

```
#include <SPI.h>          // Bibliothèque pour la communication SPI
#include <MFRC522.h>    // Bibliothèque pour le lecteur RFID MFRC522

// ----- Définition des broches -----
#define SS_PIN 10      // Broche SDA / SS du module RFID
#define RST_PIN 9       // Broche RESET du module RFID
#define LED_PIN 3        // LED verte (accès autorisé)
#define LED_PIN1 6       // LED rouge (accès refusé)
```

```
// ----- Crédation de l'objet RFID -----
MFRC522 mfrc522(SS_PIN, RST_PIN); // Instance du lecteur RFID
```

```
// ----- UID autorisé -----
byte uidAutorise[] = {0x92, 0x87, 0x37, 0x03}; // UID autorisé (à modifier)
byte tailleUIDAutorise = sizeof(uidAutorise); // Taille de l'UID autorisé
```

```
// ----- Fonction exécutée une seule fois -----
void setup() {
    Serial.begin(9600);      // Initialisation de la communication série
    SPI.begin();            // Initialisation du bus SPI
    mfrc522.PCD_Init();    // Initialisation du module RFID
```

```
pinMode(LED_PIN, OUTPUT); // LED verte en sortie
pinMode(LED_PIN1, OUTPUT); // LED rouge en sortie
```

```
digitalWrite(LED_PIN, LOW); // LED verte éteinte au démarrage
digitalWrite(LED_PIN1, LOW); // LED rouge éteinte au démarrage
```

```
    Serial.println("Approche une carte RFID...");
}
```

```
// ----- Boucle principale -----
void loop() {
```

```
    // Vérifie si une nouvelle carte est détectée
    if (!mfrc522.PICC_IsNewCardPresent()) return;
```

```
    // Lit l'UID de la carte détectée
    if (!mfrc522.PICC_ReadCardSerial()) return;
```

```
    // Affichage de l'UID dans le moniteur série
    Serial.print("UID détecté : ");
    for (byte i = 0; i < mfrc522.uid.size; i++) {
        Serial.print(mfrc522.uid.uidByte[i], HEX); // Affiche chaque octet en HEX
```

```
    Serial.print(" ");
}

Serial.println();
```

```
// ----- Comparaison des UID -----
bool autorise = true; // On suppose que l'accès est autorisé
```

```
// Vérifie si la taille de l'UID est correcte
if (mfrc522.uid.size != tailleUIDAutorise) {
    autorise = false;
}
else {
    // Compare chaque octet de l'UID
    for (byte i = 0; i < mfrc522.uid.size; i++) {
        if (mfrc522.uid.uidByte[i] != uidAutorise[i]) {
            autorise = false; // UID différent
            break;
        }
    }
}
```

```
// ----- Action selon le résultat -----
if (autorise) {
    Serial.println("Accès autorisé");
    digitalWrite(LED_PIN1, LOW); // Éteint LED rouge
    digitalWrite(LED_PIN, HIGH); // Allume LED verte
}
else {
    Serial.println("Accès refusé");
    digitalWrite(LED_PIN, LOW); // Éteint LED verte
    analogWrite(LED_PIN1, 255); // Allume LED rouge (PWM max)
}
```

```
// ----- Fin de communication RFID -----
mfrc522.PICC_HaltA(); // Arrête la communication avec la carte
mfrc522.PCD_StopCrypto1(); // Arrête le chiffrement
}
```