

Los lectores RFID(Radio Frequency IDentification) en la actualidad están teniendo bastante acogida en los sistemas de identificación, su uso abarca desde sistemas de seguridad, acceso de personal, identificación y logística de productos, como llaves de puertas eléctricas, entre otras aplicaciones.

Su principio de funcionamiento consiste en pasar un TAG, cerca de un lector RFID, el TAG tiene la capacidad de enviar información al lector. Dicha información puede ser desde un simple código o todo un paquete de información guardado en la memoria del Tag.

Los TAGs viene en diferentes modelos, los más comunes son en tarjetas y en llaveros, pero también vienen como etiquetas adhesivas e incluso ya viene incrustados en algunos productos. Los Tags tienen internamente una antena y un microchip, encargado de realizar todo el proceso de comunicación, la energía lo obtiene de la señal de radiofrecuencia, que si bien la energía en la señal es pequeña, es suficiente para hacer trabajar el microchip, esto es la razón por la que es necesario acercarlos a una pequeña distancia generalmente menor a 10 cm. Pero existen Tags activos, que incorporan baterías, estos tiene alcance de varios metros de distancia.

Módulo RFID RC522

Es el que usaremos en este tutorial, este módulo funciona como Lector y Grabador de Tags RFID.



Este módulo utiliza un sistema de modulación y demodulación de 13.56MHz, frecuencia que en la actualidad utiliza la tecnología RFID.

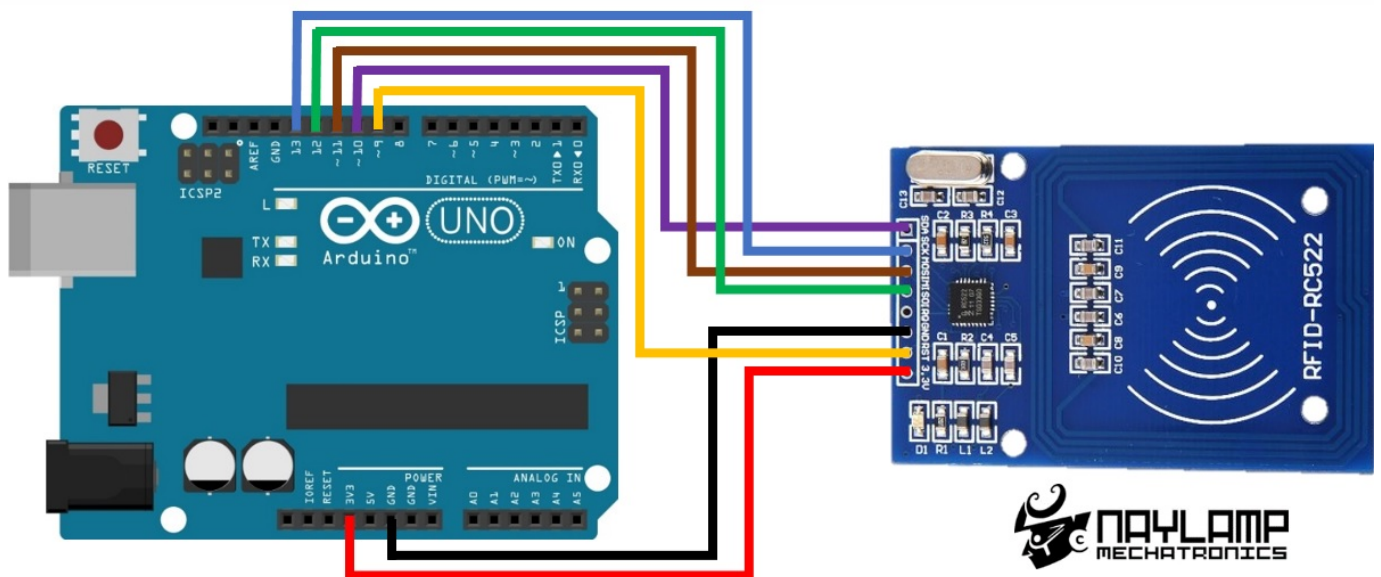
EL módulo se comunica por SPI, por lo que se puede implementar con cualquier microcontrolador con interfaz SPI, como un Arduino.

Bien, empecemos a armar nuestro proyecto:

1. Conexión del entre el módulo RFID y Arduino

Módulo RC522	Arduino Uno, Nano	Arduino Mega
SDA (SS)	10	53

Módulo RC522	Arduino Uno, Nano	Arduino Mega
SCK	13	52
MOSI	11	51
MISO	12	50
IRQ	No conectado	No conectado
GND	GND	GND
RST	9	9
3.3V	3.3V	3.3V



Como se observa en la conexión, el módulo trabaja con un voltaje de 3.3V, por lo que la parte lógica también debería trabajar con el mismo nivel de voltaje, para prueba y testeo se puede conectar directamente a los pines del Arduino (nivel TTL 5V), pero se recomienda usar convertidores de niveles de voltaje.

2. Programación del Módulo RC522: Lectura del código de identificación

Para poder trabajar el Modulo en Arduino es necesario descargar su librería correspondiente, la que usaremos será una de las más comunes, la desarrollada por Miguel Balboa.

Librería RFID

Una vez descargada, importamos la librería a nuestro IDE de Arduino, con esto estamos listos para programar.

A continuación se muestra un sketch para leer el código de identificación de nuestros Tags

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

#define RST_PIN 9 //Pin 9 para el reset del RC522
#define SS_PIN 10 //Pin 10 para el SS (SDA) del RC522
MFRC522 mfrc522(SS_PIN, RST_PIN); //Creamos el objeto para el RC522

void setup() {
    Serial.begin(9600); //Iniciamos la comunicación serial
    SPI.begin(); //Iniciamos el Bus SPI
    mfrc522.PCD_Init(); // Iniciamos el MFRC522
    Serial.println("Lectura del UID");
}

void loop() {
    // Revisamos si hay nuevas tarjetas presentes
    if ( mfrc522.PICC_IsNewCardPresent() )
    {
        //Seleccionamos una tarjeta
        if ( mfrc522.PICC_ReadCardSerial() )
        {
            // Enviamos serialemente su UID
            Serial.print("Card UID:");
            for (byte i = 0; i < mfrc522.uid.size; i++) {
                Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
                Serial.print(mfrc522.uid.uidByte[i], HEX);
            }
            Serial.println();
            // Terminamos la lectura de la tarjeta actual
            mfrc522.PICC_HaltA();
        }
    }
}
```

Como se observa es fácil de entender, pero expliquemos las funciones referentes al módulo RC522

Instanciar el RC522

```
#define RST_PIN 9 //Pin 9 para el reset del RC522
#define SS_PIN 10 //Pin 10 para el SS (SDA) del RC522
MFRC522 mfrc522(SS_PIN, RST_PIN);
```

Es necesario utilizar la librería MFRC522.h, solo es necesario especificar los pines Reset y SDA(SS) del módulo, los demás pines trabajan con los pines SPI del Arduino.

Iniciar el RC522

```
SPI.begin(); //Iniciamos el Bus SPI
mfrc522.PCD_Init(); // Iniciamos el MFRC522
```

Esta función inicia y configura al RC522 para su posterior lectura, solo es necesario llamarlo una vez por lo que generalmente se lo llama en void setup()

Ver si hay una tarjeta presente

```
mfrc522.PICC_IsNewCardPresent()
```

Esta función nos devuelve verdadero o falso dependiendo si hay una tarjeta presente cerca al módulo RC522.

Seleccionar una tarjeta para la lectura

```
mfrc522.PICC_ReadCardSerial()
```

Se llama a esta función cuando queremos comunicarnos con una tarjeta, nos devuelve un valor verdadero si logra seleccionar una tarjeta para la lectura, de lo contrario nos retorna un valor falso.

Obtener el tamaño del código de identificación

```
mfrc522.uid.size
```

Nos retorna el tamaño en Bytes del código de identificación de la tarjeta seleccionada.

Leer el código de identificación

```
mfrc522.uid.uidByte
```

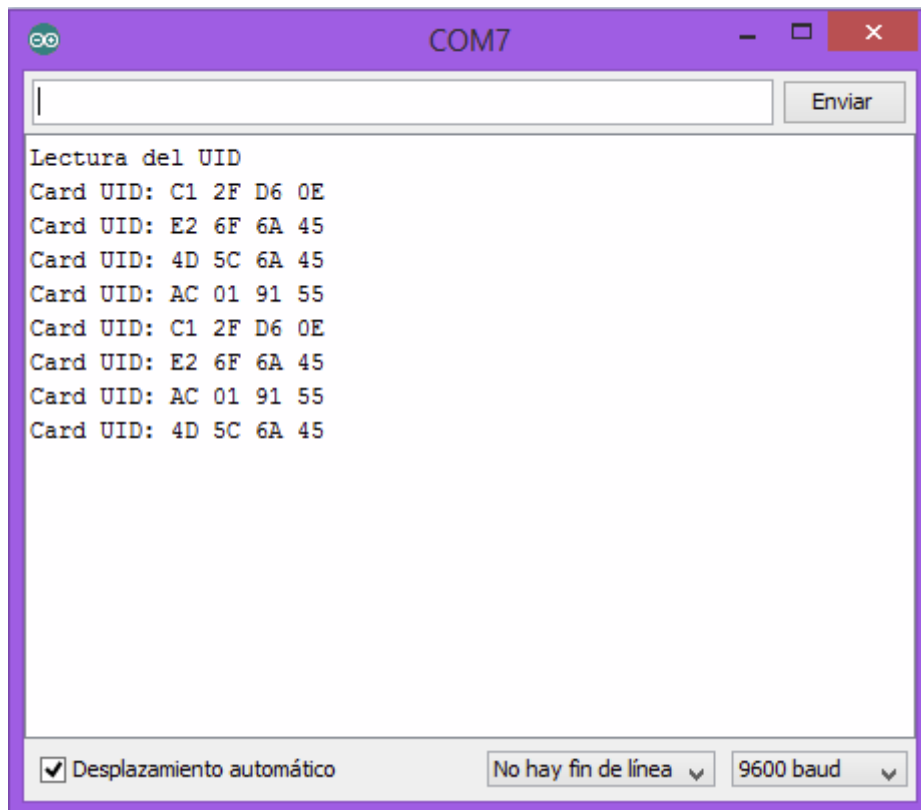
Para acceder al código de identificación es recomendable indexar la variable por ejemplo mfrc522.uid.uidByte[0] para el byte en la posición inicial.

Finalizar la lectura.

```
mfr522.PICC_HaltA();
```

Con esta función le indicamos que hemos terminado la lectura de la tarjeta presente y `IsNewCardPresent()` devolverá falso para esta tarjeta mientras no se retire. Si después de retirar la tarjeta, si se ingresa nuevamente la misma tarjeta se le considera como una nueva tarjeta.

En el sketch, el código de identificación lo enviamos por el puerto serial, para visualizar el código es necesario abrir el Monitor Serial del IDE de Arduino.



3. Una Aplicación Simple: Control de acceso usando RFID

Bien, visto el ejemplo anterior, ahora apliquémoslo para un control de acceso, que puedes aplicarse en una puerta eléctrica, caja fuerte, sistema de encendido, desactivar alarmas, etc.

Para este ejemplo estamos usando 4 Tags de los cuales solo dos deben tener acceso al sistema. Para implementar esto partimos del sketch anterior y solo le agregamos unas líneas más al final del programa para comparar los códigos y determinar si tienen o no acceso.

A continuación se muestra el Sketch.

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

#define RST_PIN 9 //Pin 9 para el reset del RC522
#define SS_PIN 10 //Pin 10 para el SS (SDA) del RC522
MFRC522 mfrc522(SS_PIN, RST_PIN); ///Creamos el objeto para el RC522

void setup() {
  Serial.begin(9600); //Iniciamos La comunicacion serial
  SPI.begin(); //Iniciamos el Bus SPI
  mfrc522.PCD_Init(); // Iniciamos el MFRC522
  Serial.println("Control de acceso:");
}

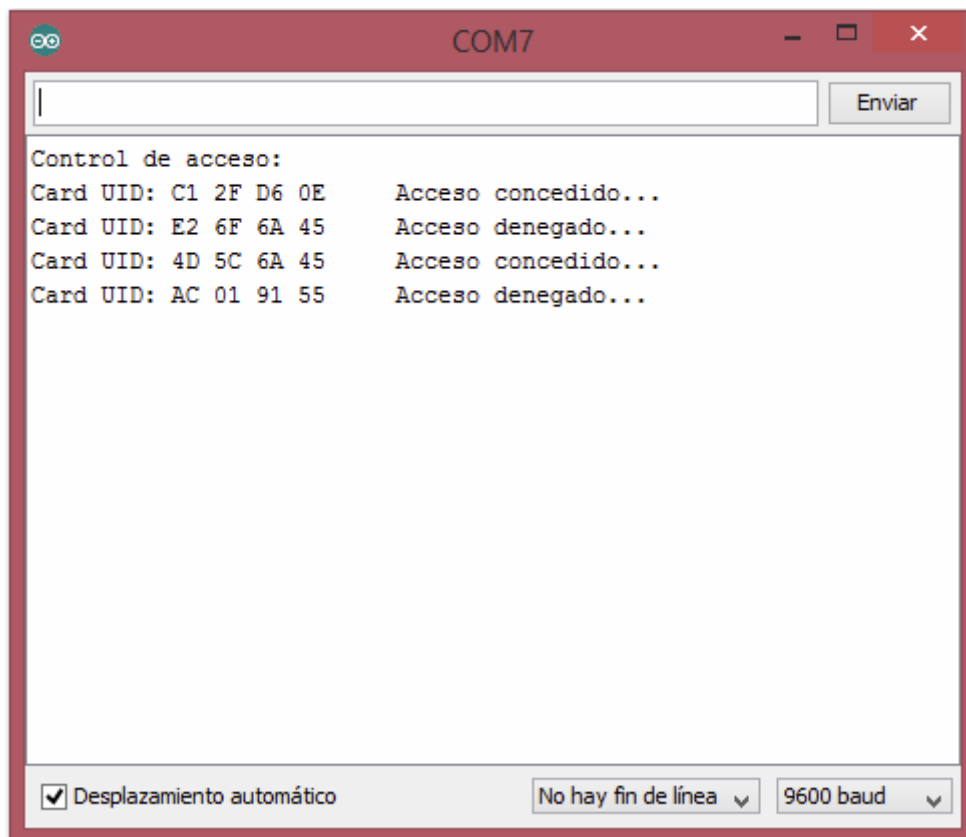
byte ActualUID[4]; //almacenará el código del Tag leído
byte Usuario1[4]= {0x4D, 0x5C, 0x6A, 0x45} ; //código del usuario 1
byte Usuario2[4]= {0xC1, 0x2F, 0xD6, 0x0E} ; //código del usuario 2
void loop() {
  // Revisamos si hay nuevas tarjetas presentes
  if ( mfrc522.PICC_IsNewCardPresent())
  {
    //Seleccionamos una tarjeta
    if ( mfrc522.PICC_ReadCardSerial())
    {
      // Enviamos serialmente su UID
      Serial.print(F("Card UID:"));
      for (byte i = 0; i < mfrc522.uid.size; i++) {
        Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
        Serial.print(mfrc522.uid.uidByte[i], HEX);
        ActualUID[i]=mfrc522.uid.uidByte[i];
      }
      Serial.print(" ");
      //comparamos los UID para determinar si es uno de nuestros usuarios
      if(compareArray(ActualUID,Usuario1))
        Serial.println("Acceso concedido...");
      else if(compareArray(ActualUID,Usuario2))
        Serial.println("Acceso concedido...");
      else
        Serial.println("Acceso denegado...");

      // Terminamos la lectura de la tarjeta tarjeta actual
      mfrc522.PICC_HaltA();
    }
  }
}

//Función para comparar dos vectores
```

```
boolean compareArray(byte array1[],byte array2[])
{
  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);
}
```

Y el resultado lo podemos observar al pasar los Tags y visualizarlo en el monitor serial.



En lugar de enviarlo serialmente la confirmación se puede programar para activar una salida digital, que puede estar conectado a una chapa eléctrica, una alarma, etc. dependerá de la aplicación en particular que están trabajando.

Es bueno aclarar que si bien cumple el objetivo, no es una forma 100% segura, cualquiera que sepa del tema podría clonar las tarjetas. Para una mayor seguridad es necesario trabajar con los bloques de la memoria interna del TAG, pudiéndose configurar una clave para la lectura, además se podría trabajar con un código mayor a los 4 bytes del UID, incluso se podrían trabajar con varios datos. (La lectura y escritura de bloques de memoria se verá en un próximo tutorial)

Pueden adquirir los materiales usados en este tutorial en nuestra tienda