CODE FOR DIGITAL BUS PASS

```
#include <SPI.h>
#include <MFRC522.h>
#include <SoftwareSerial.h>
#include <EEPROM.h>

#define RST_PIN 9
#define SS_PIN 10

MFRC522 mfrc522(SS_PIN, RST_PIN);
byte accessUID[4] = {0x13, 0x4C, 0x10, 0xF5};
int GreenPin = 2;
int RedPin = 3;
int buzzerPin = 4;
int accessLimit = 2;  // Number of accesses allowed per day

SoftwareSerial mySerial(9, 10);  // RX, TX for GSM module

void setup() {
  pinMode(GreenPin, OUTPUT);
  pinMode(RedPin, OUTPUT);
  digitalWrite(RedPin, LOW);
  digitalWrite(GreenPin, LOW);
  pinMode(buzzerPin, OUTPUT);
  Serial.begin(9600);
  while (!Serial);

  SPI.begin();
  mfrc522.PCD_Init();
  delay(4);
  mfrc522.PCD_DumpVersionToSerial();
  Serial.println(F("Scan PICC to see UID, SAK, type, and data blocks..."));
```

```
  mySerial.begin(9600);

  // Initialize EEPROM
  EEPROM.write(0, 0);  // Initialize access count to 0
}

void sendSMS(String message) {
  if (Serial.available() > 0) {
  mySerial.println("AT+CMGF=1");
  delay(1000);
  mySerial.println("AT+CMGS=\"+919688113385\" "); // Replace with your phone number
  delay(1000);
  mySerial.print(message);
  delay(100);
  mySerial.println((char)26);
  delay(1000);
}
}


bool isCardValid() {
  int accessCount = EEPROM.read(0);  // Read access count from EEPROM
  return accessCount < accessLimit;
}

void accessGrantedAction() {
  int accessCount = EEPROM.read(0);  // Read access count from EEPROM
  if (isCardValid()) {
    accessCount++;  // Increment the access count
    EEPROM.write(0, accessCount);  // Write the updated access count to EEPROM
    digitalWrite(GreenPin, HIGH);
    tone(buzzerPin, 1000, 300);
    delay(100);
```

```
    digitalWrite(GreenPin, LOW);
    noTone(buzzerPin);
    Serial.println("Name:Sowmiya \n Access:Granted \n");
    sendSMS("Name:Sowmiya \n Access:Granted \n ");
  } else {
    Serial.println("Name:Sowmiya \n Card limit exceeded - Access Denied");
    accessDeniedAction();  // Implement access denied action
  }

  // Ensure LEDs are turned off after actions are completed
  digitalWrite(GreenPin, LOW);
  digitalWrite(RedPin, LOW);
}

void accessDeniedAction() {

  digitalWrite(RedPin, HIGH);
  digitalWrite(buzzerPin, HIGH);
  delay(200);
  digitalWrite(RedPin, LOW);
  noTone(buzzerPin);
  Serial.println("Name: Madhumitha \n Access:Denied\n");
  sendSMS("Access Denied");

}

void loop() {
  delay(100);
  digitalWrite(GreenPin, LOW);
  digitalWrite(RedPin, LOW);
  if (!mfrc522.PICC_IsNewCardPresent()) {
    return;
  }
```

```
  if (!mfrc522.PICC_ReadCardSerial()) {
    return;
  }

  if (mfrc522.uid.uidByte[0] == accessUID[0] && mfrc522.uid.uidByte[1] == accessUID[1] &&
mfrc522.uid.uidByte[2] == accessUID[2] && mfrc522.uid.uidByte[3] == accessUID[3]) {
    accessGrantedAction();
  } else {
    accessDeniedAction();
  }

  mfrc522.PICC_HaltA();
}
```