

02-MARZO-2023

ACTIVIDAD 7

ALUMNOS:

ARAIZA MISSAEL
CARDONA DENISSE

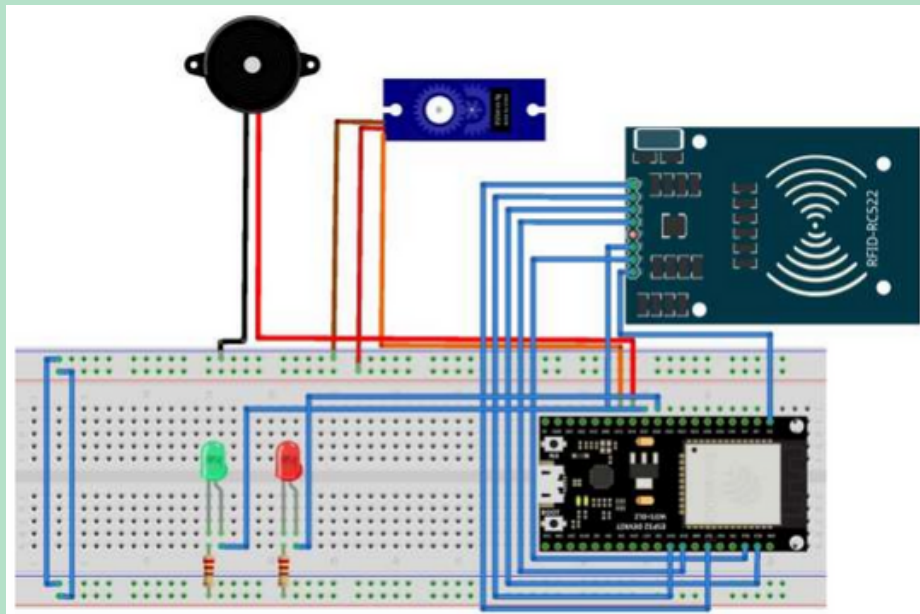
DSM 502

Objetivo: Entender el funcionamiento del modulo rfid

Materiales:

- 1 ESP32
- 1 buzzer
- 1 servomotor
- 1 modulo rfid
- 9 Cables dupont
- 2 resistencias 220 ohms
- 2 leds
- 1 Protoboard

Diagrama electrónico



Código en C

```
#include <Servo.h>
#include <SPI.h>
#include <MFRC522.h>
```

```
#define RST_PIN 9 // configuramos el pin 9 para
reset del mfrc522
#define SS_PIN 10 // configuramos el pin 53 para el
selector de chip para el caso del Arduino Mega
#define ledVerde 7 // led verde en el pin 4
#define ledRojo 8 // led rojo en el pin 5
#define buzzer 6
```

```
MFRC522 mfrc522(SS_PIN, RST_PIN); //
MFRC522::MIFARE_Key key;
```

```
Servo servo1;
```

```
byte tag2[4] = { 0xC3, 0x6E, 0x4D, 0x1F };
byte tag1[4] = { 0xE3, 0x91, 0x04, 0x0E };
//byte tag2[4] = {0x32, 0xDE, 0xAC, 0x1E};
```

```
byte tagActual[4];
```

```
void setup() {
  Serial.begin(9600); // velocidad del puerto serie
  while (!Serial)
    ; // comprobar si ya ha iniciado
```

```
SPI.begin();    // inicia la comunicacion SPI
mfrc522.PCD_Init(); //inicia la comunicacion con el
modulo RFID
servo1.attach(5); // asigna el pin para el Servo
pinMode(ledVerde, OUTPUT); // configuram como
salida
pinMode(ledRojo, OUTPUT);
}

void loop() {
  if (!mfrc522.PICC_IsNewCardPresent()) // comprobar
si esta presente la tarjeta
    return;

  // Selecciona una tarjeta
  if (!mfrc522.PICC_ReadCardSerial()) // leer la tarjeta
    return;

  Serial.print(F("Card UID:"));
  dump_byte_array(mfrc522.uid.uidByte,
mfrc522.uid.size); // obtiene los valores leidos de la
tarjeta

  servo1.write(90);    // coloca el servo en la posicion
cero
  digitalWrite(ledVerde, LOW); // apaga led
  digitalWrite(ledRojo, LOW);
}

void dump_byte_array(byte *buffer, byte bufferSize) {
  // realiza la lectura del codigo de la tarjeta y
```

```
comprueba
for (byte i = 0; i < bufferSize; i++) {    // extrae
valores del codigo
  Serial.print(buffer[i] < 0x10 ? " 0" : " ");
  Serial.print(buffer[i], HEX);
  tagActual[i] = buffer[i];
}
if (compararArray(tagActual, tag1)) { // comprobamos
el acceso para el primer usuario
  Serial.println(" Acceso Permitido...");
  digitalWrite(ledVerde, HIGH);
  servo1.write(180);
  tone(buzzer, 290);
  delay(500);
  noTone(buzzer);
  delay(2000);
} else if (compararArray(tagActual, tag2)) { //
comprobamos el acceso para el segundo usuario
  Serial.println(" Acceso Permitido...");
  digitalWrite(ledVerde, HIGH);
  servo1.write(180);
  tone(buzzer, 290);
  delay(500);
  noTone(buzzer);
  delay(2000);
} else {
  Serial.println(" Desconocido"); // si el codigo no esta
registrado denegar acceso
  digitalWrite(ledRojo, HIGH);
  tone(buzzer, 100);
  delay(100);
}
```

```
noTone(buzzer);
delay(100);
tone(buzzer, 100);
delay(100);
noTone(buzzer);
delay(1000);
}
}
```

```
boolean compararArray(byte array1[], byte array2[]) {
// comprobara codigo leido con el registrado
previamente
```

```
if (array1[0] != array2[0]) return (false);
if (array1[1] != array2[1]) return (false);
if (array1[2] != array2[2]) return (false);
if (array1[3] != array2[3]) return (false);
return (true);
}
```

Imagen de la actividad realizada

