

MÓDULO DE LECTURA Y ESCRITURA RFID RC522 13.56MHz

AR-RFID



MÓDULO DE LECTURA Y ESCRITURA RFID RC522 13.56MHz

AR-RFID

DESCRIPCIÓN

La tecnología de identificación por radiofrecuencia o RFID (del inglés, Radio Frequency Identification) es un sistema capaz de identificar objetos a través de un identificador único (UID) como si fuera un documento nacional de identidad (DNI) gracias a la radiofrecuencia. El sistema RFID consta de dos componentes principales: una etiqueta RFID o transpondedor y un lector RFID o transceptor. Esta señal es captada por las etiquetas dentro del alcance, las cuál responden transmitiendo la información que almacenada que, finalmente, es captada y decodificada por el lector RFID.



ESPECIFICACIONES

PÁRAMETROS	DESCRIPCIÓN
Modelo	MF522
Frecuencia de operacion	13.56Mhz
Protocolo de comunicacion	SPI
Velocidad de datos máxima	10Mbit/s
Voltaje de alimentación	3.3 V
Corriente de funcionamiento	13-26 mA
Corriente máxima	30 mA
Distancia de lectura	0 a 60 mm
Tipos de tarjetas compatibles	Mifare S50, S70 Mifare 1, MIFARE Ultralight, Mifare Pro, Mifare DESFire
Temperatura de funcionamiento	-20 a 80°C
Corriente en espera (Isb)	10-13mA a 3,3V
Corriente en modo reposo (Ism)	<80uA
Incluye pines	Llavero y tarjeta
Dimensiones	40x60mm
Peso total	20g

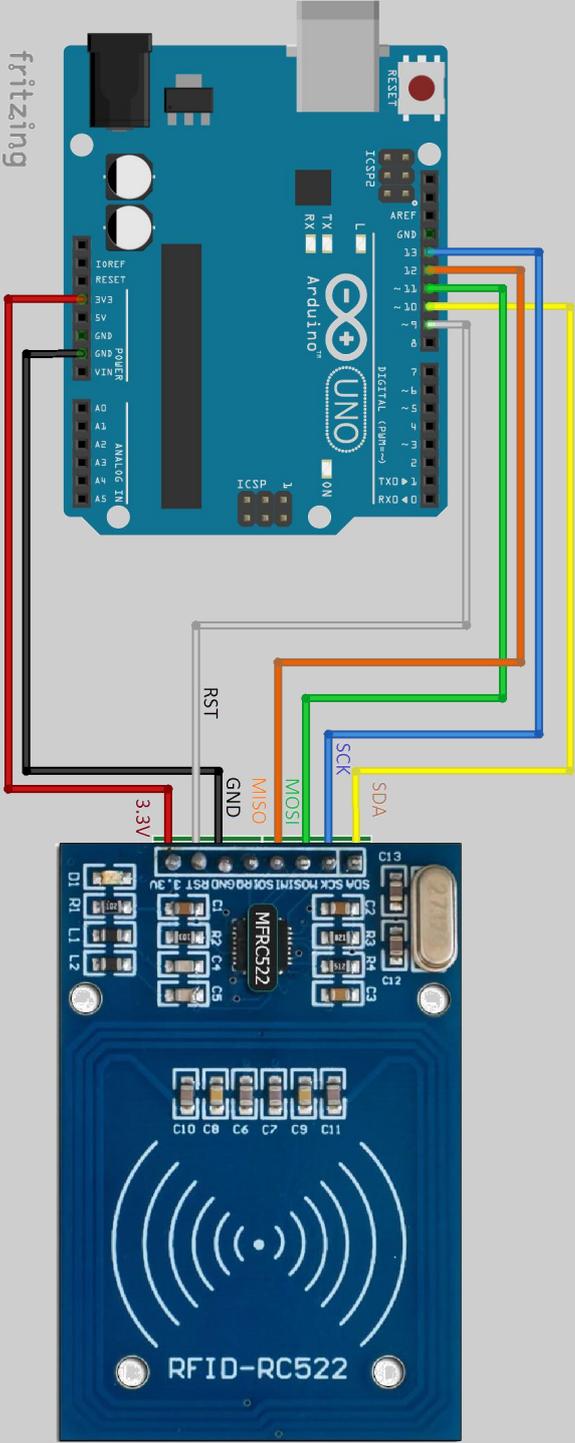
TARJETA TAG RFID

PÁRAMETROS	DESCRIPCIÓN
Tipo	Tarjeta RFID
Modelo	MFS50
Frecuencia	13,56 MHz
Protocolo	ISO/IEC 14443 tipo A
Velocidad en baudios	106 kbps
Rango de lectura/escritura	0-5 cm (aprox.)
EEPROM	1 Kbytes
Identificador único	4 bytes
Tiempo de lectura	100 ms
Ciclos de escritura	100.000
Dimensiones	86 mm x 54 mm x 1 mm
Color	Blanco
Material	PVC

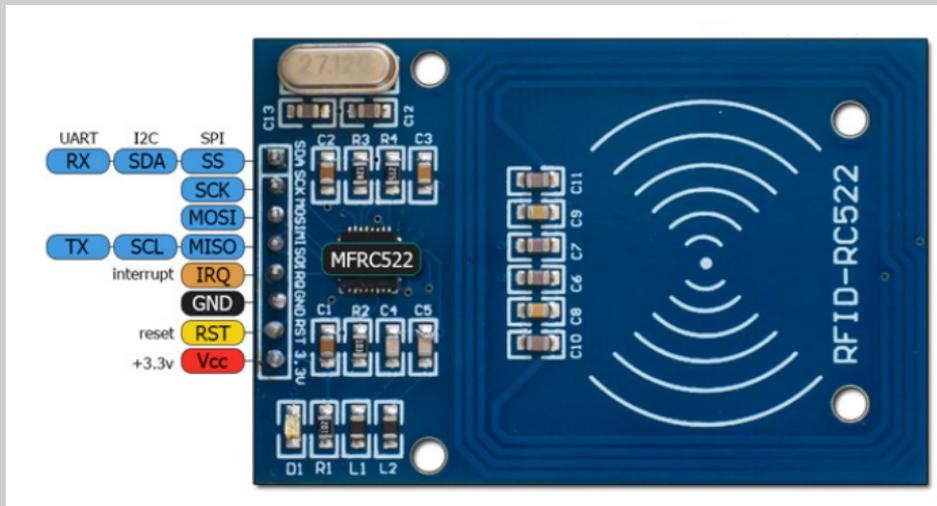
LLAVERO TAG RFID

PÁRAMETROS	DESCRIPCIÓN
Tipo	Llavero RFID
Frecuencia	13.56 MHz
Material	ABS/PVC
EEPROM	1 Kbytes
Rango de lectura/escritura	0-5cm (Aprox. Depende del lector)
Protocolo	ISO14443A
Identificador único	4 Bytes
Velocidad en baudios	106 Kbps

DIAGRAMA DE CONEXIÓN



PINOUT



VCC: pin de alimentación del lector RFID RC522. Admite un voltaje de alimentación entre 2.5V y 3.3V.

RST: es un pin para encender y apagar el módulo. Mientras el pin esté en estado LOW se mantendrá apagado sin apenas consumir. Cuando el estado cambia a HIGH el RC522 se reinicia.

GND: pin de tierra o GND.

IRQ: pin de interrupción que alerta al microcontrolador cuando una etiqueta RFID se acerca al lector RFID RC522.

MISO/SCL/TX: este pin tiene tres funciones. Cuando la interfaz SPI está habilitada funciona como salida de esclavo y entrada de máster. Cuando está activada la interfaz I2C funciona como señal de reloj y como salida serie cuando la interfaz UART está habilitada.

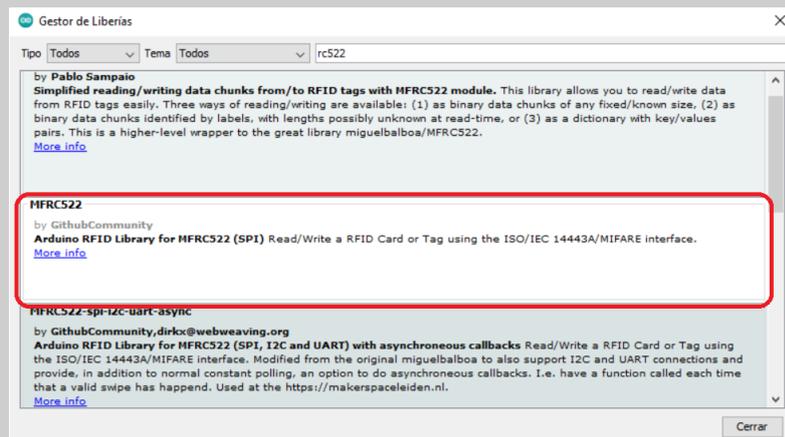
MOSI: entrada en la interfaz SPI.

SCK: señal de reloj de la interfaz SPI.

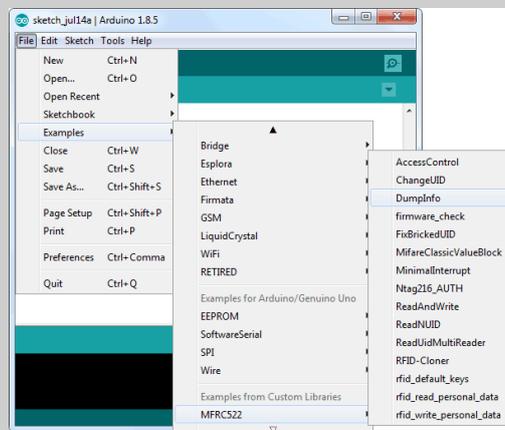
SS/SDA/RX: el pin actúa como entrada de señal cuando la interfaz SPI está habilitada. Si la interfaz I2C está activa actúa como entrada de datos y como entrada de datos serie cuando la interfaz UART está habilitada.

Instalación librería MFRC522

Para el siguiente programa se utilizan las librerías SPI para la comunicación, y MFRC522 que contiene procesos específicos del módulo RFDI-RC55 mismas que puedes instalar entrando en la pestaña de Herramientas, seleccionando Administrar Bibliotecas, y posteriormente Gestor de Librerías, donde puedes buscar por nombre la librería y te dará el estatus de la librería, si no está instalada, aparecerá la opción de instalar del lado derecho.



Utilizaremos el ejemplo Dumpinfo



```

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

#define RST_PIN      9           // Configurable, see typical pin
layout above
#define SS_PIN       10          // Configurable, see typical pin
layout above

MFRC522 mfrc522(SS_PIN, RST_PIN); // Create MFRC522 instance

void setup() {
  Serial.begin(9600);           // Initialize serial
communications with the PC
  while (!Serial);             // Do nothing if no serial port
is opened (added for Arduinos based on ATMEGA32U4)
  SPI.begin();                 // Init SPI bus
  mfrc522.PCD_Init();          // Init MFRC522
  delay(4);                     // Optional delay. Some
board do need more time after init to be ready, see Readme
  mfrc522.PCD_DumpVersionToSerial(); // Show details of PCD -
MFRC522 Card Reader details
  Serial.println(F("Scan PICC to see UID, SAK, type, and data
blocks..."));
}

void loop() {
  // Reset the loop if no new card present on the sensor/reader.
This saves the entire process when idle.
  if ( ! mfrc522.PICC_IsNewCardPresent() ) {
    return;
  }

  // Select one of the cards
  if ( ! mfrc522.PICC_ReadCardSerial() ) {
    return;
  }

  // Dump debug info about the card; PICC_HaltA() is
automatically called
  mfrc522.PICC_DumpToSerial (&(mfrc522.uid));
}

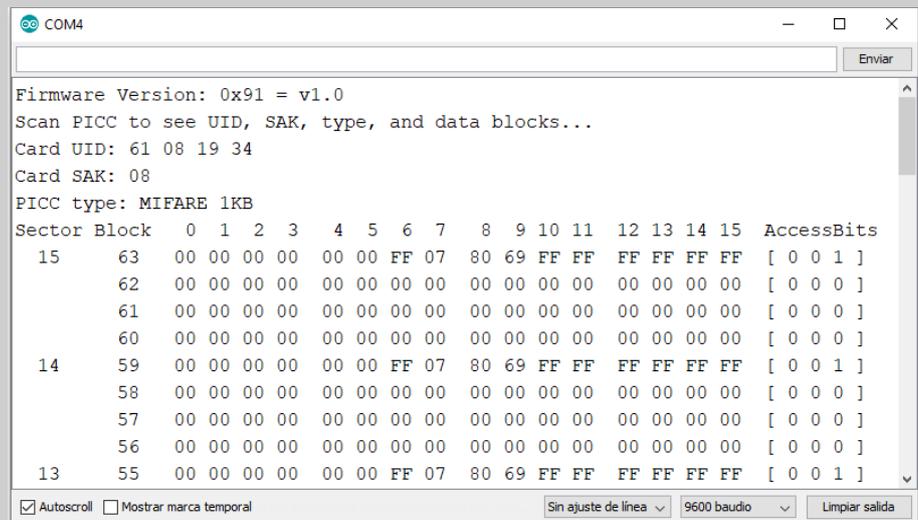
```

Este programa no escribe ningún dato en la etiqueta RFID. Simplemente lee si puede la etiqueta RFID y muestra la información por el monitor serie.

Es muy interesante para saber qué información hay almacenada dentro de la etiqueta RFID antes de usarla.



Tienes que dejar la tarjeta junto al lector RFID RC522 hasta que se muestre toda la información por el monitor serie. Todo eso que aparece son los 1024 bytes (1 Kb) de memoria que tiene la tarjeta.



Programa lectura de UID

```
#include <SPI.h> // incluye libreria bus SPI
#include <MFRC522.h> // incluye libreria especifica
para MFRC522

#define RST_PIN 9 // constante para referenciar
pin de reset
#define SS_PIN 10 // constante para referenciar
pin de slave select

MFRC522 mfrc522(SS_PIN, RST_PIN); // crea objeto mfrc522 enviando
pines de slave select y reset

void setup() {
  Serial.begin(9600); // inicializa comunicacion por
monitor serie a 9600 bps
  SPI.begin(); // inicializa bus SPI
  mfrc522.PCD_Init(); // inicializa modulo lector
}
```

```

void loop() {
    if ( ! mfrc522.PICC_IsNewCardPresent()) // si no hay una tarjeta
presente
        return; // retorna al loop
esperando por una tarjeta

    if ( ! mfrc522.PICC_ReadCardSerial()) // si no puede obtener
datos de la tarjeta
        return; // retorna al loop
esperando por otra tarjeta

    Serial.print("UID:"); // muestra texto
UID:
    for (byte i = 0; i < mfrc522.uid.size; i++) { // bucle recorre
de a un byte por vez el UID
        if (mfrc522.uid.uidByte[i] < 0x10){ // si el byte
leido es menor a 0x10
            Serial.print(" 0"); // imprime espacio en
blanco y numero cero
        }
        else{ // sino
            Serial.print(" "); // imprime un espacio en
blanco
        }
        Serial.print(mfrc522.uid.uidByte[i], HEX); // imprime el
byte del UID leido en hexadecimal
    }
    Serial.println(); // nueva linea
    mfrc522.PICC_HaltA(); // detiene comunicacion
con tarjeta
}

```



```

COM4
UID: 23 0E 76 95
UID: 23 0E 76 95
UID: 21 35 B4 4C
UID: 21 35 B4 4C

```

Programa simple control de acceso

```
#include <SPI.h> // incluye libreria bus SPI
#include <MFRC522.h> // incluye libreria especifica
para MFRC522

#define RST_PIN 9 // constante para referenciar
pin de reset
#define SS_PIN 10 // constante para referenciar
pin de slave select

MFRC522 mfrc522(SS_PIN, RST_PIN); // crea objeto mfrc522 enviando
pines de slave select y reset

byte LecturaUID[4]; // crea array para
almacenar el UID leido
byte Usuario1[4]= {0x23, 0x0E, 0x76, 0x95}; // UID de tarjeta
leido en programa 1
byte Usuario2[4]= {0x21, 0x35, 0xB4, 0x4C}; // UID de llavero
leido en programa 1

void setup() {
  Serial.begin(9600); // inicializa comunicacion por
monitor serie a 9600 bps
  SPI.begin(); // inicializa bus SPI
  mfrc522.PCD_Init(); // inicializa modulo lector
  Serial.println("Listo"); // Muestra texto Listo
}

void loop() {
  if ( ! mfrc522.PICC_IsNewCardPresent() // si no hay una
tarjeta presente
  return; // retorna al
loop esperando por una tarjeta
  if ( ! mfrc522.PICC_ReadCardSerial() // si no puede
obtener datos de la tarjeta
  return; // retorna al
loop esperando por otra tarjeta

  Serial.print("UID:"); // muestra texto
UID:
  for (byte i = 0; i < mfrc522.uid.size; i++) { // bucle recorre
de a un byte por vez el UID
    if (mfrc522.uid.uidByte[i] < 0x10){ // si el byte
leido es menor a 0x10
      Serial.print(" 0"); // imprime
espacio en blanco y numero cero
    }
    else{ // sino
      Serial.print(" "); // imprime un
espacio en blanco
    }
  }
}
```

```

Serial.print(mfrc522.uid.uidByte[i], HEX);          // imprime el
byte del UID leido en hexadecimal
    LecturaUID[i]=mfrc522.uid.uidByte[i];          // almacena en
array el byte del UID leido
    }
    Serial.print("\t");                             // imprime un
espacio de tabulacion
    if(comparaUID(LecturaUID, Usuario1))           // llama
a funcion comparaUID con Usuario1
        Serial.println("Bienvenido Usuario 1"); // si retorna
verdadero muestra texto bienvenida
    else if(comparaUID(LecturaUID, Usuario2)) // llama a
funcion comparaUID con Usuario2
Serial.println("Bienvenido Usuario 2");           // si retorna verdadero
muestra texto bienvenida
        else                                       // si
retorna falso
            Serial.println("No te conozco");      //
muestra texto equivalente a acceso denegado
                mfrc522.PICC_HaltA();             // detiene
comunicacion con tarjeta
    }
boolean comparaUID(byte lectura[],byte usuario[]) // funcion
comparaUID
{
    for (byte i=0; i < mfrc522.uid.size; i++){    // bucle recorre
de a un byte por vez el UID
        if(lectura[i] != usuario[i])             // si
byte de UID leido es distinto a usuario
            return(false);                       // retorna falso
        }
        return(true);                            // si los
4 bytes coinciden retorna verdadero
    }
}

```

```

COM4
Listo
UID: 23 0E 76 95      No te conozco
UID: 23 0E 76 95      No te conozco
UID: 21 35 B4 4C      No te conozco

```

Resultados al incluir el UID de las tarjetas de acceso

COM4

|

Listo

UID: 23 0E 76 95 Bienvenido Usuario 1

UID: 21 35 B4 4C Bienvenido Usuario 2

REALIZÓ: VEJL

REVISÓ: GAC