Table des matières

1 Introduction

L'évaluation par QCM est couramment utilisée dans les établissements d'enseignement pour sa simplicité et son objectivité. Toutefois, lorsque le nombre d'élèves devient important, la correction manuelle devient chronophage et sujette à l'erreur. Dans un contexte pédagogique où les examens se répètent régulièrement, l'automatisation du traitement de QCM devient une nécessité.

C'est dans cette optique que s'inscrit le projet **GTO** (Grading Test Operator), qui vise à fournir une solution intégrée, fiable et légère pour générer des sujets, lire des copies scannées et afficher automatiquement les résultats. Le système a été pensé pour être utilisé de manière autonome dans un environnement pédagogique, avec une prise en main intuitive et un coût de déploiement minimal.

Le projet repose sur une architecture modulaire combinant génération des sujets, traitement d'image pour lecture des réponses, format de stockage structuré, et affichage embarqué sur écran OLED. La navigation entre les résultats des candidats est assurée par une interface physique simple, adaptée aux contraintes d'un microcontrôleur.

Le présent rapport détaille les choix techniques et méthodologiques qui ont guidé la conception du système, l'état d'avancement actuel ainsi que les perspectives d'évolution, notamment en termes d'ergonomie et de traitement automatisé à grande échelle.

SCHEMA DE MATT

2 Informations générales du projet

— **Nom du projet** : GTO (Grading Test Operator)
— **Encadrants** : Jordan DUFRESNE et Antoine BULARD
— **Équipe projet** : Matthieu DAMIEN, Emmanuel BRUCHARD et Bastien BOURDIN
— **Formation concernée** : ESEO E3e
— **Durée du projet** : janvier à mai 2025
— **Date de rendu prévue** : 2025

3 Cahier des charges fonctionnel et technique

3.1 Environnement d'utilisation

— Le système doit fonctionner dans une température ambiante comprise entre 10°C et 30°C.
— Le système doit rester opérationnel sous une hygrométrie relative comprise entre 40% et 95%.

3.2 Alimentation

— Le système doit être alimenté par une source 5V, 1A de type AC/DC.
— La consommation électrique du système ne doit pas excéder 12W.

3.3 Entrées

— Le système doit permettre la lecture de fichiers `.gto` depuis une carte microSD via un lecteur compatible.
— Le système doit intégrer un emplacement pour carte microSD assurant le stockage local des copies et résultats.
— Le système doit comporter trois entrées numériques supportant des tensions entre 0 et 3.3V.

3.4 Sorties

— Le système doit afficher les résultats sur un écran OLED intégré.
— Le système doit comporter une LED d'indication signalant le traitement en cours.

3.5 Interfaces Homme-Machine

— Le système doit permettre la navigation dans les résultats à l'aide d'un potentiomètre ou d'un codeur incrémental.
— Le système doit afficher sur l'écran OLED le nom du candidat, son numéro d'identification et sa note.
— Le système doit fournir un retour visuel clair (via LED) indiquant l'état d'activité (repos, lecture, traitement).

4 Conception et architecture du projet

4.1 Schéma Bloc

Présentation visuelle des blocs fonctionnels : STM32, SD, OLED, LED, codeur incrémental, alimentation.

4.2 Choix techniques

Justification des choix techniques (STM32, OLED, carte SD, etc.)

5 Génération automatique des QCM (Python/Turtle)

5.1 Structure visuelle des feuilles

— Format A4, 2 colonnes, 20 questions
— Bandeau d'identification ID élève
— Cercles réponses (A/B)
— Simulation remplissage réaliste

5.2 Génération des fichiers associés

— Export PDF via `epstopdf`
— Génération coordonnées cases (format .gto/json)

6 Traitement d'image

6.1 Prétraitement (MATLAB/OpenCV)

— Chargement images scannées
— Conversion niveaux de gris
— Seuillage adaptatif, opérations morphologiques

6.2 Extraction des réponses

— Détection réponses cochées
— Validation positions (bounding boxes)

6.3 Comparaison et notation automatique

— Chargement corrigé (.gto/json)
— Comparaison réponses élève/corrigé
— Calcul et affichage notes (OLED)

7 Interfaces et interactions

7.1 Écran OLED

Sélection fichier, affichage résultats

7.2 Codeur incrémental

Navigation intuitive, validation

7.3 LED

Indicateur traitement en cours

8 Résultats préliminaires et validation

Performance actuelle (temps traitement, précision)

9 Difficultés rencontrées et solutions apportées

Alignement graphique, précision traitement image, conversion coordonnées

10 Perspectives d'amélioration

— QR code identification
— Traitement par lots
— Interface utilisateur avancée (Tkinter/webapp)
— Reconnaissance OCR pour ID

11 Conclusion

Résumé objectifs atteints, bilan état actuel, prochaines étapes

A Annexes

— Extraits code Python (Turtle)
— Extraits code MATLAB/OpenCV
— Schéma bloc détaillé
— Tableau spécifications techniques
— Captures d'écran résultats tests
— Arborescence projet