussi
   - Installation automatique
2. **Test capture caméra**
   - Résolution 1280×720
   - Autofocus fonctionnel
   - Différents éclairages
3. **Test OCR**
   - Polices variées (Arial, Times)
   - Tailles de texte (10-24pt)
   - Langues (FR, EN)
   - Texte manuscrit (si modèle disponible)

Tests Performance et Stabilité

Métriques performance :

- Latence capture→audio
- Utilisation CPU/GPU
- Consommation mémoire
- Température Pi 5
- Autonomie batterie

Objectifs :

- Latence ¡ 5 secondes
- CPU ¡ 70% en usage
- RAM ¡ 2GB utilisé
- Température ¡ 60°C

Tests stabilité :

- Mode `--loop` 24h
- 100+ captures consécutives
- Changements de langue
- Gestion des erreurs :
  - ”Aucun texte détecté”
  - ”Caméra non disponible”
  - ”Erreur audio”

Tests Utilisateur (Simulés)

Scénarios réels :

- **Livre imprimé** : Pages texte standard
- **Document manuscrit** : Notes personnelles
- **Journal/quotidien** : Polices variées
- **Étiquettes** : Texte petit (10-12pt)
- **Panneaux** : Texte contrasté

Métriques utilisateur :

- Compréhension audio (clarté)
- Vitesse de lecture (TTS rate)
- Facilité d'utilisation (1 bouton)
- Feedback sonore (capture/erreur)
- Portabilité/confort

Roadmap et Améliorations

Prochaines Étapes (Phase 3)

Immédiat (semaine prochaine) :

- ✔ Clonage code sur Raspberry Pi
- ✔ Configuration SSH et environnement
- ✔ Tests fonctionnels complets
- ✔ Optimisation performance
- ✔ Documentation résultats

Court terme :

- Tests utilisateurs réels
- Amélioration précision OCR
- Réduction latence
- Optimisation consommation
- Interface vocale simple

Améliorations Futures

Fonctionnalités avancées :

OCR amélioré :

- Support manuscrit (CRNN)
- Plus de langues (AR, ES, DE)
- Reconnaissance formules mathématiques
- Détection mise en page

Interface utilisateur :

- Reconnaissance vocale
- Commandes ”Capture”, ”Relis”
- Traduction instantanée
- Mode économie d’énergie

Hardware :

- Batterie haute capacité
- Écran tactile feedback
- Connectivité Bluetooth audio
- Capteur de distance (focus auto)

Conclusion et Démonstration

État d'Avancement Global

Progression du projet :

Phase	Élément	Statut
Phase 1	Prototypage Desktop	✔ COMPLET
Phase 2	Design 3D + Impression	✔ COMPLET
Phase 2	Développement Code	✔ COMPLET
Phase 2	Intégration Hardware	✔ COMPLET
Phase 3	Déploiement RPi	EN COURS
Phase 3	Tests & Validation	À VENIR
Phase 4	Optimisation	À VENIR

Prêt pour :

- Démonstration fonctionnelle
- Tests complets Raspberry Pi
- Validation avec utilisateurs

Démonstration Prête

Matériel disponible pour démo :

B oîtier assemblé :

- Raspberry Pi 5 intégré
- Caméra Pi Module 3
- Bouton de capture
- Sortie audio 3.5mm
- Alimentation USB-C

Logiciel prêt :

- Code sur GitHub
- Scripts d'installation
- Configuration optimisée
- Documentation complète

✔ PRÊT POUR LA DÉMONSTRATION LIVE

(SSH → Clone → Installation → Test)

Merci de votre attention

État d'Avancement : Phase 2 Complète

Ressources :

- GitHub : `https://github.com/BoubaAhmed/reading-eye-raspberry-pi`
- Documentation : README.md détaillé
- Contact : ah.bouba@edu.umi.ac.ma & simolkhalidi22@gmail.com