Lectura y escritura en la tarjeta o llavero. *Versión: 6-6-18*

Chip: NXP MIFARE 1 IC S50

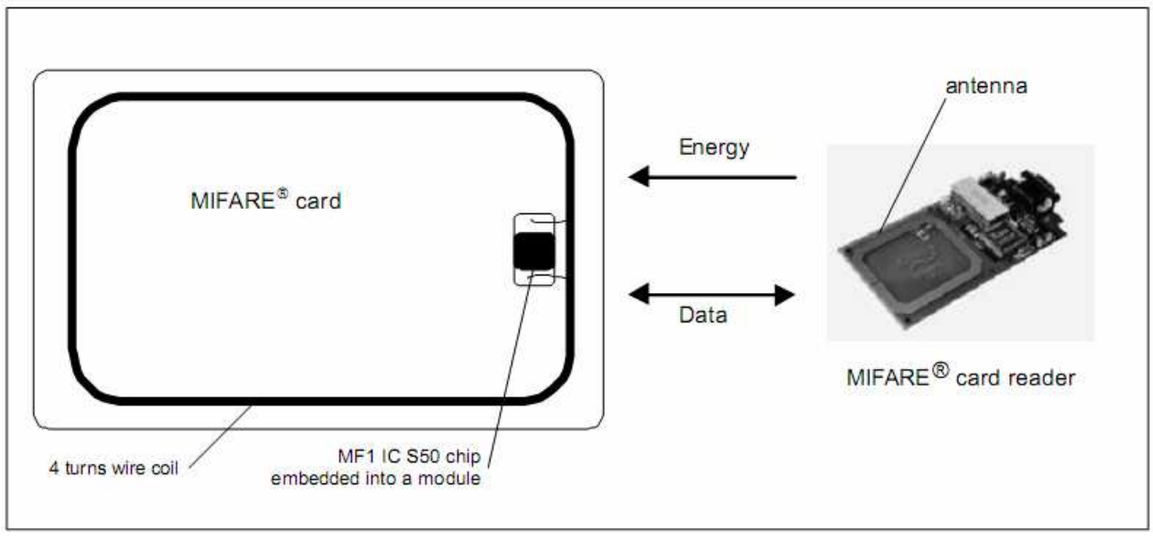
Capacidad de almacenamiento: 1K Bytes
Frecuencia de trabajo: 13.56 MHz
Tasa de Comunicación: 106Kbps
Distancia de funcionamiento: 0 ~ 100mm (dependiendo de la geometría de la antena).
Tiempo de funcionamiento: 1 ~ 5ms
Temperatura de funcionamiento: -20 ~ 55 ℃ (-68℉ ~ +131℉)
Transmisión de datos y suministro sin contacto energía (no se necesita batería)
Alta integridad de datos.
Transacción de ticketing típica: <100 ms (incluido gestión de respaldo.
EEPROM: >100, 000 times
Retención de datos: >10 years
Dimensión: ISO standard card 85.6x54x0.82
Materiales de embalaje: PVC, PET, PETG, 0.13mm copper wire
Tecnología de encapsulación: Ultrasonic automatic plant line / Automatic touch welding.
Estándar RF: ISO14443A
Disponible para el tamaño: 85.6×54, 83×20, 70×40, 50×50, 45×45, 45×28, 44×20, 38×38, 35×30

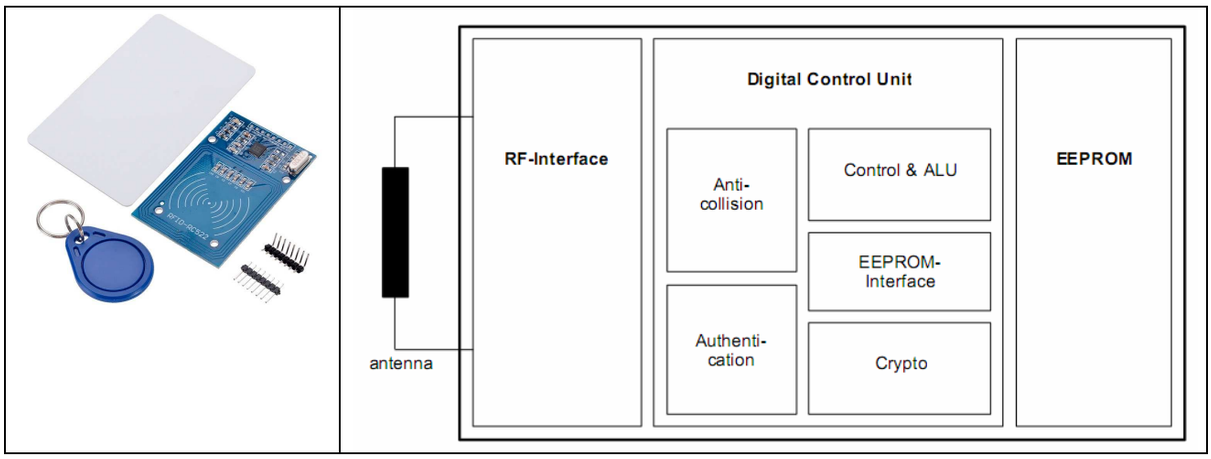
EEPROM

1 Kbyte, organizado en 16 sectores con 4 bloques de 16 bytes cada uno (un bloque consta de 16 bytes). Condiciones de acceso definidas por el usuario para cada bloque de memoria. (En cada bloque podría escribir 16 caracteres.)

SEGURIDAD

Autenticación mutua de tres pasos (ISO / IEC DIS9798-2).
Cifrado de datos en el canal de RF con repetición protección contra ataques. Entre otras características





Esquema básico de memoria

Sector	Bloque	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8	Byte9	Byte10	Byte11	Byte12	Byte13	Byte14	Byte15
0	3																
	2	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	1	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	0																
1	7	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	6	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	5	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	4	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
2	11																
	10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	9	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	8	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
3	15																
	14	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	13	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	12	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4	19																
	18	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	17	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	16	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
5	23																
	22	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	21	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
6	27																
	26	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	25	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	24	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
7	31																
	30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	29	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	28	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
8	35																
	34	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	33	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	32	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
9	39																
	38	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	37	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	36	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
10	43																
	42	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	41	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	40	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
11	47																
	46	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	45	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	44	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
12	51																
	50	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	49	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	48	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
13	55																
	54	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	53	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	52	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
14	59																
	58	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	57	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	56	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
15	63																
	62	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	61	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

NO todos los bloques están disponibles totalmente para el usuario.

El primer bloque de datos (bloque 0) del primer sector (sector 0) contiene los datos del fabricante del IC. Debido a los requisitos de seguridad y sistema, este bloque es de escritura protegida después de haber sido programado por el fabricante del IC durante la producción.

*Todos los demás sectores contienen **3 bloques** de **16 bytes** para almacenar datos, salvo como ya se menciono, el Sector 0 que contiene solo dos bloques de datos y un bloque del fabricante de solo lectura).*

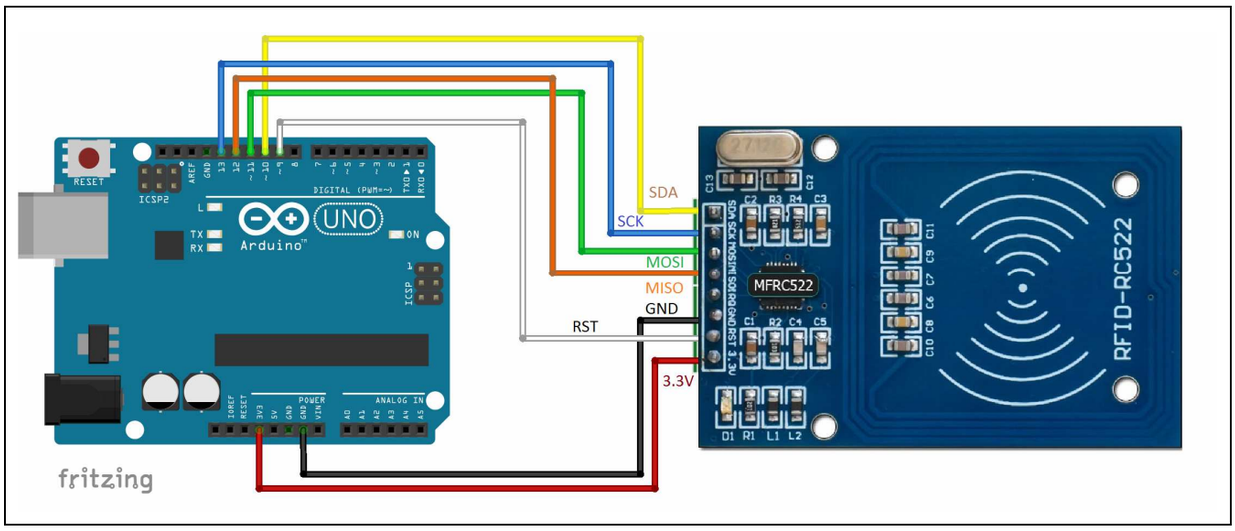
Cada sector tiene un trailer del sector que contiene claves secretas A y B (opcional),
Si la clave B no es necesaria, los últimos 6 bytes del bloque 3 pueden ser utilizados como bytes de datos. El byte 9 del trailer del sector está disponible para los datos del usuario.

Byte Number	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Description	Key A						Access Bits				Key B (optional)					

Sector	Block	Byte Number within a Block 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Description
15	3	Key A						Access Bits				Key B						Sector Trailer 15
	2																	Data
	1																	Data
	0																	Data
14	3	Key A						Access Bits				Key B						Sector Trailer 14
	2																	Data
	1																	Data
	0																	Data
:	:																	

Para un completo control sobre las prestaciones de la tarjeta, se debe acceder a la hoja de datos.

Para todos los ensayos usaremos la siguiente conexión del lector



Ejemplo: Programa que permite ver el contenido de la memoria

```
/*LeerTodaMemoria
 * Se leeran sectores y bloques
 *

*/


#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.

MFRC522::MIFARE_Key key;

/**
* Initialize.
*/
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 card

// using FFFFFFFFFFFFh which is the default at chip delivery from the factory
for (byte i = 0; i < 6; i++) {
key.keyByte[i] = 0xFF;
}

Serial.println(F("Ingrese Tarjeta"));
}

/**
* Main loop.
*/

//----------------------------------------

void(* resetFunc) (void) = 0; // esta es la funcion de reset
//----------------------------------------


void loop() {
// Look for new cards
if ( ! mfrc522.PICC_IsNewCardPresent())
return;

// Select one of the cards
if ( ! mfrc522.PICC_ReadCardSerial())
return;
//---Leo sector 0----------
byte sector = 0;

byte dataBlock[] = {
0x01, 0x02, 0x03, 0x04, // 1, 2, 3, 4,
0x05, 0x06, 0x07, 0x08, // 5, 6, 7, 8,
0x08, 0x09, 0xff, 0x0b, // 9, 10, 255, 12,
0x0c, 0x0d, 0x0e, 0x0f // 13, 14, 15, 16
};

MFRC522::StatusCode status;
byte buffer[18];
byte size = sizeof(buffer);

Serial.println(F("Datos actuales en el sector:"));
mfrc522.PICC_DumpMifareClassicSectorToSerial(&(mfrc522.uid), &key, sector);
Serial.println();

if (status != MFRC522::STATUS_OK) {
Serial.print(F("MIFARE_Read() failed: "));
Serial.println(mfrc522.GetStatusCodeName(status));
}

Serial.println();

//------Fin leer sector 0------------------

//---------------leo el sector1-----------
sector = 1;

buffer[18];
size = sizeof(buffer);

Serial.println(F("Datos actuales en el sector:"));
mfrc522.PICC_DumpMifareClassicSectorToSerial(&(mfrc522.uid), &key, sector);
Serial.println();

if (status != MFRC522::STATUS_OK) {
Serial.print(F("MIFARE_Read() failed: "));
Serial.println(mfrc522.GetStatusCodeName(status));
}

Serial.println();
```

```
//------------Fin lectura del sector 1--------------
/* Ahora leemos desde el sector 2
 *  Solo se hace asi para mostras las opciones
 *  que tenemos para leer sectores.
 */

//---------------leo el sector n-----------

for (int a=2; a<16; a++) // declara i y prueba si es menor
                         //que 16, incrementa i.
        {

sector = a; //carga numero de sector a mostrar

buffer[18];
size = sizeof(buffer);

Serial.println(F("Datos actuales en el sector:"));
mfrc522.PICC_DumpMifareClassicSectorToSerial(&(mfrc522.uid), &key, sector);
Serial.println();

if (status != MFRC522::STATUS_OK) {
Serial.print(F("MIFARE_Read() failed: "));
Serial.println(mfrc522.GetStatusCodeName(status));
}

Serial.println();
//------------Fin lectura del sector n--------------

      delay(50);

       } //del FOR


Serial.println(F("Espere para colocar otra tarjeta....."));

Serial.println(F("**********************************************"));
Serial.println(F("**********************************************"));

//Reinicio luego de un tiempo

delay(3000);

resetFunc();  // reseteo el Arduino

}
```

El resultado para una tarjeta ensayada:

(Se ha marcado en rojo en donde observamos que hay datos almacenados de una escritura realizada en un ejercicio anterior).

```
Ingrese Tarjeta

Datos actuales en el sector:

  0      3   00 00 00 00  00 00 FF 07  80 69 FF FF  FF FF FF FF  [ 0 0 1 ]

         2   20 20 20 20  20 20 20 87  42 07 04 FF  FF FF 32 42  [ 0 0 0 ]

         1   44 61 6E 69  65 6C 20 20  20 20 20 20  20 20 20 20  [ 0 0 0 ]

         0   94 43 D7 2B  2B 08 04 00  62 63 64 65  66 67 68 69  [ 0 0 0 ]

Datos actuales en el sector:
```

1 7 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [0 0 1]

6 0D 0A 50 41 42 4C 4F 20 20 20 20 20 20 20 20 [0 0 0]

5 50 45 44 52 4F 54 45 20 20 20 20 20 20 20 20 [0 0 0]

4 0D 0A 4A 75 61 6E 20 20 20 20 20 20 20 20 20 [0 0 0]

Datos actuales en el sector:

2 11 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [0 0 1]

10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [0 0 0]

9 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [0 0 0]

8 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [0 0 0]

Datos actuales en el sector:

3 15 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [0 0 1]

14 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [0 0 0]

13 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [0 0 0]

12 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [0 0 0]

Datos actuales en el sector:

4 19 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [0 0 1]

18 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [0 0 0]

17 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [0 0 0]

16 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [0 0 0]

Datos actuales en el sector:

5 23 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [0 0 1]

22 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [0 0 0]

21 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [0 0 0]

20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [0 0 0]

Datos actuales en el sector:

6 27 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [0 0 1]

26 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [0 0 0]

25 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [0 0 0]

24 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [0 0 0]

Datos actuales en el sector:

7 31 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [0 0 1]

30 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [0 0 0]

29 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [0 0 0]

28 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [0 0 0]

Datos actuales en el sector:

8 35 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [0 0 1]

34 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [0 0 0]

33 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [0 0 0]

32 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [0 0 0]

Datos actuales en el sector:

9 39 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [0 0 1]

38 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [0 0 0]

37 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [0 0 0]

36 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [0 0 0]

Datos actuales en el sector:

10 43 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [0 0 1]

42 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [0 0 0]

41 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [0 0 0]

40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [0 0 0]

Datos actuales en el sector:

11 47 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [0 0 1]

46 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [0 0 0]

45 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [0 0 0]

44 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [0 0 0]

Datos actuales en el sector:

12 51 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [0 0 1]

50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [0 0 0]

49 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [0 0 0]

48 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [0 0 0]

Datos actuales en el sector:

13 55 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [0 0 1]

54 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [0 0 0]

53 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [0 0 0]

52 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [0 0 0]

Datos actuales en el sector:

14 59 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [0 0 1]

58 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [0 0 0]

57 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [0 0 0]

```
      56   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  [ 0 0 0 ]

Datos actuales en el sector:

 15    63   00 00 00 00  00 00 FF 07  80 69 FF FF  FF FF FF FF  [ 0 0 1 ]

      62   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  [ 0 0 0 ]

      61   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  [ 0 0 0 ]

      60   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  [ 0 0 0 ]

Espere para colocar otra tarjeta.....

**************************************

**************************************

Ingrese Tarjeta
```

Si por alguna razón se sobrescribe los bloques de los trailer, entonces quedaran inutilizados los sectores correspondientes y el programa anterior no dará los resultados esperados. Generalmente deja de mostrar el contenido luego del primer sector inutilizado.

Recopilado de Internet.

Con MIFARE Classic 1K, cada 4º bloque es el trailer del sector (cada 4 bloques se agrupan en un sector). El trailer del sector contiene las claves de acceso (clave A en los bytes 0..5, clave B en los bytes 10..15) y las condiciones de acceso (bits de acceso en los bytes 6..8) para un sector.
Las condiciones de acceso están protegidas por un mecanismo de redundancia en el que cada bit de acceso está presente varias veces en lógica positiva y negativa. Una tarjeta MIFARE Classic permite sobrescribir estas condiciones de acceso con valores no válidos (combinaciones imposibles de bits de acceso). Sin embargo, una vez que las condiciones de acceso se establecen en un valor tan inválido, la lógica de seguridad del chip desactivará todo acceso al sector wole. En consecuencia, la escritura de condiciones de acceso no válidas para el sector trailer deja inutilizable todo el sector . Este estado es permanente y no puede revertirse.

Si no se cometió el mismo error en otros sectores, estos podran seguir usandose, descartando los inutilizados.

Ejemplo: Programa que permite Escribir en un Bloque especifico

El siguiente programa esta basado en ejemplos de Internet, pero fue adaptado por el docente para que el usuario tenga la posibilidad de elegir el bloque que desea usar para guardar un dato (16 caracteres máximo), y con el control necesario para que no se escriba en los bloques de los trailer. La cosmética del programa puede ser mejorada para que sea más intuitiva para el usuario.

```
//ProgramaEscribrTarjLlaveUnBloque3B (Se depuro lineas respecto al A)
//Luego lee el bloque escrito
#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
//Variables globales----------------------------

int num=999;// Variable para ingresar numero de bloque a escribir

//-----------Fin variables globales-------------
MFRC522 mfrc522(SS_PIN, RST_PIN); // Create MFRC522 instance


//-----------------------------------------

void(* resetFunc) (void) = 0; // esta es la funcion de reset
//-----------------------------------------

void setup() {
delay(3000);
Serial.begin(9600); // Initialize serial communications with the PC
```

```
//**********Ingresar numero de bloque*****************************

//Se va a pedir ingresar el numero de bloque para escribir----
  Serial.println("Coloque tarjeta en lector para escribir...(3seg)");
  delay(3000);
 Serial.println("***********************************************");

  Serial.println("Ingrese bloque a escribir...(5seg)");

delay(5000);//Da un tiempo para que usuario ingrese numero de bloque

 /*
   * Evaluamos el momento en el cual recibimos un caracter
   * a travÃ©s del puerto serie
   */
  if (Serial.available()>0) {

    //A continuacion acciones realizadas solo desde PC--

  //Delay para favorecer la lectura de caracteres
    delay(200);
    //Se crea una variable que servirÃ¡ como buffer
    String bufferString = "";
    /*
     * Se le indica a Arduino que mientras haya datos
     * disponibles para ser leÃ-dos en el puerto serie
     * se mantenga concatenando los caracteres en la
     * variable bufferString
     */
    while (Serial.available()>0) {
      bufferString += (char)Serial.read();
    }

    num = bufferString.toInt();

      //Analiza si no se ingreso nada----

     if ((num==999)||(num==3)|| (num==11)|| (num==15)|| (num==19)|| (num==23)|| (num==27)|| (num==31)||
(num==35)|| (num==39)|| (num==43)|| (num==47)|| (num==51)|| (num==55)|| (num==59)|| (num==63))
            {

             Serial.println("No se eligio bloque valido.... Espere..");

             Serial.println("++++++++++++++++++++++++++++++++");
             delay(2000);
             Serial.println();


             resetFunc();  // llamo funcion especifica de reseteo

            }

    Serial.print("Numero de bloque a escribir -->");

    Serial.print(num);

  Serial.println("  ------------------Aceptado");

  Serial.println("Preparece para escribir...");
  delay(3000); //Tiempo para que el usuario lea numero de bloque ingresado
  //Serial.print(num);

}

 Serial.println("****----------------------------****");
  delay(3000);


//**********FIN  Ingresar numero de bloque***************
SPI.begin(); // Init SPI bus
mfrc522.PCD_Init(); // Init MFRC522 card
Serial.println(F("Coloque (si no lo hizo) tarjeta para Escribir dato en bloque"));
delay(3000); //tiempo para colocar tarjeta si ya lo se hizo

}

void loop() {

// Prepare key - all keys are set to FFFFFFFFFFFFh at chip delivery from the factory.
MFRC522::MIFARE_Key key;
for (byte i = 0; i < 6; i++) key.keyByte[i] = 0xFF;

// Look for new cards
if ( ! mfrc522.PICC_IsNewCardPresent()) {
return;
}

// Select one of the cards
```

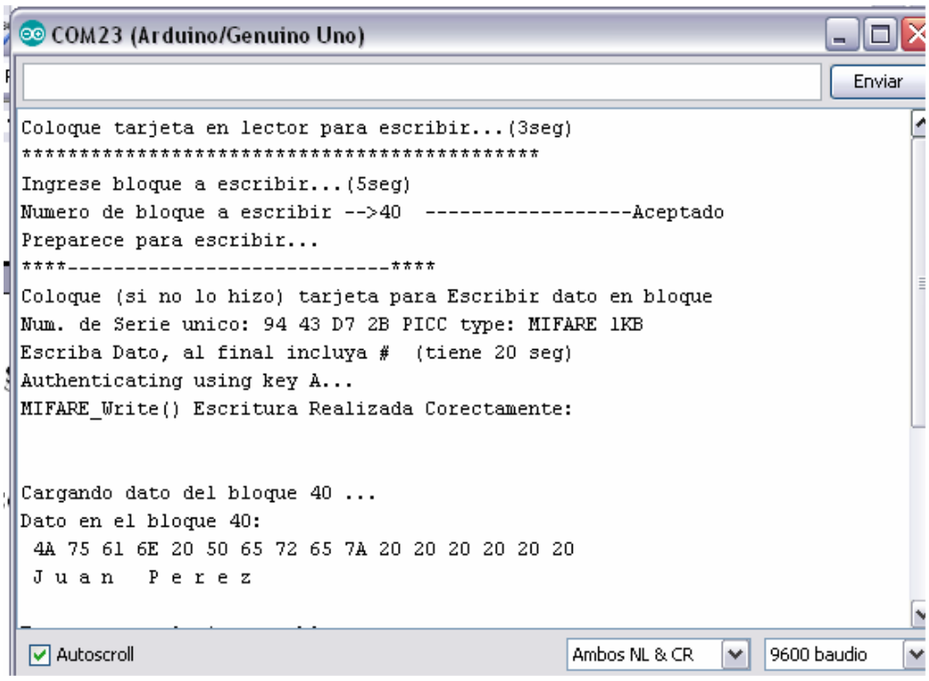
```
if ( ! mfrc522.PICC_ReadCardSerial()) return;

Serial.print(F("Num. de Serie unico:")); //Dump 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.print(F(" PICC type: ")); // Dump PICC type
MFRC522::PICC_Type piccType = mfrc522.PICC_GetType(mfrc522.uid.sak);
Serial.println(mfrc522.PICC_GetTypeName(piccType));

byte buffer[34];
byte block;
MFRC522::StatusCode status;
byte len;

Serial.setTimeout(20000L) ; // wait until 20 seconds for input from serial

// Ask personal data: First name
Serial.println(F("Escriba Dato, al final incluya #  (tiene 20 seg)"));
len=Serial.readBytesUntil('#', (char *) buffer, 20) ; // read first name from serial
for (byte i = len; i < 20; i++) buffer[i] = ' '; // pad with spaces

block = num; //Se especifica en que bloque se escribira _____------_______-----_______

Serial.println(F("Authenticating using key A..."));
status = mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A, block, &key, &(mfrc522.uid));
if (status != MFRC522::STATUS_OK) {
Serial.print(F("PCD_Authenticate() failed: "));
Serial.println(mfrc522.GetStatusCodeName(status));

Serial.println("Re Iniciando...");
Serial.println("++++++++++++++++++++++++++++++++++++++");

delay(2000);
resetFunc();  // reseteo el Arduino

return;
}

// Write block
status = mfrc522.MIFARE_Write(block, buffer, 16);
if (status != MFRC522::STATUS_OK) {
Serial.print(F("MIFARE_Write() failed: "));
Serial.println(mfrc522.GetStatusCodeName(status));

Serial.println("Re Iniciando...");
Serial.println("++++++++++++++++++++++++++++++++++++++");

delay(2000);
resetFunc();  // reseteo el Arduino

return;
}
else Serial.println(F("MIFARE_Write() Escritura Realizada Corectamente: "));
Serial.println(" ");
//Se leera el bloque escrito

//---------------leo el bloque num-----------

byte blockAddr = num; //Se especifica que bloque se leera _____------_______-----_______

buffer[18];
byte size = sizeof(buffer);

Serial.println();

// Read data from the block
Serial.print(F("Cargando dato del bloque ")); Serial.print(blockAddr);
Serial.println(F(" ..."));
status = (MFRC522::StatusCode) mfrc522.MIFARE_Read(blockAddr, buffer, &size);
if (status != MFRC522::STATUS_OK) {
Serial.print(F("MIFARE_Read() failed: "));
Serial.println(mfrc522.GetStatusCodeName(status));
Serial.println("Re Iniciando...");
delay(2000);
resetFunc();  // reseteo el Arduino

}
Serial.print(F("Dato en el bloque ")); Serial.print(blockAddr); Serial.println(F(":"));
dump_byte_array(buffer, 16); Serial.println();
Serial.println();
//------------Fin lectura del bloque num----------

mfrc522.PICC_HaltA(); // Halt PICC
mfrc522.PCD_StopCrypto1(); // Stop encryption on PCD

Serial.println(F("Espere para elegir otro bloque....."));
```

```
Serial.println(F("*****************************************"));
Serial.println(F("*****************************************"));

//Reinicio luego de un tiempo
delay(3000);
resetFunc();  // reseteo el Arduino
}

void dump_byte_array(byte *buffer, byte bufferSize) {
    for (byte i = 0; i < bufferSize; i++)
        {
        Serial.print(buffer[i] < 0x10 ? " 0" : " ");
        Serial.print(buffer[i], HEX);//Muestra en HEX

        }
        Serial.println();//Renglon en blanco

    for (byte i = 0; i < bufferSize; i++)
        {
        Serial.print(buffer[i] < 0x10 ? " 0" : " ");

        Serial.write(buffer[i]);//Muestra contenido para leer
        }

}
```

El programa va guiando en los pasos de escritura, de no poderse llevar a cabo o terminada la misma, se reinicia.

Ejemplo para escribir el bloque 40 con el nombre Juan Perez:

Usamos el Monitor Serie para interactuar con el Arduino Uno.



Ejemplo: Programa que permite LEER en un Bloque especifico

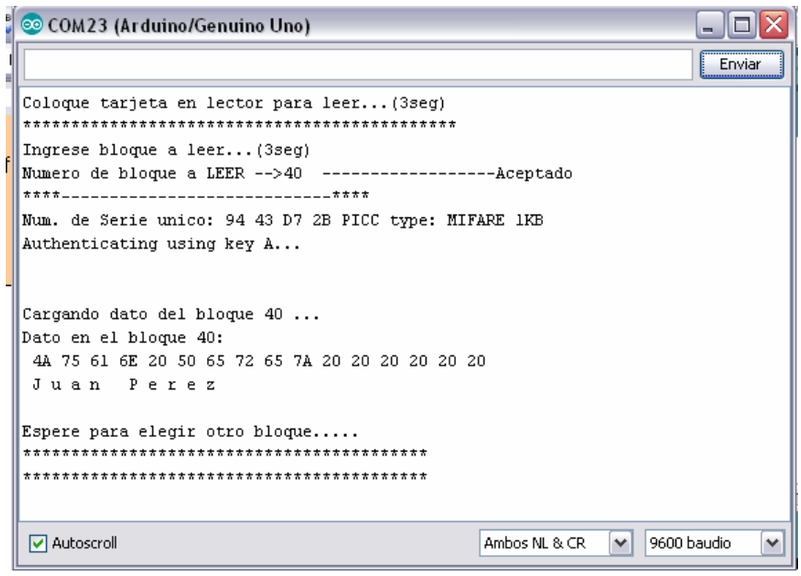
También basado en ejemplos de Internet, pero adaptado por el docente.

```
//Programa LeerBloque2 (Depurado algunas lineas respecto a version 1)
//Leer  bloque elegido
#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
//Variables globales----------------------------

int num=999;// Variable para ingresar numero de bloque a leer

//-----------Fin variables globales-------------
MFRC522 mfrc522(SS_PIN, RST_PIN); // Create MFRC522 instance
//----------------------------------------
void(* resetFunc) (void) = 0; // esta es la funcion de reset
//----------------------------------------
void setup() {
```

```
delay(3000);
Serial.begin(9600); // Initialize serial communications with the PC

//**********Ingresar numero de bloque*****************************

//Se va a pedir ingresar el numero de bloque para escribir----
  Serial.println("Coloque tarjeta en lector para leer...(3seg)");

 Serial.println("*********************************************");

  Serial.println("Ingrese bloque a leer...(3seg)");

delay(3000);//Da un tiempo para que usuario ingrese numero de bloque

 /*
   * Evaluamos el momento en el cual recibimos un caracter
   * a través del puerto serie
   */
 if (Serial.available()>0) {

    //A continuacion acciones realizadas solo desde PC--

  //Delay para favorecer la lectura de caracteres
    delay(200);
    //Se crea una variable que servirá como buffer
    String bufferString = "";
    /*
     * Se le indica a Arduino que mientras haya datos
     * disponibles para ser leídos en el puerto serie
     * se mantenga concatenando los caracteres en la
     * variable bufferString
     */
    while (Serial.available()>0) {
      bufferString += (char)Serial.read();
    }

    num = bufferString.toInt();
      //Analiza si no se ingreso nada----

     if (num==999)
           {

            Serial.println("No se eligio bloque valido.... Espere..");

            Serial.println("++++++++++++++++++++++++++++++++++++++++++++++++");
            delay(2000);
            Serial.println();

            resetFunc();  // llamo funcion especifica de reseteo

           }

    Serial.print("Numero de bloque a LEER -->");

    Serial.print(num);

    Serial.println("  ------------------Aceptado");

 }

 Serial.println("****----------------------------****");
  delay(3000);


//**********FIN  Ingresar numero de bloque***************
SPI.begin(); // Init SPI bus
mfrc522.PCD_Init(); // Init MFRC522 card

}

void loop() {

// Prepare key - all keys are set to FFFFFFFFFFFFh at chip delivery from the factory.
MFRC522::MIFARE_Key key;
for (byte i = 0; i < 6; i++) key.keyByte[i] = 0xFF;

// Look for new cards
if ( ! mfrc522.PICC_IsNewCardPresent()) {
return;
}

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

Serial.print(F("Num. de Serie unico:")); //Dump 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.print(F(" PICC type: ")); // Dump PICC type
MFRC522::PICC_Type piccType = mfrc522.PICC_GetType(mfrc522.uid.sak);
Serial.println(mfrc522.PICC_GetTypeName(piccType));

byte buffer[34];
byte block;
MFRC522::StatusCode status;
byte len;

block = num; //Se especifica en que bloque se leera _____------_______-----_______

Serial.println(F("Authenticating using key A..."));
status = mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A, block, &key, &(mfrc522.uid));
if (status != MFRC522::STATUS_OK) {
Serial.print(F("PCD_Authenticate() failed: "));
Serial.println(mfrc522.GetStatusCodeName(status));

Serial.println("Re Iniciando...");
Serial.println("++++++++++++++++++++++++++++++++++++++++++++++");

delay(2000);
resetFunc();  // reseteo el Arduino

return;
}

Serial.println(" ");
//Se leera el bloque elegido

//---------------leo el bloque num-----------

byte blockAddr = num; //Se especifica que bloque se leera _____------_______-----_______
buffer[18];
byte size = sizeof(buffer);

Serial.println();

// Read data from the block
Serial.print(F("Cargando dato del bloque ")); Serial.print(blockAddr);
Serial.println(F(" ..."));
status = (MFRC522::StatusCode) mfrc522.MIFARE_Read(blockAddr, buffer, &size);
if (status != MFRC522::STATUS_OK) {
Serial.print(F("MIFARE_Read() failed: "));
Serial.println(mfrc522.GetStatusCodeName(status));
Serial.println("Re Iniciando...");
delay(2000);
resetFunc();  // reseteo el Arduino
}
Serial.print(F("Dato en el bloque ")); Serial.print(blockAddr); Serial.println(F(":"));
dump_byte_array(buffer, 16); Serial.println();
Serial.println();
//------------Fin lectura del bloque num--------------
mfrc522.PICC_HaltA(); // Halt PICC
mfrc522.PCD_StopCrypto1(); // Stop encryption on PCD

Serial.println(F("Espere para elegir otro bloque....."));

Serial.println(F("**********************************************"));
Serial.println(F("**********************************************"));

//Reinicio luego de un tiempo

delay(3000);

resetFunc();  // reseteo el Arduino

}

void dump_byte_array(byte *buffer, byte bufferSize) {
    for (byte i = 0; i < bufferSize; i++)
        {
        Serial.print(buffer[i] < 0x10 ? " 0" : " ");
        Serial.print(buffer[i], HEX);//Muestra en HEX

        }
        Serial.println();//Renglon en blanco

    for (byte i = 0; i < bufferSize; i++)
        {
        Serial.print(buffer[i] < 0x10 ? " 0" : " ");

        Serial.write(buffer[i]);//Muestra contenido para leer
        }

}
```

Vamos a leer el bloque 40 que escribimos en el programa anterior.



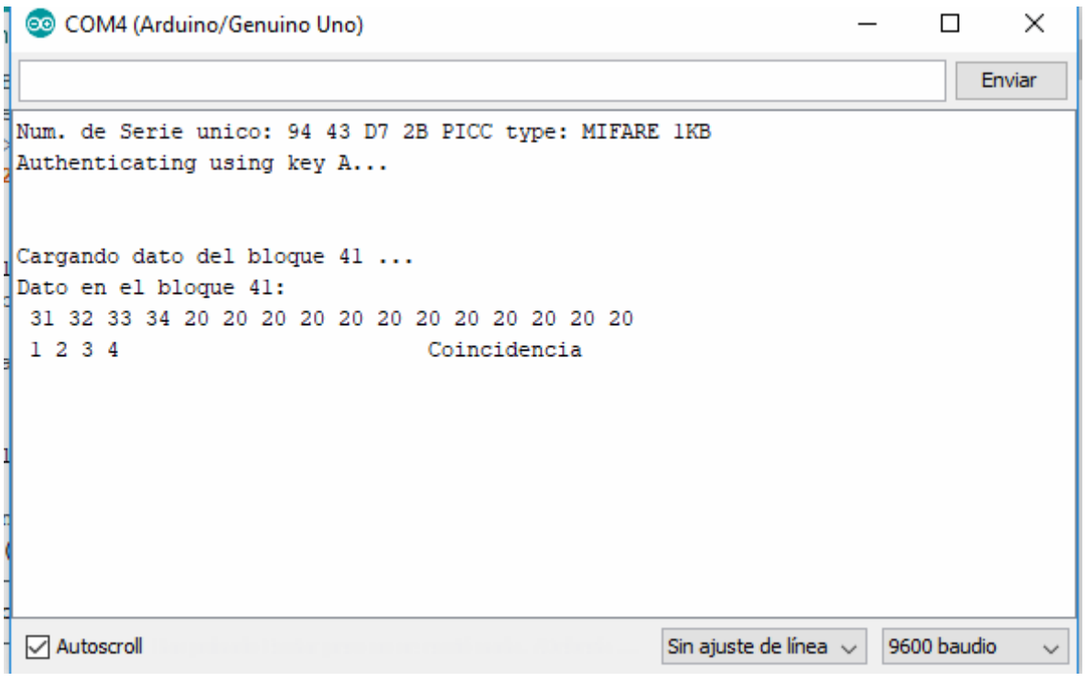
Ejemplo: Programa que permite Accion luego de LEER en un Bloque especifico

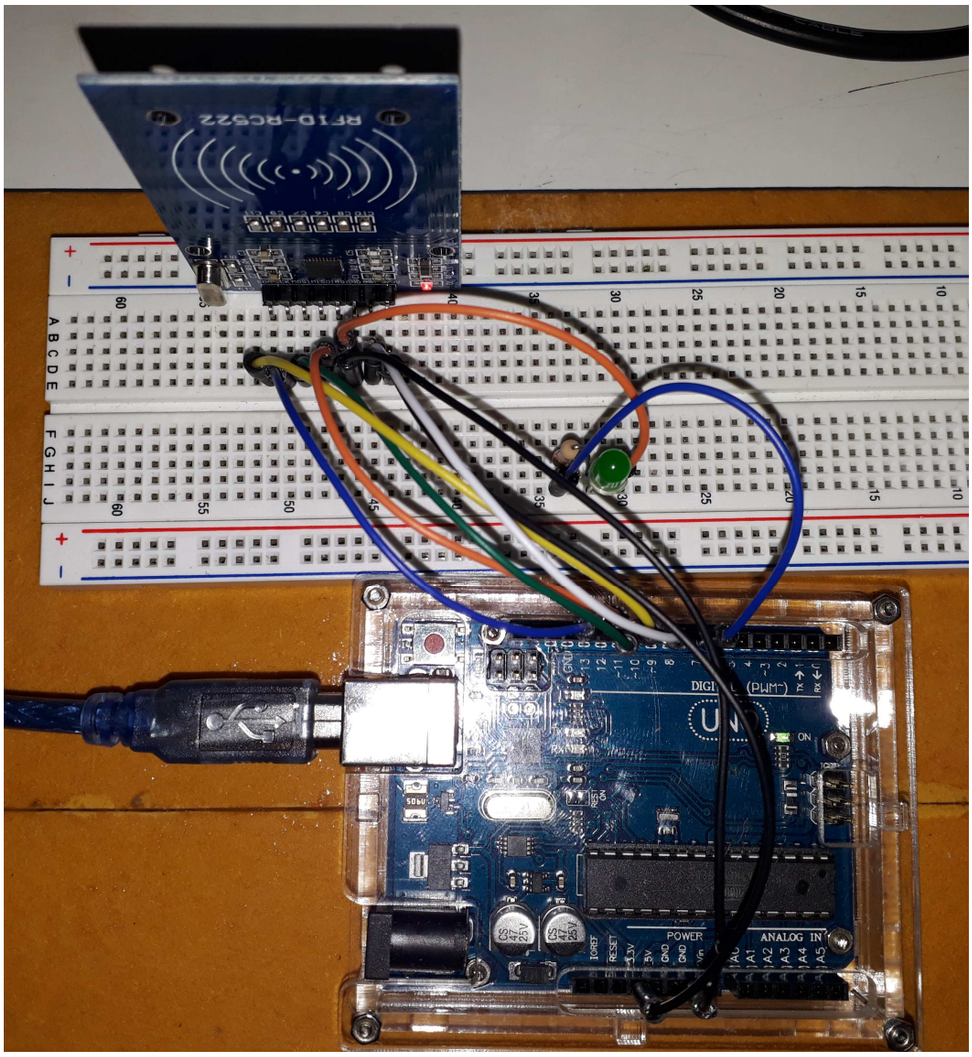
Se diseñara un programa que leyendo un bloque especifico (EJ: 41), y si este contiene una clave especifica (EJ: 1234), se ejecute una acción especifica (EJ: encendido de un LED VERDE en PIN 7 durante un tiempo, además de un mensaje “Coincidencia” en el monitor serie.

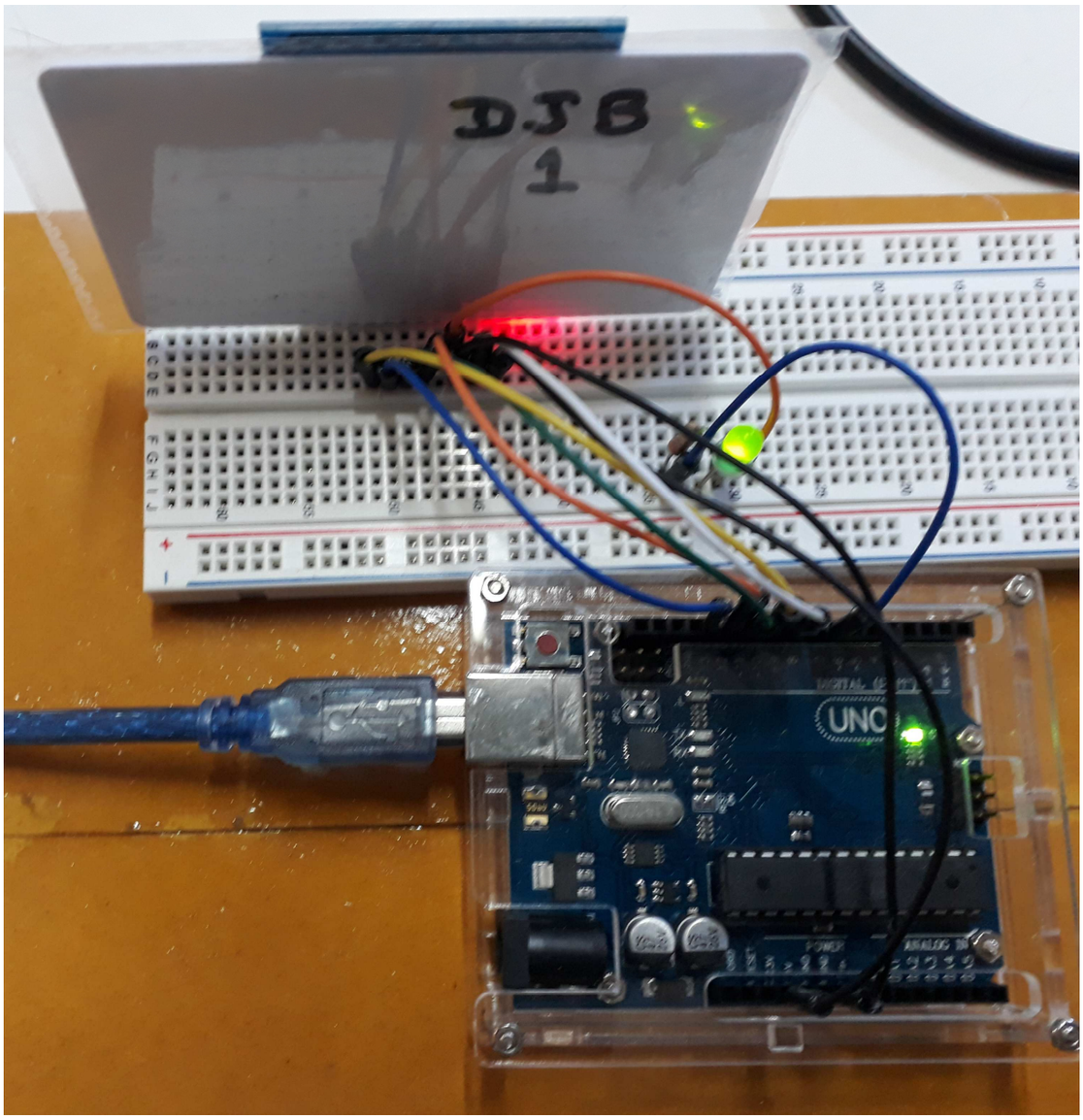
```
//Programa LeerBloqueAccion2 (Depurado de lineas del anterior)
//Leer  bloque elegido
#include <SPI.h>
#include <MFRC522.h>

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

int num=41;// Variable para indicar numero de bloque a leer

byte permitido[10] ={1,2,3,4};//Aqui se define contraseña

//-----------Fin variables globales-------------
MFRC522 mfrc522(SS_PIN, RST_PIN); // Create MFRC522 instance
//----------------------------------------
void(* resetFunc) (void) = 0; // esta es la funcion de reset
//----------------------------------------
void setup() {

pinMode(7, OUTPUT); // configura ‘pin 7’ como salida

delay(3000);
Serial.begin(9600); // Initialize serial communications with the PC

SPI.begin(); // Init SPI bus
mfrc522.PCD_Init(); // Init MFRC522 card

}

void loop() {

digitalWrite(7,LOW );//Apaga LED testigo

// Prepare key - all keys are set to FFFFFFFFFFFFh at chip delivery from the factory.
MFRC522::MIFARE_Key key;
for (byte i = 0; i < 6; i++) key.keyByte[i] = 0xFF;

// Look for new cards
if ( ! mfrc522.PICC_IsNewCardPresent()) {
return;
}

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

```
Serial.print(F("Num. de Serie unico:")); //Dump 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.print(F(" PICC type: ")); // Dump PICC type
MFRC522::PICC_Type piccType = mfrc522.PICC_GetType(mfrc522.uid.sak);
Serial.println(mfrc522.PICC_GetTypeName(piccType));

byte buffer[34];
byte block;
MFRC522::StatusCode status;
byte len;

block = num; //Se especifica se leera _____------_______-----_______

Serial.println(F("Authenticating using key A..."));
status = mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A, block, &key,
&(mfrc522.uid));
if (status != MFRC522::STATUS_OK) {
Serial.print(F("PCD_Authenticate() failed: "));
Serial.println(mfrc522.GetStatusCodeName(status));

Serial.println("Re Iniciando...");
Serial.println("++++++++++++++++++++++++++++++++++++++++++++++");

delay(2000);
resetFunc();  // reseteo el Arduino

return;
}

Serial.println(" ");
//Se leera el bloque elegido

//---------------leo el bloque num-----------

byte blockAddr = num; //Se especifica que bloque se leera _____------_______-----_______
buffer[18];
byte size = sizeof(buffer);

Serial.println();

// Read data from the block
Serial.print(F("Cargando dato del bloque ")); Serial.print(blockAddr);
Serial.println(F(" ..."));
status = (MFRC522::StatusCode) mfrc522.MIFARE_Read(blockAddr, buffer, &size);
if (status != MFRC522::STATUS_OK) {
Serial.print(F("MIFARE_Read() failed: "));
Serial.println(mfrc522.GetStatusCodeName(status));
Serial.println("Re Iniciando...");
delay(2000);
resetFunc();  // reseteo el Arduino
}
Serial.print(F("Dato en el bloque "));

Serial.print(blockAddr);
Serial.println(F(":"));
dump_byte_array(buffer, 16);

Serial.println();
Serial.println();
//------------Fin lectura del bloque num--------------
mfrc522.PICC_HaltA(); // Halt PICC
mfrc522.PCD_StopCrypto1(); // Stop encryption on PCD

//Reinicio luego de un tiempo

delay(2000);

resetFunc();  // reseteo el Arduino

}

void dump_byte_array(byte *buffer, byte bufferSize) {
    for (byte i = 0; i < bufferSize; i++)
        {
        Serial.print(buffer[i] < 0x10 ? " 0" : " ");
```

```
      Serial.print(buffer[i], HEX);//Muestra en HEX

       }
      Serial.println();//Renglon en blanco

    for (byte i = 0; i < bufferSize; i++)
       {
      Serial.print(buffer[i] < 0x10 ? " 0" : " ");

      Serial.write(buffer[i]);//Muestra contenido para leer

       }

//Cmparamos valor leido con contraseña----------------------

if (((buffer[0] -48)== permitido[0]) && ((buffer[1]-48)== permitido[1])&& ((buffer[2]-48)== permitido[2])&&
((buffer[3]-48)== permitido[3]))
     {

       Serial.println("Coincidencia");
       digitalWrite(7,HIGH );//Enciende LED testigo
       delay(5000);
       digitalWrite(7,LOW );//Apaga LED testigo

       }
else
    {

       Serial.println("No hubo Coincidencia");

      }
//Termina de comparar----------------------------------------

}
```




RFID-RC522
POWER
ANALOG IN


DJB
1
UNO