#include <SPI.h>

#include <MFRC522.h>

#define SS\_PIN 10

#define RST\_PIN 9

#define LED\_G 5

#define LED\_R 4

#define RELAY 3 //connect the relay to number 3 pin

#define BUZZER 2 // connect the buzzer to 2 pin

#define ACCESS\_DELAY 2000

#define DENIED\_DELAY 1000

MFRC522 mfrc522(SS\_PIN, RST\_PIN); // Create MFRC522 instance.

const char \*accepted\_cards[] = {

"C9 C0 DD 14",

"B2 3A 6B 2D",

// Add more here

};

const int total\_accepted\_cards = sizeof(accepted\_cards) / sizeof(accepted\_cards[0]);

void setup()

{

Serial.begin(9600); // Initiate a serial communication

SPI.begin(); // Initiate SPI bus

mfrc522.PCD\_Init(); // Initiate MFRC522

pinMode(LED\_G, OUTPUT);

pinMode(LED\_R, OUTPUT);

pinMode(RELAY, OUTPUT);

pinMode(BUZZER, OUTPUT);

noTone(BUZZER);

digitalWrite(RELAY, HIGH);

Serial.println("Put your card to the reader for scanning …");

Serial.println();

}

void loop()

{

// Look for new cards

if ( ! mfrc522.PICC\_IsNewCardPresent())

{

return;

}

// Select one of the cards

if ( ! mfrc522.PICC\_ReadCardSerial())

{

return;

}

//Show UID on serial monitor

Serial.print("UID tag :");

String content= "";

byte letter;

for (byte i = 0; i < mfrc522.uid.size; i++)

{

Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");

Serial.print(mfrc522.uid.uidByte[i], HEX);

content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));

content.concat(String(mfrc522.uid.uidByte[i], HEX));

}

Serial.println();

Serial.print("Message : ");

content.toUpperCase();

for (int i = 0; i < total\_accepted\_cards; i++) {

if (content.substring(1) == accepted\_cards[i]) {

Serial.println("Authorized access");

Serial.println();

delay(500);

digitalWrite(RELAY, HIGH);

digitalWrite(LED\_G, HIGH);

delay(ACCESS\_DELAY);

digitalWrite(RELAY, LOW);

digitalWrite(LED\_G, LOW);

return;

}

}

Serial.println("Authorized denied");

digitalWrite(LED\_R, HIGH);

tone(BUZZER, 300);

delay(DENIED\_DELAY);

digitalWrite(LED\_R, LOW);

noTone(BUZZER);

}