

PRÁCTICA 6.2: LECTURA DE ETIQUETA RFID

CÓDIGO

```
#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 serialamente 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();
        }
    }
}
```

FUNCIONAMIENTO

Primero incluimos las librerías, la primera nos permite comunicarnos con los dispositivos SPI y con la segunda leemos y escribimos una tarjeta o etiqueta RFID utilizando la interfaz ISO/IEC 14443A/MIFARE.

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

Para empezar especificamos los pines Reset y SDA(SS) del módulo, los demás pines no hace falta especificarlos, ya que trabajan con los pines SPI del Arduino. Y creamos el objeto para el RC522.

```
#define RST_PIN 9
#define SS_PIN 10
MFRC522 mfrc522(SS_PIN, RST_PIN);
```

Después empieza el `void setup()`, este en la primera línea inicializa una comunicación en serie a una velocidad de 9600 bauds. En la segunda línea iniciamos el Bus SPI. En la siguiente línea la función `mfrc522.PCD_Init()`; inicia y configura al RC522 para su posterior lectura. Y en la última fila del `void setup()` hacemos una impresión por pantalla.

```
void setup() {
  Serial.begin(9600);
  SPI.begin();
  mfrc522.PCD_Init();
  Serial.println("Lectura del UID");
}
```

A continuación abrimos un `void loop()`, dentro de este lo primero que hacemos es un `if (mfrc522.PICC_IsNewCardPresent())` esta función nos devuelve verdadero o falso dependiendo si hay una tarjeta presente cerca al módulo RC522. Dentro de este *if* haremos otro condicional con la función `mfrc522.PICC_ReadCardSerial()` esta función sirve para comunicarnos con una tarjeta, lo que hace es devolver un valor verdadero si logra seleccionar una tarjeta para la lectura, de lo contrario nos retorna un valor falso. Después creamos un bucle *for* para imprimir por pantalla el código de identificación usando la función `mfrc522.uid.uidByte[i]` donde la variable *i* es la que se ve recorriendo en el bucle. Y finalmente la función `mfrc522.PICC_HaltA()` indicamos que hemos terminado la lectura de la tarjeta presente y `IsNewCardPresent()` devolverá falso para esta tarjeta mientras no se retire. El código del `void loop()` es el siguiente:

```
void loop() {  
  if ( mfrc522.PICC_IsNewCardPresent())  
  {  
    if ( mfrc522.PICC_ReadCardSerial())  
    {  
      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();  
      mfrc522.PICC_HaltA();  
    }  
  }  
}
```