

# Control Acceso RFID

IOT2

JOSUE ARMANDO RIVERA HERNANDEZ

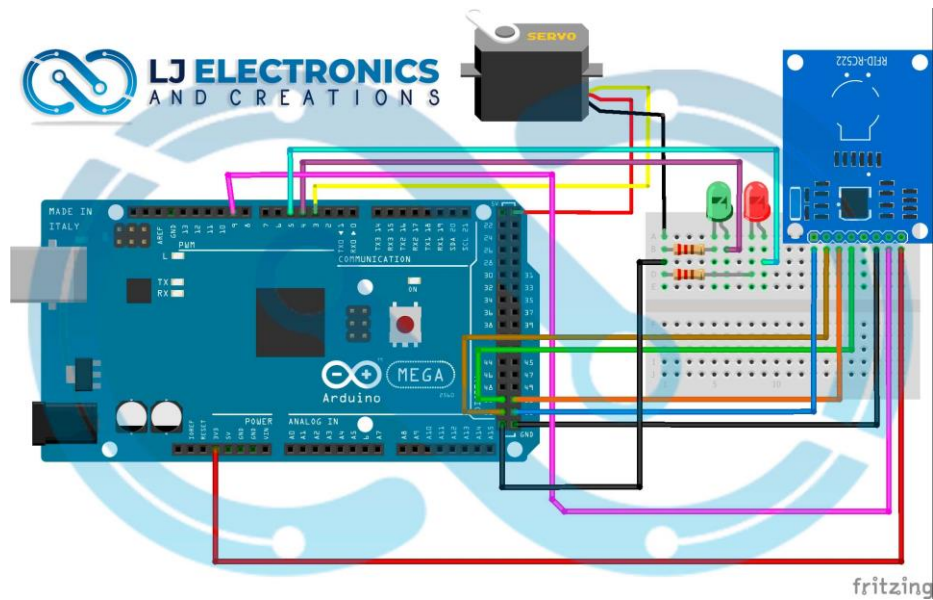
## Contenido

Propósito .....	2
Diagrama .....	2
Materiales .....	2
Código .....	3
Evidencias .....	5

## Propósito

Hacer un control de acceso con una placa de rfid, el cuál tenga el identificador de una tarjeta y cuando sea correcto se abra un servomotor, encienda un led y se mande un mensaje de acceso correcto y cuando no sea correcto encienda un led Rojo y mande un mensaje de acceso no concedido.

## Diagrama



## Materiales

- \* Arduino Mega
- \* RFID-RC522
- \* Servomotor
- \* Resistencia 220Ohmn
- \* Diodo LED
- \* Tarjeta y llaveros RFID
- \* Cables
- \*Protoborard

## Código

```
#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      6           // led verde en el pin 4
#define ledRojo       5          // led rojo en el pin 5

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

Servo servo1;
const int buzzer = 7;
byte tag2[4] = {0xA4, 0x98, 0x39, 0xA5};
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(3);    // 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 leídos de la tarjeta

servo1.write(3);          // coloca el servo en la posición cero
digitalWrite(ledVerde,LOW);      // apaga led
digitalWrite(ledRojo,LOW);
}

void dump_byte_array(byte *buffer, byte bufferSize) {      // realiza la
lectura del código de la tarjeta y comprueba
    for (byte i = 0; i < bufferSize; i++) {                // extrae valores del
código
        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(90);
        delay(1000);
        tone(buzzer,450);
        delay(400);
        noTone(buzzer);
    }
    else if(compararArray(tagActual,tag2)){                // comprobamos el
acceso para el segundo usuario
        Serial.println(" Acceso Permitido...");
        digitalWrite(ledVerde,HIGH);
        servo1.write(90);
        delay(1000);
    }
    else {
        Serial.println(" Desconocido" );                  // si el código no
esta registrado denegar acceso
        digitalWrite(ledRojo,HIGH);
        delay(1000);
        tone(buzzer,900);
        delay(200);
        noTone(buzzer);
        tone(buzzer,600);
        delay(200);
        noTone(buzzer);
    }
}

```

```

    }
}

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);
}

```

## Evidencias

