RFID

Fco. Javier Rodríguez Navarro

Cuadernos técnicos
www.pinguytaz.net

Indice

1. HISTÓRICO

Versión	Fecha	Autor	Observaciones
1.0	Mayo 2017	FJRN	Creación
2.0	Agosto 2017	FJRN	Añadimos Tarjeta Ultralight

2. INTRODUCCIÓN

Este cuaderno cubre el dispositivo RFID que nos permitirá recoger información de tarjetas RFID y NFC (Subconjunto de RFID con menos alcance)y etiquetas.

2.1. ¿Que es RFID?

RFID (Radio Frequency Identification), es un sistema de identificación por radio frecuencia de etiquetas o tarjetas en las que se puede almacenar información.

2.2. Etiquetas RFID (PICC)

También conocido como transpondedor, son unos dispositivos (Tarjetas, pegatinas pequeñas, etc.) que contienen una antena para permitirles recibir y responder a peticiones por radiofrecuencia desde un emisor-receptor RFID. Las etiquetas pasivas no necesitan alimentación eléctrica interna, mientras que las activas sí lo requieren.

Poseen una memoria interna con una capacidad que depende del modelo y son de varios tipos:

- **Solo lectura**: el código de identificación que contiene es único y es personalizado durante la fabricación de la etiqueta.
- **Lectura y escritura:** la información de identificación puede ser modificada por el lector. Estas son las utilizadas para tarjetas monedero que ademas permiten enviarlas comando para el control y la encriptación de eta información.

a) Etiquetas Pasivas

Las etiquetas pasivas no poseen alimentación eléctrica. La señal que les llega de los lectores induce una corriente eléctrica pequeña y suficiente para operar el circuito integrado CMOS de la etiqueta. Suelen tener distancias de uso de hasta unos 10 cm (ISO 14443 13,56 Mhz)y unos pocos metros dependiendo de la frecuencia y la antena.

- **ISO 14443-1**:Características físicas
- **ISO 14443-2**:Potencia RF y el interface de señal.
- **ISO 14443-3**:Funciones de inicialización y anticolisión.

- **ISO 14443-4**: Protocolo de transmisión.

Tenemos tipo A que son las Mifare, las que leera nuestro lector RC522, y tipo B las Calypso.

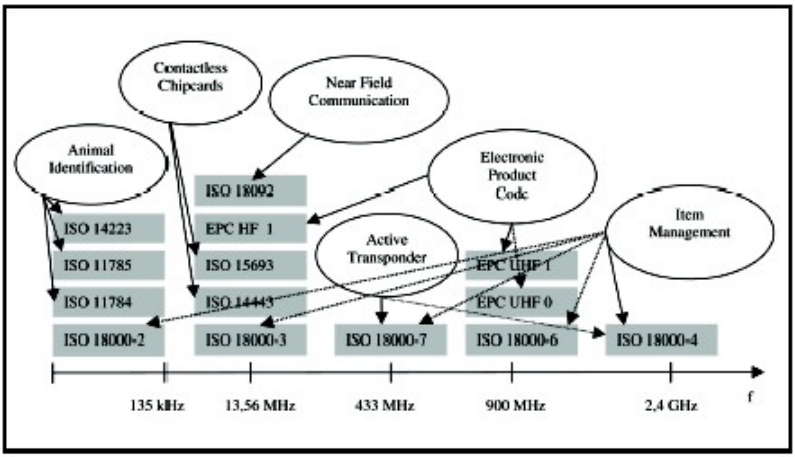


Figure 3: RFID technology standards and frequency bands

Al no precisan de alimentación pueden ser muy pequeñas, tanto como una pegatina o incluso un minúsculo dispositivo que se inserta en la piel.

b) Etiqueta activa

Las activas poseen su propia fuente autónoma de energía, que utilizan para dar corriente a sus circuitos

integrados y propagar su señal al lector. Estas son mucho más fiables debido a su capacidad de establecer sesiones con el lector y son capaces de transmitir señales más potentes que son efectivas a distancias mayores, hablamos de metros, pero como era de esperar son más grandes.

2.3. Lectores (PCD)

Llamado Transceptor son los dispositivos que nos permiten leer las etiquetas o configurarlas, sin tener visión directa como pasa en los códigos de barra, y ademas nos permite actualizar la información según el tipo de etiqueta.

Dependiendo del lector y los tipos de etiquetas, que tendrán que ser *anticolisión*, se permite también leer varias etiquetas a la vez sin tener que pasar una a una por el lector.

El lector envía periódicamente señales para ver si hay alguna etiqueta en sus inmediaciones. Cuando capta una señal de una etiqueta extrae la información y se pasa a procesarla.

3. MODULO RC522

Es el modulo de lectura / escritura basado en el CHIP MFRC522 de NXP.

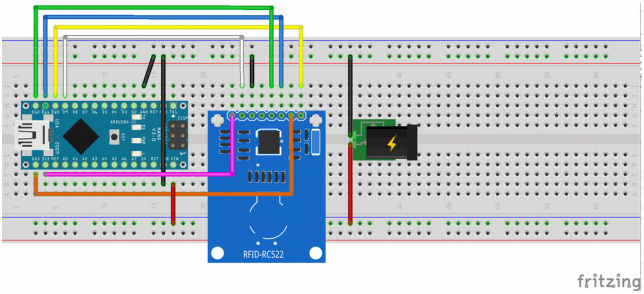
Características:

- Alimentación de 3,3V
- Maxima corriente 30 mA
- Frecuencia 13,56 Mhz.
- Distancia lectura hasta 60 mm.
- Protocolo SPI. Otros I2C y UART.
- Soporta tarjetas soporta las tarjetas Mifare1 S50, Mifare1 S70, Mifare UltraLight, Mifare Pro y Mifare Desfire.

Utilizaremos y describiremos el lector RFID-RC522 que se comunicara a un arduino mediante el protocolo *SPI*, y para ayudarnos en la programación usaremos la librería *MFRC522* de Miguel Balboa.

3.1. Esquema de conexión

Las conexiones al arduino se realizan a los pines del protocolo SPI (SDA-10, SCK-13, MOSI-11 y MISO-12).



RFID	Arduino
SDA o SS	D10
SCK	D13
MOSI	D11
MISO	D12

IORQ	
GND	GND
RST	D9
3.3V	3,3V

Aunque en el esquema no esta, para un desarrollo en producción es conveniente utilizar un pequeño conversor de niveles lógicos de 5V/3.3 en los canales SPI, ya que nuestro arduino utiliza 5V y el lector recomienda 3.3V.

a) Breve introducción SPI

Protocolo de datos sincrono, con 4 cables, para comunicar dispositivos.

- **MISO(D12/ICSP-1)**: Por donde el maestro recibe los datos de los esclavos.
- **MOSI(D11/ICSP-4)**: Por donde el maestro enviá datos a los esclavos.
- **SCK(D13/ICSP-3)**: Reloj para sincronizar las transmisiones y es generado por el maestro.
- **SS(D10 en el esclavo)**: Pin por el que el maestro habilita o no a los dispositivos

esclavos, el maestro dispondrá de diferentes SSx. Lógica inversa es decir a bajo comunica.

Algunas funciones de la librería SPI:

- *begin()*: Inicializa el bus SPI
- *end()*: Desactiva el bus SPI
- *transfer(valor)/transfer16(val)*: Enviá uno o dos bytes y recibe la respuesta.
- *transfer(buffer,tamaño)*: Envía los datos que se indican y el retorno va a ese mismo buffer de tipo array.

3.2. Programación con MFRC522

La programación la realizaremos con la librería de Miguel Balboa MFRC522 y tiene las siguientes características:

- Protocolo de comunicaciones con el lector "PCD" SPI
- Solo admite el encriptado *Crypto1* y que no se aconseja utilizar pues fue roto hace años.
- PICC (Tarjetas) 13,56 MHz
- PICC ISO 14443A (Mifare Classic, S50, NTAG203, NTAG213)

Lo primero que hacemos es crea un objeto de la clase MFRC522, *antes de la función setup()* para que el objeto sea global a todo el programa

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

Ya en la función setup() inciamos el bus SPI y el lector RFID y si queremos podemos imprimir el firmware del lector.

```
SPI.begin(9600);
rfid.PCD_Init(); // Inicia el lector.
rfid.PCD_DumpVersionToSerial(); // Imprime Firmware, es opcional.
```

Ya en la función *loop()* esperaremos a tener tarjeta para leerla y realizar operaciones con ella.

```
if (! rfid.PICC_IsNewCardPresent()) return; // Si no detecta tarjeta sale
if (! rfid.PICC_ReadCardSerial()) return;  // Lee los datos de la tarjeta, si da error sale.
```

Podemos recoger el tipo de tarjeta, para saber si debemos o no tratarla.

```
piccTipo = rfid.PICC_GetType(rfid.uid.sak);  // Recoge el tipo de tarjeta PIC.
Serial.print("Tipo PICC: "+ String(rfid.PICC_GetTypeName(piccTipo)));
```

- ***PICC_Type PICC_GetType(byte)***

  Nos da Tipo Tarjeta pasandole el sak después de una lectura.

Según el modelo de tarjeta sabremos como están estructurados los datos y que comandos podremos ejecutar.

- ***String PICC_GetTypeName(PICC_Type)***

  Nos retorna modelo de tarjeta.

Las funciones anteriores realizan una primera lectura de los datos de fabrica: *UID* y *SAK,* que permiten saber el tipo de tarjeta para de esta forma saber como operar con ellas.

Así en una tarjeta **PICC_TYPE_MIFARE_1K** es la classic 1K y la **PICC_TYPE_MIFARE_UL** es la Mifare UltraLight, dependiendo de la tarjeta el flujo y forma de lectura y escritura es distintos por eso para má detalle de la forma estructuración de la memoria y lecturas deberemos leer el apartado correspondiente a la tarjeta. Y si deseamos profundizar más leer los documentos tecnicos que estan en los anexos.

Para una **Mifare classic 1K** debemos autentificarnos para cada sector que deseemos leer o escribir y el codigo que usaremospara

esa autentificación es

```
MFRC522::MIFARE_Key clave;
byte Bloque0 = sector*4; // Calculamos direccion bloque 0 del sector
byte dirBloque = Bloque0 + bloque;  // Direccion del bloque a leer
// Clave por defecto 0xFFFFFFFFFFFF
for(byte i =0 ; i< 6 ; i++) clave.keyByte[i] = 0xFF;
estado = rfid.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A,  // Tipo A
                                Bloque0, // Dirección Bloque 0 del sector
                               &clave,
                               &rfid.uid);
```

Una vez la autenticación nos ha dado *MFRC522::STATUS_OK* podemos empezar a leer.

En caso de querer autentificarnos con la clave B seria *"MFRC522::PICC_CMD_MF_AUTH_KEY_B"*

Las claves ya sean la A o la B dan permiso para en ese sector a las operaciones siguientes:

- Lee un bloque de memoria.
- Escribe en los bloques de memoria.
- Incrementa el contenido de un bloque y lo almacena en un registro dato.
- Resta el contenido de un bloque y lo almacena en un registro dato.
- Trasfiere de un registro dato a un bloque valor.
- Lee los datos de un bloque valor y lo pasa a un registro dato.

En el *sector trailer solo podremos leer y escribir, en los otros bloque ademas de lectura y escritura podremos realizar el resto de operaciones.*

Los permisos de los bloques de datos y del *sector trailer* se definen en 3 bits (Access Bits) que están en el *sector Trailer* y que tenemos más detalles en la descripción de la tarjeta.

En el caso de **Ultralight** podremos leer las autentificación en principio.

- ***MFRC522::StatusCod MIFARE_Read****(byte, *byte, *byte)*

  Nos permite leer un bloque de un sectores

  - *Pasando la dirección del sector a leer.*
  - *El buffer donde almacenara los datos, debe ser al menos de 18 bytes (16 datos 2 de CRC)*
  - *Una variable que indicara los byte leídos.*

- ***MFRC522::StatusCod MIFARE_Write****(byte,*

**byte, byte)*

Nos permite escribir un bloque de un sectores

- *Pasando la dirección del sector a escribir. El buffer con los datos, 16 bytes.*
- *bytes a escribir.*

NOTAS MifareClassic

- *Autentificarse a cada lectura del sector.*
- *Deberemos tener cuidado cuando se escribe en el bloque 3 que esta la configuración de acceso pues podríamos bloquear con una clave desconocida o limitar accesos.*

NOTAS Ultralight

- Cuidado con los Bytes 2 y 3 de la pagina 2 pues bloquean la escritura de paginas para siempre.
- Cuidado también con los tres bytes de bloqueos permanente de la página 0x40 de una tarjeta NTAG213, que es para bloquear el resto de las paginas.

a) Otras funciones de la librería

- PICC_HaltA() Finaliza lectura con la tarjeta.

b) Otras librerías

La librería de Miguel Balboa se basa en el código de Dr.Leong y de ella se han adaptado otras como:

- Paul Kourany adaptación para Spark.

4. TARJETAS

Tarjetas soportadas por el RFC522 comunicación 13,56MHz, ISO/IEC 14443, A/MIFARE y NTAG (NFC).

- MF1xxS20
- MF1xxS70 “Mifare Classic 4K”
- **MF1xxS50** “Mifare Classic EV1 1K”
  - MF1S503x “Tag Mifare 1K”
- MIFARE Mini
- MIFARE Ultralight
  - NTAG 210(80 bytes)
  - NTAG 212(164 bytes)
  - **NTAG 213(144 bytes)**
  - NTAG 215(504 bytes)
  - NTAG 216 (888 Bytes)
- MIFARE DESFire EV1
- MIFARE Plus RF

- NTAG 210(80 bytes), 212(164 bytes), 213(144 bytes),215(504 bytes),216 (888

Bytes)

4.1. TAG MIFARE 1 K

Tarjeta MF1S503x del grupo de las MIFARE_Classic, 4 bytes de UID, tambien llamado NIUD.

Características:

- Anticolisión
- 1k (16 sectores de 4 bloques de 16 bytes)
- ISO 14443A
- Algoritmo de encriptación CRYPTO1

La secuencia de operación de estas tarjetas es una vez seleccionada la tarjeta (ya tenemos el SAK conocemos el tipo pediremos autenticación correspondiente (tipo A o B) para cada bloque que deseemos realiza una operación (leer, escribir, operaciones) y una vez hayamos operado con los sectores necesarios ejecutaremos “HALT” para poder seleccionar otra tarjeta.

De los 4 bloques del sector (0-3) el bloque 3

de cada sector es el llamado sector trailer, por lo tanto tendremos 16 sectores trailer (0-15) los otros son bloques de datos. El Bloque 0 del sector 0 es el bloque especial y es información del fabricante, por lo tanto el sector 0 solo contara con dos bloques de datos.

Los sectores están protegidos cada uno por dos claves, permitiendo y bloqueando así la lectura o la escritura.

a) Bloque fabricantes

Bloque 0 / Sector 0															
0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
NUID				BCC	Datos del fabricante										

- **NUID**: Identificador único de tarjeta.
- **BCC**: CRC que es un XOR del NUID
- **Byte 5**: SAK que es 08
- **Byte 6 y 7**: ATQA que es un 0004

b) Bloque de datos

Como ya hemos comentado anteriormente cada sector contiene 3 bloques de datos, del 0 al 2, a excepción del sector 0 que solo tendrá 2 que serán el bloque 1 y el 2.

Un bloque de datos podrá como bloque de lectura/escritura o como bloque valor, y se definen en los bits de acceso que se explicara al hablar del sector trailer.

Lectura / escritura

Es un bloque simple para almacenar datos ASCII.

Bloque Valor

Nos permite ejecutar comando sobre ellos, suelen usarse para tarjetas monedero, y deben autentificarse para permitir cualquier información sobre ellos.

Las operaciones que podremos realizar son: leer,escribir, incrementar, decrementar, restaurar, transferir.

Estos bloques tienen un formato fijo que

es el siguiente.

Bloque Valor															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Valor				Valor Inv.				Valor				A	AI	A	AI

- **Valor:** El byte significativo más bajo de un valor se almacena en complemento a 2, y se almacena 3 veces, la primera y la tercera normal pero la del medio esta invertida.
- **Adr.** (A y AI el invertido) una dirección de 1 byte que se utiliza para guardar la dirección del bloque de copia de seguridad.

c) Sector trailer

Se encuentra en el cuatro bloque, que llamamos bloque 3, de los sectores y su misión es tener la información de permisos y códigos de acceso de los datos de cada sector.

Sector trailer															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Sector trailer		
Key A	**Bits acceso**	**Key B**

En una tarjeta nueva las claves A y B estarán a 0xF y la de los bits de acceso 0xFF078069, que permite el acceso de lectura/escritura total.

Los Bits de acceso estarán en su forma directa e invertida Byte-Bit/**Byte-Bit** del Bit1 al Bit3:

Bloque 3: 7-7/**6-3** 8-3/**6-7** 8-7/**7-3**

Bloque 2: 7-6/**6-2** 8-2/**6-6** 7-6/**7-2**

Bloque 1: 7-5/**6-1** 8-1/**6-5** 8-5/**7-1**

Bloque 0: 7-4/**6-0** 8-0/**6-4** 8-4/**7-0**

Acceso Sector Trailer Bloque 0								
Bits Acceso			**KEY A**		**Bit Acc.**		**KEY B**	
C1	**C2**	**C3**	**L**	**E**	**L**	**E**	**L**	**E**
0	0	0	N	A	A	N	A	A
0	1	0	N	N	A	N	A	N
1	0	0	N	B	AB	N	N	B
1	1	0	N	N	AB	N	N	N
0	0	1	N	A	A	A	A	A

0	1	1	N	B	AB	B	N	B
1	0	1	N	N	AB	B	N	N
1	1	1	N	N	AB	N	N	N

Escritura(**E**), Lectura (**L**)
Nada (**N**), Clave A (**A**), Clave B (**B**) y ambas claves (**AB**)

Y la tabla para los bloques de datos

Acceso bloques de datos							
Bits Acceso			**Condiciones acceso**				
C1	**C2**	**C3**	**L**	**E**	**I**	**DTR**	**B.Va**
0	0	0	AB	AB	AB	AB	
0	1	0	AB	N	N	N	
1	0	0	AB	B	N	N	
1	1	0	AB	B	B	AB	Si
0	0	1	AB	N	N	AB	Si
0	1	1	B	B	N	N	
1	0	1	B	N	N	N	
1	1	1	N	N	N	N	N

Escritura(**E**), Lectura (**L**), Incremento(I) y Decremento, transferencia y restaura (DTR)
Nada (**N**), Clave A (**A**), Clave B (**B**) y ambas claves (**AB**)
(B.Va) Bloque Valor

```
Firmware Version: 0x91 = v1.0
Volcado de informacion de tarjetas  RFID (Fco. Javier Rodriguez Navarro)
A la espera de leer una tarjeta
Card UID: 4C 87 F1 C5
Card SAK: 08
PICC type: MIFARE 1KB
Volcado de tarjeta 0x08 Classic 1K
NUID(4) 4C 87 F1 C5 SAK: 8
Dir.   Sector  Bloque   0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 [Acceso]
0      0       0        4C 87 F1 C5 FF 08 04 00 62 63 64 65 66 67 68 69   [000]  NUID: 4c87f1c5
```

```
                                        CRC:ff
                                        SAK:8
                                        ATQA:80
                                        Fabricante: b c d e f g h i
1    0    1    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   [000]  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0
2    0    2    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   [000]  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0
3    0    3    00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF   [001] KEY A:( 0x0  0x0  0x0  0x0  0x0  0x0 ) KEY B:( 0xFF  0xFF  0xFF  0xFF  0xFF  0xFF ) Usu:(i )
-----------------------
4    1    0    28 63 29 47 41 52 55 4D 20 32 30 31 37 20 20 20   [000] ( c ) G A R U M  0x20 2 0 1 7  0x20  0x20  0x20
5    1    1    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   [000]  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0
6    1    2    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   [000]  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0
7    1    3    00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF   [001] KEY A:( 0x0  0x0  0x0  0x0  0x0  0x0 ) KEY B:( 0xFF  0xFF  0xFF  0xFF  0xFF  0xFF ) Usu:(i )
-----------------------
8    2    0    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   [000]  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0
9    2    1    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   [000]  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0
10   2    2     00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   [000]  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0
11   2    3     00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF   [001] KEY A:( 0x0  0x0  0x0  0x0  0x0  0x0 ) KEY B:( 0xFF  0xFF  0xFF  0xFF  0xFF  0xFF ) Usu:(i )
-----------------------
12   3    0    4E 6F 20 73 61 62 65 6D 6F 73 20 71 75 65 20 65   [000] N o  0x20 s a b e m o s  0x20 q u e  0x20 e
13   3    1    50 72 75 65 62 61 20 64 65 20 63 61 72 61 63 74   [000] P r u e b a  0x20 d e  0x20 c a r a c t
14   3    2    28 63 29 20 47 41 52 55 4D 20 32 30 31 37 20 4D   [000] ( c )  0x20 G A R U M  0x20 2 0 1 7  0x20 M
15   3    3    00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF   [001] KEY A:( 0x0  0x0  0x0  0x0  0x0  0x0 ) KEY B:( 0xFF  0xFF  0xFF  0xFF  0xFF  0xFF ) Usu:(i )
-----------------------
16   4    0    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   [000]  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0
17   4    1    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   [000]  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0
18   4    2    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   [000]  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0
19   4    3    00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF   [001] KEY A:( 0x0  0x0  0x0  0x0  0x0  0x0 ) KEY B:( 0xFF  0xFF  0xFF  0xFF  0xFF  0xFF ) Usu:(i )
-----------------------
                              .............


-----------------------
60   15   0    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   [000]  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0
61   15   1    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   [000]  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0
62   15   2    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   [000]  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0  0x0
63   15   3    00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF   [001] KEY A:( 0x0  0x0  0x0  0x0  0x0  0x0 ) KEY B:( 0xFF  0xFF  0xFF  0xFF  0xFF  0xFF ) Usu:(i )
-----------------------
```

4.2. TAG MIFARE Ultralight

Tarjeta Ultralight del grupo NTAG21x (**NTAG213 180**, NTAG215 540,NTAG216 924) pertenecen al grupo 2 de las NFCs

Características:

- UID de 7 Bytes
- Anticolisión
- Paginas de 4 bytes (para el NTAG213 45 y

36 para usuario)

- ISO 14443A 13,56 MHz

La organización de la memoria es la siguiente:

Pagina	**Byte 0**	**Byte 1**	**Byte 2**	**Byte 3**
0x00	Numero de serie			
0x01	Numero de serie			
0x02	N/S	Inter.	Lock S	Lock S
0x03	Capacidad (Byte2)			
0x04-0xYY	Usuario finaliza 0x27-NTAG213 0x81 NTAG215 0xE2 NTAG216			
0xYY+1	Dynamic Lock			RFUI
0xYY+2	CFG0			
0xYY+3	CFG1			
0xYY+4	PWD			
0xYY+5	PACK		RFUI	

El byte 2 de la pagina 3 nos dará la capacidad de usuario y así por lo tanto el modelo de tarjeta.

- **0x12** NTAG213 (144 Bytes usuario) 0x04-0x027 que son 36 paginas.
- **0x3E** NTAG215 (496 Bytes 0x04-0x81)

- **0x6D** NTAG216 (872 Bytes 0x04-0xE1)

a) Static-Lock

Los Bytes 2 y 3 de la pagina 2 son el mecanismo de bloqueo para solo lectura de las páginas 3(CC) a la 15. Se bloquean individualmente la escritura poniendo 1, son los bits L, también se puede bloquear por bloques y son los registros BL.

Byte 2: L7 L6 L5 L4 L3(CC) BL10-15 BL 4-9 BLCC

Byte 3: L15 L14 L13 L12 L11 L10 L9 L8

OJO una vez se pone a uno ya no puede ponerse a 0 y por lo tanto ya no podremos escribir en las paginas protegidas.

b) Dinamic-Lock

Nos permite bloquear las paginas restante de la 0x10-en adelante, se encuentra en la pagina 0x28(40) para NTAG213 y consta de 3 bytes con una granularidad de 2 paginas, en el caso de NTAG215 NTAG216 16 paginas, y podremos bloquear grupos de dos “L” o de 4 BL.

Byte 0: L30 L28 L26 L24 L22 L20 L18 L16

Byte 1: 0 0 0 0 L38 L36 L34 L32

Byte 2: 0 0 BL36 BL32 BL28 BL24 BL20 BL16

Byte 3: RFUI

c) CFG0, CFG1, PWD y PACK

Nos permite configurar restricciones de acceso, configuración de contadores.

```
A la espera de leer una tarjeta
Card UID: 04 43 D5 42 E7 4C 81
Card SAK: 00
PICC type: MIFARE Ultralight or Ultralight C
Volcado de tarjeta 0x00 Ultralight
UID(7) 04 43 D5 42 E7 4C 81 SAK: 0
Pagina:         0  1  2  3     Descripción
0x00(0)         04 43 D5 1A    Fabricante: 0x4 N/S 0x43 0xd5 CRC(0) 0x1a
0x01(1)         42 E7 4C 81    N/S:  42 E7 4C 81
0x02(2)         68 48 00 00    CRC(1): 0x68 Interno 0x48
                        Bloqueo Estatico: L7-3 BL10-15,4-9 CC: 0x0
                                                L15-8 0x0
0x03(3)         E1 10 12 00    Capability Container: NTAG213 0x04-0x39 de usuario
USU 0x4 4       01 03 A0 0C ( 0x1  0x3  0xA0  0xC )
USU 0x5 5       34 03 00 FE (4  0x3  0x0  0xFE )
USU 0x6 6       00 00 00 00 ( 0x0  0x0  0x0  0x0 )
USU 0x7 7       00 00 00 00 ( 0x0  0x0  0x0  0x0 )
USU 0x8 8       00 00 00 00 ( 0x0  0x0  0x0  0x0 )
USU 0x9 9       00 00 00 00 ( 0x0  0x0  0x0  0x0 )
USU 0xa 10      00 00 00 00 ( 0x0  0x0  0x0  0x0 )
USU 0xb 11      00 00 00 00 ( 0x0  0x0  0x0  0x0 )
USU 0xc 12      00 00 00 00 ( 0x0  0x0  0x0  0x0 )
USU 0xd 13      00 00 00 00 ( 0x0  0x0  0x0  0x0 )
USU 0xe 14      00 00 00 00 ( 0x0  0x0  0x0  0x0 )
USU 0xf 15      00 00 00 00 ( 0x0  0x0  0x0  0x0 )
USU 0x10 16      32 2E 2D 20 (2 . -  0x20 )
USU 0x11 17      43 72 75 7A (C r u z )
USU 0x12 18      0A 32 30 31 ( 0xA 2 0 1 )
USU 0x13 19      37 20 20 20 (7  0x20  0x20  0x20 )
USU 0x14 20      00 00 00 00 ( 0x0  0x0  0x0  0x0 )
                        ........
USU 0x27 39      00 00 00 00 ( 0x0  0x0  0x0  0x0 )
0x28 40         00 00 00 BD    Bloqueo Dinamico para NTAG213:
                        L30 L28 L26 L24 L22 L20 L18 L16: 0x0
                        0 0 0 0 L38 L36 L34 L32: 0x0
                        0 0 BL36 BL32 BL28 BL24 BL20 BL16: 0x0
0x29 41         04 00 00 FF    CFG 0
0x2A 42         00 05 00 00    CFG 1
0x2B 43         00 00 00 00    PWD
```

0x2C 44	00 00 00 00	Byte 0-1 PACK y Byte 2-3 RFUI

5. ANEXO EJEMPLOS

- http://www.pinguytaz.net/M_Archivos/RFID/RFID_Volcado1K_V2.ino

  Realiza un volcado de una tarjeta MIFARE de 1K, podemos así ver como leer una tarjeta.

- http://www.pinguytaz.net/M_Archivos/RFID/RFID_Volcado.ino

  Realiza un volcado de una tarjeta MIFARE Classic o la Ultralight.

- http://www.pinguytaz.net/M_Archivos/RFID/RFID_Lectura1K_V1.ino

  Lectura de un sector de la tarjeta RFID Mifare Classic de 1K.

- http://www.pinguytaz.net/M_Archivos/RFID/RFID_Escritura1K_V2.ino

  Escribimos datos en un sector y un bloque de la tarjeta RFID Mifare Classic de 1K.

- http://www.pinguytaz.net/M_Archivos/RFID/RFID_EscrituraNTAG213.ino

  Escribimos datos en una tarjeta Ultralight, exactamente en una NTAG213.

6. ANEXO ENLACES

- CHIP MFRC522

  https://www.nxp.com/documents/data_sheet/MFRC522.pdf

- Tarjeta Mifare Classic 1K (S50)

  http://cache.nxp.com/documents/data_sheet/MF1S50YYX_V1.pdf

  - Autentificacion y anticolisión

    http://www.nxp.com/documents/application_note/AN10833.pdf

  - UID

    http://www.nxp.com/documents/application_note/AN10927.pdf

- Tarjeta Mifare Ultralight NTAG 213

  https://www.nxp.com/docs/en/data-sheet/NTAG213_215_216.pdf

- Librería MFRC522

  https://github.com/miguelbalboa/rfid

  - Paul Kourany
    https://github.com/pkourany/MFRC522_RFID_Library

https://github.com/pkourany/RC522_RFID

Tecnologia RFID y NFC, para poder leer y escribir TAG que nos permitan identificar objetos y autorizaciones.