

Rapport de Montage - Système de Présence Automatique

Nom et Prénom : Bouba Ahmed & Lkhalidi Mohammed

Projet IoT - Système de Présence Automatique

École Normale Supérieure de Meknès

Date : 15 décembre 2025

Introduction

Ce rapport présente les étapes de montage pour le prototype du système de présence automatique. Deux montages distincts sont réalisés : l'un avec le module RFID RC522, l'autre avec le capteur d'empreinte digitale. Ces montages seront ensuite intégrés dans un système unique permettant une identification rapide et fiable des étudiants.

1. Montage 1 : Système RFID avec ESP32

Matériel nécessaire

- ◇ ESP32
- ◇ Module RFID RC522
- ◇ Écran LCD 16x2 avec interface I2C
- ◇ Buzzer
- ◇ 2 LEDs (verte et rouge)
- ◇ 2 résistances 220
- ◇ Breadboard
- ◇ Câbles Dupont (M-M, M-F)

Connexions entre ESP32 et RFID RC522

TABLE 1 – Broches utilisées pour le système RFID

Pin RFID RC522	Pin ESP32	Description
SDA (SS)	GPIO5	Chip Select
SCK	GPIO18	Horloge SPI
MOSI	GPIO23	Master Out Slave In
MISO	GPIO19	Master In Slave Out
GND	GND	Masse
RST	GPIO22	Reset
3.3V	3.3V	Alimentation

Composant	Pin ESP32	Description
LCD SDA	GPIO21	Données I2C
LCD SCL	GPIO22	Horloge I2C
Buzzer	GPIO25	Signal sonore
LED Verte	GPIO26 (+ résistance 220)	Présence validée
LED Rouge	GPIO27 (+ résistance 220)	Erreur

Connexions des périphériques additionnels

Code de test RFID

```

1 #include <SPI.h>
2 #include <MFRC522.h>
3 #include <Wire.h>
4 #include <LiquidCrystal_I2C.h>
5
6 #define SS_PIN 5
7 #define RST_PIN 22
8 #define BUZZER_PIN 25
9 #define LED_GREEN 26
10 #define LED_RED 27
11
12 MFRC522 mfrc522(SS_PIN, RST_PIN);
13 LiquidCrystal_I2C lcd(0x27, 16, 2);
14
15 void setup() {
16     Serial.begin(115200);
17     SPI.begin();
18     mfrc522.PCD_Init();
19
20     lcd.init();
21     lcd.backlight();
22     lcd.print("Systeme_RFID");
23     lcd.setCursor(0, 1);
24     lcd.print("Pret");
25
26     pinMode(BUZZER_PIN, OUTPUT);
27     pinMode(LED_GREEN, OUTPUT);
28     pinMode(LED_RED, OUTPUT);
29
30     delay(1000);
31 }
32
33 void loop() {
34     if (!mfrc522.PICC_IsNewCardPresent()) return;
35     if (!mfrc522.PICC_ReadCardSerial()) return;
36
37     String uid = "";
38     for (byte i = 0; i < mfrc522.uid.size; i++) {
39         uid += String(mfrc522.uid.uidByte[i], HEX);
40     }
41
42     lcd.clear();

```

```

43  lcd.print("UID:_" + uid);
44
45  digitalWrite(BUZZER_PIN, HIGH);
46  digitalWrite(LED_GREEN, HIGH);
47  delay(200);
48  digitalWrite(BUZZER_PIN, LOW);
49  digitalWrite(LED_GREEN, LOW);
50
51  delay(1000);
52  lcd.clear();
53  lcd.print("Presentez_carte");
54  }

```

2. Montage 2 : Système Empreinte Digitale avec ESP32

Matériel nécessaire

- ◇ ESP32
- ◇ Capteur d'empreinte digitale (R307 ou FPM10A)
- ◇ Écran LCD 16x2 avec interface I2C
- ◇ Buzzer
- ◇ 2 LEDs (verte et rouge)
- ◇ 2 résistances 220
- ◇ Breadboard
- ◇ Câbles Dupont

Connexions entre ESP32 et capteur d'empreinte

TABLE 2 – Broches utilisées pour le système empreinte digitale

Pin Capteur	Pin ESP32	Description
VCC	3.3V	Alimentation 3.3V
GND	GND	Masse
TX (Out)	GPIO16 (RX2)	Transmission données
RX (In)	GPIO17 (TX2)	Réception commandes

Code de test empreinte digitale

```

1  #include <Adafruit_Fingerprint.h>
2  #include <SoftwareSerial.h>
3  #include <Wire.h>
4  #include <LiquidCrystal_I2C.h>
5
6  #define FINGERPRINT_RX 16
7  #define FINGERPRINT_TX 17
8  #define BUZZER_PIN 25
9  #define LED_GREEN 26

```

```

10 #define LED_RED 27
11
12 SoftwareSerial fingerSerial(FINGERPRINT_RX, FINGERPRINT_TX);
13 Adafruit_Fingerprint finger = Adafruit_Fingerprint(&fingerSerial);
14 LiquidCrystal_I2C lcd(0x27, 16, 2);
15
16 void setup() {
17     Serial.begin(115200);
18     fingerSerial.begin(57600);
19
20     lcd.init();
21     lcd.backlight();
22     lcd.print("Systeme_Empreinte");
23
24     pinMode(BUZZER_PIN, OUTPUT);
25     pinMode(LED_GREEN, OUTPUT);
26     pinMode(LED_RED, OUTPUT);
27
28     if (finger.verifyPassword()) {
29         lcd.setCursor(0, 1);
30         lcd.print("Capteur_OK");
31     } else {
32         lcd.setCursor(0, 1);
33         lcd.print("Erreur_capteur");
34         while (1);
35     }
36     delay(2000);
37 }
38
39 void loop() {
40     uint8_t p = finger.getImage();
41     if (p != FINGERPRINT_OK) return;
42
43     p = finger.image2Tz();
44     if (p != FINGERPRINT_OK) return;
45
46     p = finger.fingerFastSearch();
47     if (p == FINGERPRINT_OK) {
48         lcd.clear();
49         lcd.print("Empreinte_ID:");
50         lcd.setCursor(0, 1);
51         lcd.print(String(finger.fingerID));
52
53         digitalWrite(LED_GREEN, HIGH);
54         digitalWrite(BUZZER_PIN, HIGH);
55         delay(300);
56         digitalWrite(BUZZER_PIN, LOW);
57         delay(1000);
58         digitalWrite(LED_GREEN, LOW);
59     }
60     delay(50);
61 }

```

3. Intégration des deux systèmes

Tableau de connexions final

TABLE 3 – Connexions complètes pour le système intégré

Composant	Pin	ESP32	Fonction
5*RFID RC522	SDA	GPIO5	SPI Chip Select
	SCK	GPIO18	SPI Clock
	MOSI	GPIO23	SPI Data Out
	MISO	GPIO19	SPI Data In
	RST	GPIO22	Reset
2*Capteur Empreinte	TX	GPIO16	UART RX2
	RX	GPIO17	UART TX2
2*LCD I2C	SDA	GPIO21	I2C Data
	SCL	GPIO22	I2C Clock
Buzzer	Signal	GPIO25	Sortie audio
LED Verte	Anode	GPIO26	Validation
LED Rouge	Anode	GPIO27	Erreur

Procédure d'assemblage

1. **Étape 1** : Placer l'ESP32 au centre de la breadboard
2. **Étape 2** : Connecter l'alimentation 3.3V et GND aux rails de la breadboard
3. **Étape 3** : Installer le module RFID à gauche de l'ESP32
4. **Étape 4** : Installer le capteur d'empreinte à droite de l'ESP32
5. **Étape 5** : Connecter l'écran LCD en haut de la breadboard
6. **Étape 6** : Ajouter le buzzer et les LEDs
7. **Étape 7** : Vérifier toutes les connexions avec un multimètre

4. Explication du fonctionnement du système

Logique générale du système

- ★ L'étudiant présente sa carte RFID ou utilise le capteur d'empreinte digitale
- ★ Le module de lecture transmet l'identifiant unique (UID/Empreinte ID) à l'ESP32
- ★ L'ESP32 traite l'information et vérifie la validité du pointage
- ★ Le module affiche immédiatement le statut de confirmation ou d'erreur sur l'écran LCD
- ★ Les données de présence sont transmises via JSON à la base de données Firebase
- ★ Les professeurs peuvent consulter les données de présence actualisées via le Dashboard ReactJS

Résumé des broches principales

5. Configuration Firebase

Le système utilise Firebase Realtime Database pour le stockage et la gestion des données de présence. La configuration inclut :

TABLE 4 – Broches principales utilisées dans le système

Composant	Broche ESP32	Fonction
RFID RC522 (SDA)	GPIO5	Chip Select
RFID RC522 (RST)	GPIO22	Reset
Capteur Empreinte (TX)	GPIO16	Réception données
Capteur Empreinte (RX)	GPIO17	Transmission commandes
LCD I2C (SDA)	GPIO21	Données I2C
LCD I2C (SCL)	GPIO22	Horloge I2C
Buzzer	GPIO25	Signal sonore
LED Verte	GPIO26	Validation
LED Rouge	GPIO27	Erreur

Structure de la base de données

- ◇ **Utilisateurs** : Stocke les informations des étudiants (nom, ID, classe)
- ◇ **Présences** : Enregistre chaque pointage avec horodatage, méthode (RFID/Empreinte), salle
- ◇ **Statistiques** : Génère des rapports de fréquentation en temps réel

6. Tests et validation

Procédure de test

Test	Action	Résultat attendu
1	Alimenter le système	LEDs d'alimentation allumées, écran LCD affiche message initial
2	Présenter carte RFID valide	Bip sonore, LED verte allumée, écran affiche UID
3	Présenter carte invalide	Bip d'erreur, LED rouge allumée
4	Poser doigt enregistré	Bip sonore, LED verte, écran affiche ID empreinte
5	Poser doigt non enregistré	Bip d'erreur, LED rouge
6	Vérifier connexion WiFi	Message de confirmation sur écran et moniteur série
7	Envoi données Firebase	Vérifier dans la console Firebase l'arrivée des données

7. Analyse et résultats obtenus

Performances observées

- ◇ Le système RFID répond en moins de 200ms pour la lecture d'une carte
- ◇ Le capteur d'empreinte met environ 1-2 secondes pour l'identification
- ◇ La transmission des données vers Firebase s'effectue en temps réel
- ◇ L'interface LCD permet une rétroaction immédiate à l'utilisateur

Points forts

- ◇ Double méthode d'identification pour plus de flexibilité
- ◇ Interface utilisateur simple avec écran LCD et feedback sonore
- ◇ Communication fiable avec Firebase via WiFi
- ◇ Solution évolutive pour plusieurs salles

Limites et améliorations possibles

- ◇ Ajouter une synchronisation hors ligne pour les pannes de réseau
- ◇ Intégrer une batterie de secours pour l'alimentation
- ◇ Améliorer la sécurité des données biométriques
- ◇ Ajouter la reconnaissance faciale comme troisième méthode optionnelle

8. Conclusion

Ce rapport a présenté les étapes complètes de montage du système de présence automatique utilisant à la fois la technologie RFID et la reconnaissance d'empreintes digitales. Les deux montages fonctionnent indépendamment et peuvent être intégrés pour offrir une solution flexible et robuste.

Le système final permet une gestion automatisée et en temps réel des présences, avec une interface de supervision accessible aux enseignants. Ce projet illustre parfaitement les possibilités de l'IoT dans le domaine de l'éducation et constitue une base solide pour des développements futurs plus avancés.