

Documentation Fonctionnelle & Technique – Projet Simon Game ESP32

Sommaire

1. Objectif du projet
2. Fonctionnalités principales
3. Architecture générale
4. Interfaces utilisateur
5. Détails techniques
6. Schéma des menus
7. Utilisation de l'EEPROM
8. Améliorations possibles

1. Objectif du projet

Développer un jeu Simon fonctionnel sur ESP32, avec :

- Boutons physiques pour les interactions
- LEDs pour afficher la séquence
- Écran OLED pour les menus
- Paramétrage dynamique : langue + mode LED (classique ou inversé)
- Sauvegarde des préférences avec EEPROM

2. Fonctionnalités principales

- Jouer au Simon Game (mode mémoire)
- Changer la langue d'affichage
- Choisir un mode LED :
 - Classique : LEDs éteintes s'allument une par une

- Inversé : LEDs allumées s'éteignent une par une
- Naviguer via un menu affiché sur un écran OLED
- Sauvegarder les préférences (langue, mode LED) dans l'EEPROM

3. Architecture générale

- Microcontrôleur : ESP32
- Affichage : Écran OLED (SSD1306) via I2C
- Contrôle : 4 boutons (haut, bas, centre, retour)
- Affichage des séquences : 4 LEDs
- Sauvegarde des réglages : EEPROM intégrée à l'ESP32

4. Interfaces utilisateur

Menu Principal :

- Jouer
- Paramètres
- Quitter

Menu Paramètres :

- Langue
- Mode LED (Classique / Inversé)
- Retour

5. Détails techniques

LEDs

- 4 LEDs connectées aux GPIOs
- Mode Classique : LEDs éteintes s'allument temporairement pour la séquence
- Mode Inversé : LEDs allumées s'éteignent temporairement pour la séquence

Boutons

- 4 boutons :
 - Haut : navigation menu vers le haut
 - Bas : navigation menu vers le bas
 - Centre : valider/entrer
 - Retour : revenir en arrière

EEPROM

Utilisation de la bibliothèque EEPROM pour sauvegarder les réglages utilisateur.

- Adresse 0 : Langue (int)
- Adresse 4 : Mode LED (booléen, 0 = classique, 1 = inversé)

Initialisation :

```
EEPROM.begin(5);
```

Lecture :

```
int langue = EEPROM.read(0);
```

```
bool ledModelInverse = EEPROM.read(4);
```

Écriture :

```
EEPROM.write(0, langue);
```

```
EEPROM.write(4, ledModelInverse);
```

```
EEPROM.commit();
```

6. Schéma des menus

Menu Principal :

- Jouer → Lance le jeu
- Paramètres → Ouvre un sous-menu
 - Langue → Changement de langue
 - Mode LED → Classique / Inversé
 - Retour → Revenir au menu principal
- Quitter → Quitte le jeu ou met en veille

7. Utilisation de l'EEPROM

Taille utilisée : 5 octets minimum

- Adresse 0 → Langue sélectionnée
- Adresse 4 → Mode LED (classique/inversé)

Utilisation :

- EEPROM.begin(taille) pour initialiser
- EEPROM.read(adresse) pour lire
- EEPROM.write(adresse, valeur) pour écrire
- EEPROM.commit() pour sauvegarder

8. Améliorations possibles

- Ajout d'un buzzer pour feedback sonore
- Sauvegarde du meilleur score

- Mode 2 joueurs
- Animations LED personnalisées
- Ajout d'un joystick pour navigation fluide
- Affichage graphique plus poussé (icônes, transitions)