

- Presentar los RFID.
- Comprender su utilidad.
- Conectando un sensor RFID sencillo a Arduino.

MATERIAL REQUERIDO.



Arduino UNO o equivalente.



Un Lector de RFID y alguna tarjeta, llavero o tag que podamos usar.

IDENTIFICANDO ELEMENTOS

Hay cosas que nos parecen tan normales que al final ni siquiera reparamos en ellas, porque nos parece que no hay otra manera de hacer las cosas, o simplemente porque no dedicamos un minuto a pensar en cosas que damos por supuestas.

Por ejemplo, nos parece normal llegar a un supermercado y que la cajera identifique los productos con un código de barras de forma inmediata y lea su precio y características, pero esto es algo que a nuestros abuelos les hubiera parecido .

Los códigos de barras (*o Barcodes*) están muy extendidas por todos los supermercados y otras tiendas para identificar con luz un cierto producto. Son rápidos baratos y espectaculares. Solo necesitan una etiqueta con un código en forma de barras que una pistola láser identifica con rapidez.

Los códigos de barras son muy baratos, justo un papelito con barras de distinta anchura o bien impresos en la envoltura del producto, pero también tienen unos ciertos inconvenientes. Si se rayan se tornan ilegibles y esto es relativamente frecuente y además no se pueden reescribir.

Pero el tema de esta sesión no son los códigos de barras, sino más bien su sucesor, las etiquetas **RFID** (*O RFID tags*).

Para empezar ya estás acostumbrado a su uso, aunque quizás no te hayas dado cuenta. Son esas etiquetas que te pitan a destiempo cuando sales de las grandes superficies, o los antirrobo en prendas de las tiendas de ropa, que se identifican mediante un arco.

Hace ya mucho tiempo que los códigos de barras están obsoletos, pero son tan baratos y una tecnología tan bien resuelta y establecida que nadie quiere oír hablar de cambiarla, pero más pronto que tarde los **RFID** van a sustituir a los códigos de barras.

RFID son las iniciales de identificación por radio frecuencia en inglés (*Radio Frequency IDentification*)

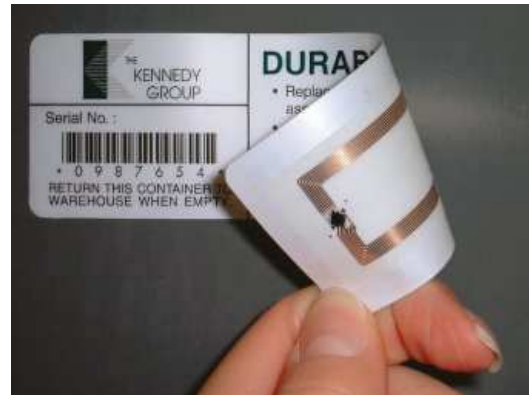
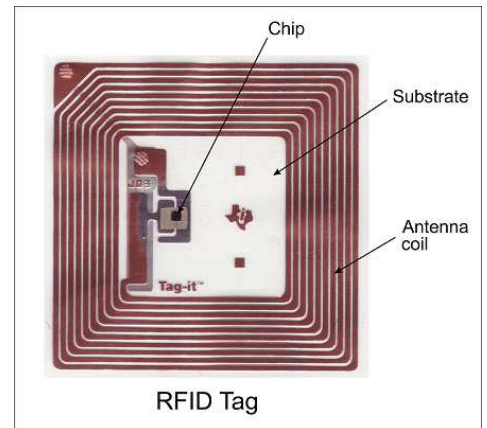
¿Y que son los RFID?

Bueno son simplemente un chip digital minúsculo de memoria, que puede contener una identificación en forma de código e incluso información de diferentes cuestiones grabados en él. Son como un pendrive pero pequeño y por radio, pero además tiene una antena que permite dos cosas poco frecuentes, una identificarlos por Radio Frecuencia (*RF*) sin contacto a distancias de entre unos pocos centímetros y varios metros, mediante un receptor adecuado, y la otra es que la propia señal de RF que usamos para identificarlos es capaz de alimentar el chip de la etiqueta **RFID** sin necesidad de batería.

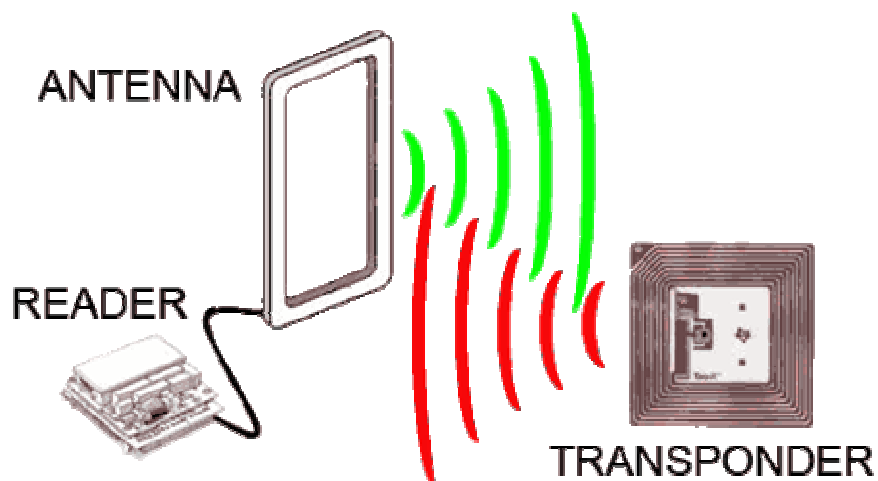


Cuando la etiqueta **RFID** (*Tag*) no lleva batería, se llaman pasivas y son las más baratas, aunque también las hay con batería y se las llama activas, pero ignoraremos estas.

Normalmente los **tags RFID** son una especie de pegatina que lleva diseños más o menos extravagantes de pistas metálicas muy finas que hacen las veces de antena y te has hartado a verlas en ropa y otros productos pero probablemente no sabías lo que era. Estas son las típicas insertas en las etiquetas de la ropa como antirrobo, pero las hay con aspectos muy variados, dependiendo de la forma y diseño e la antena y el destino final de las etiquetas.



Pero la idea es siempre la misma, identificar el tag mediante la emisión de ondas electromagnéticas desde un lector, que alimentan un circuito de memoria interno, para que nos envíen por RF una identificación que podamos usar para validar o identificar un producto cualquiera.



Tienen un montón de ventajas respecto de otros sistemas como los códigos de barras:

- Son pasivos en su mayoría y pueden permanecer inactivos durante años.
- No se rayan o degradan, ni con el tiempo ni con campos magnéticos, ni con el roce.
- No necesitan verse para realizar la identificación, porque la mayor parte de los materiales son transparentes a las ondas RF que se usan.
- Pueden disponer de una memoria interna que contenga toda la identificación del producto o su trazabilidad. Las hay de 1K, 4k y ...
- Se pueden utilizar como llaves maestras para abrir cerraduras electrónicas, tanto como para identificar productos o personas.
- Se pueden reescribir y leer tantas veces como quieras, e ir añadiendo su historia en la propia tarjeta. Por ejemplo en una tarjeta de pago, puedes conservar las últimas 100 compras realizadas, por ejemplo.

Lo que ha retrasado la implantación masiva de estos RFID ha sido más el precio que su ventaja. Imagínate una tarjeta de plástico con un tag interno. Lo puedes usar para llave de una cerradura, o probablemente la tarjeta de Bus o Metro que tienes en el bolsillo sea de estas. Puedes usar estas tarjetas para que su ID interno corresponda con tu clave de acceso al ordenador y por ello solo quien tenga tu tarjeta puede iniciar sesión en él.

O si cada producto que hay en el supermercado llevara un **tag RFID** anexo, podrías ir echando a la bolsa y salir tranquilamente por el arco de salida y este te leería sobre la marcha los productos que te llevas y la identificación de tu bolsillo o tarjeta de crédito, con lo que haría inventario inmediato y te cargaría la VISA sin más.

No serían necesarias cajas, porque la factura se haría leyendo los IDs de los productos que hayas comprado y el pago sería automático. Sin necesidad de parar en caja.

Y casualidad, **Arduino** está perfectamente preparado para este tipo de cosas ya ves, y en nuestro kit hay un set de un lector / grabador de **RFID**, además de una tarjeta de identificación de plástico y un llavero con tag, para que podamos hacer pruebas.

CONECTANDO EL LECTOR RFID

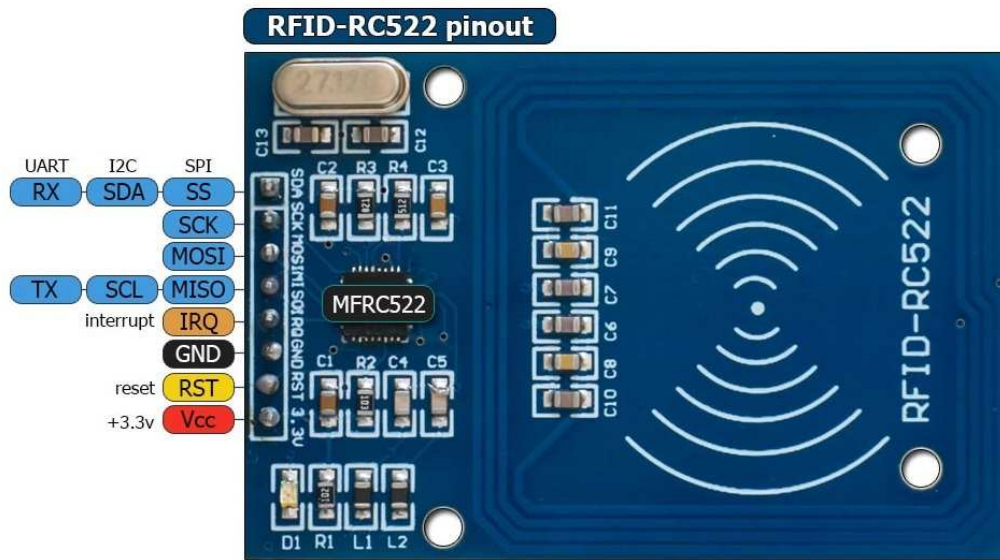
En nuestro kit de inicio Arduino, podremos encontrar un pequeño conjunto de, lector grabador de tags RFID, el RC522, así como una tarjeta y un llavero con identificadores que es con lo que vamos a hacer las pruebas.

El lector está basado en el típico chip **MFRC522** para lectura y escritura de tags **RFID**:



El chip **MFRC522**. Da soporte a la lectura y escritura de los tags en diferentes condiciones y con control de errores, de una forma sencilla, aunque **no puede identificar más que un tag a la vez**, a diferencia de otros lectores más profesionales, y requiere que la distancia de lectura sea de una par de centímetros a lo sumo.

La tarjeta presenta un **interface SPI** para comunicar nuestro **Arduino** con el **MFRC522** y no tiene más dificultad. De hecho creo que os sorprenderá ver lo fácil que resulta leer estas etiquetas **RFID**. Empecemos por el lector y sus pines:



Como podemos ver no hay más que los pines de control **SPI** y la alimentación del módulo. La conexión es trivial:

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

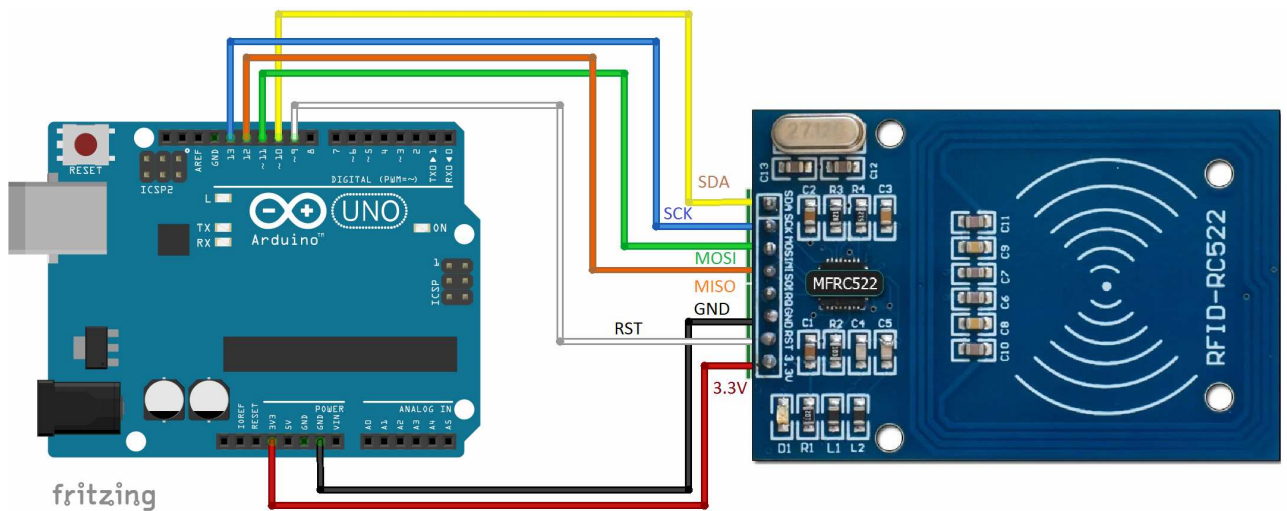


Figura A

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.

Programación del Módulo RC522: **Lectura del código de identificación**

Para poder trabajar el Módulo 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

La conexión circuital es como se ve en Figura A.

```
#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.

NOTA: Una variable indexada es una variable cuyo valor varía en función de otra variable.

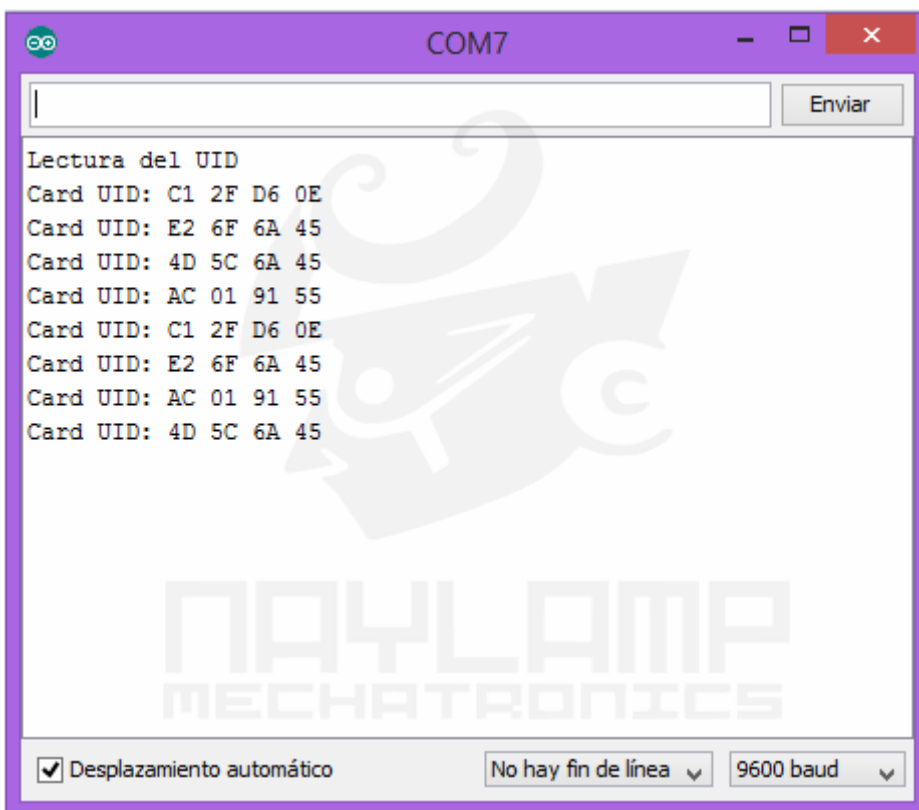
Finalizar la lectura.

```
mfrc522.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.



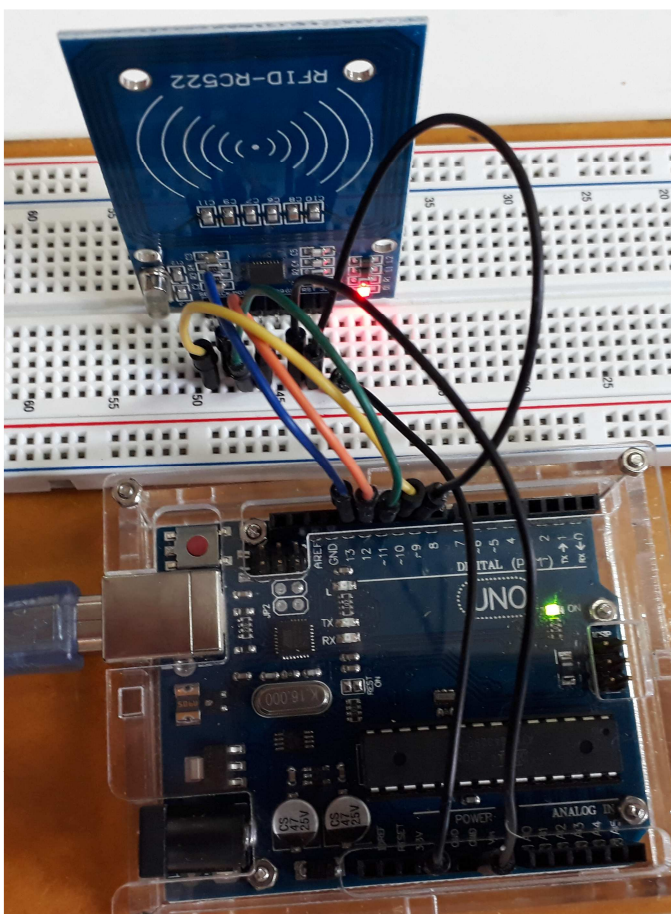
Para el caso de las tarjeta y llavero ensayado por el docente se registro:

Lectura del UID

Card UID: 94 43 D7 2B para la tarjeta

Card UID: C6 34 65 AC para el llavero

Card UID: CB C9 E6 A6 para tarjeta SUBE ensayada



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.

La conexión circuital es como se ve en Figura A.

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

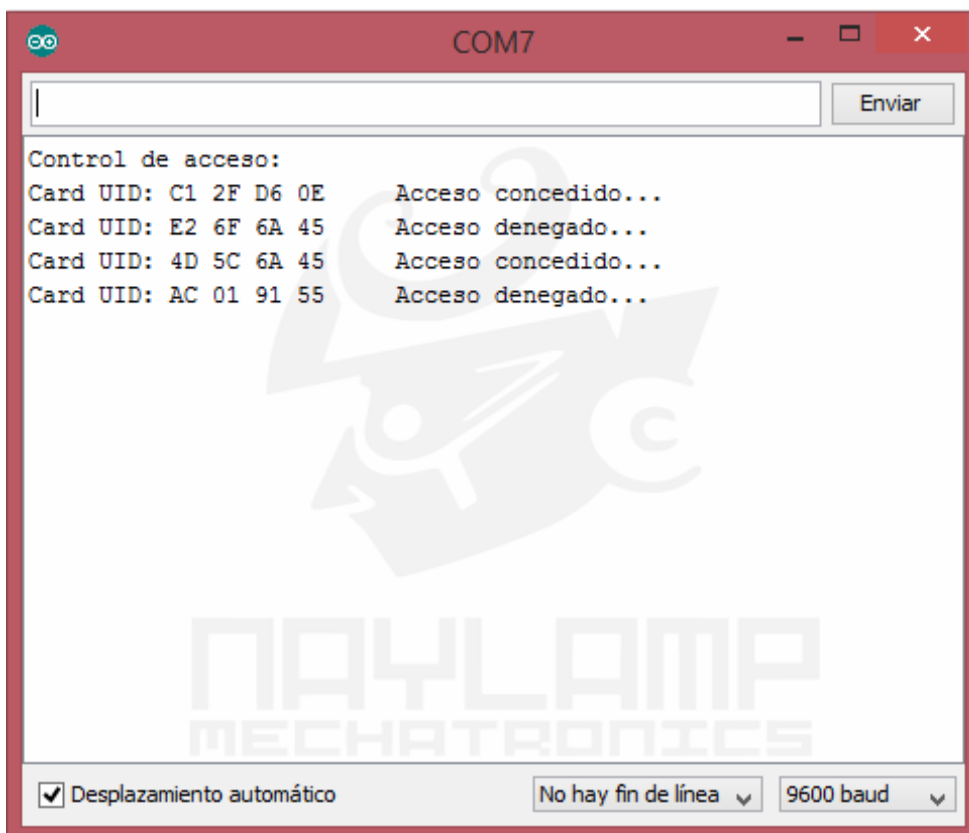
void setup() {
    Serial.begin(9600); //Iniciamos La comunicacion serial
    SPI.begin(); //Iniciamos el Bus SPI
    mfrc522.PCD_Init(); // Iniciamos el MFRC522
    Serial.println("Control de acceso:");
}

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 serialemente 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.



Variantes

En lugar de enviarlo por el Monitor Serie, la confirmación se puede programar para activar una salida digital, que puede estar conectado a un LED, un motor eléctrico, 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 luego).

Creo que hasta ahora no habíamos comentado, que estos llaveros y tarjetas sencillos que usamos para estas pruebas, viene con un numero de ID pregrabado, diferente para cada uno de ellos y quedaria la tarea de calcular la posibilidad de que dos fabricantes usen el mismo número.

Sería muy interesante poner un Buzzer, que emitiera un pitido cada vez que lee correctamente una tarjeta y de eso modo sabríamos que hay una identificación positiva.

RESUMEN DE LA SESIÓN

- Hemos presentado el concepto de los RFID y un poco como funcionan.
- Vimos el chip **MFRC522** y el modulo Modulo **RC522**.
- Vimos que ARduino no tiene problema con leer RFID.
- Presentamos un programa de ejemplo de como leer la identificacion de un tag RFID.