

## SENSORES Y ACTUADORES

Los dispositivos a usar serán los siguientes

1. Lector RFID (MFRC522, con una tarjeta y un llavero MIFARE CLASSIC S50)



El RFID es un conjunto de tecnologías inalámbricas diseñadas para obtener una información almacenada en un dispositivo denominado etiqueta (tag)

El lector (transceptor) es en realidad un emisor-receptor que, en primer lugar, emite una señal para iniciar la comunicación con las etiquetas (transpondedores). 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.

El Mifare MFRC522 es un lector de tarjetas RFID que incorpora comunicación por bus SPI, bus I2C y UART, por lo que es sencillo de conectar con Arduino. El MFRC522 soporta las tarjetas Mifate S50, Mifare S70, Mifare UltraLight, Mifare Pro y Mifare Desfire.

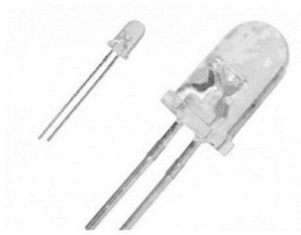
El siguiente código nos permite leer el ID de la tarjeta y comprobar si este es válido, nuestra idea es implementarlo como manera de acceso a una supuesta biblioteca y que dependiendo de la hora a la que se haya reservado la lectura será válida o no.

```

1.  //RST          D9
2.  //SDA(SS)      D10
3.  //MOSI         D11
4.  //MISO         D12
5.  //SCK          D13
6.
7.  #include <SPI.h>
8.  #include <MFRC522.h>
9.
10. const int RST_PIN = 9;          // Pin 9 para el reset del RC522
11. const int SS_PIN = 10;         // Pin 10 para el SS (SDA) del RC522
12. MFRC522 mfrc522(SS_PIN, RST_PIN); // Crear instancia del MFRC522
13.
14. byte validKey1[4] = { 0xA0, 0xB1, 0xC2, 0xD3 }; // Ejemplo de clave
15.
16. //Función para comparar dos vectores
17. bool isEqualArray(byte* arrayA, byte* arrayB, int length)
18. {
19.     for (int index = 0; index < length; index++)
20.     {
21.         if (arrayA[index] != arrayB[index]) return false;
22.     }
23.     return true;
24. }
25.
26. void setup() {
27.     Serial.begin(9600); // Iniciar serial
28.     SPI.begin();        // Iniciar SPI
29.     mfrc522.PCD_Init(); // Iniciar MFRC522
30. }
31.
32. void loop() {
33.     // Detectar tarjeta
34.     if (mfrc522.PICC_IsNewCardPresent())
35.     {
36.         //Seleccionamos una tarjeta
37.         if (mfrc522.PICC_ReadCardSerial())
38.         {
39.             // Comparar ID con las claves válidas
40.             if (isEqualArray(mfrc522.uid.uidByte, validKey1, 4))
41.                 Serial.println("Tarjeta valida");
42.             else
43.                 Serial.println("Tarjeta invalida");
44.
45.             // Finalizar lectura actual
46.             mfrc522.PICC_HaltA();
47.         }
48.     }
49.     delay(250);
50. }
51. }

```

## 2. Leds Arduino rojo y verde



Un LED es un diodo emisor de luz, es decir, un tipo particular de diodo que emite luz al ser atravesado por una corriente eléctrica. Los diodos (emisor de luz, o no) son unos de los dispositivos electrónicos fundamentales.

Emplearemos estos leds con un código de activación de estos que funcionen cuando se den determinados sucesos como el acierto/fallo de una contraseña.

Este es el código de activación de un led, cuando lo realicemos nosotros este código será utilizado 2 veces, una por cada led, y asignando a cada uno puertos diferentes.

```
1.  const int ledPIN = 9;
2.
3.  void setup() {
4.      Serial.begin(9600);    //iniciar puerto serie
5.      pinMode(ledPIN , OUTPUT); //definir pin como salida
6.  }
7.
8.  void loop(){
9.      digitalWrite(ledPIN , HIGH);    // poner el Pin en HIGH
10.     delay(1000);                     // esperar un segundo
11.     digitalWrite(ledPIN , LOW);      // poner el Pin en LOW
12.     delay(1000);                     // esperar un segundo
13. }
```

## 3. Alarma Arduino y Buzzer activo



Los buzzer activos, en ocasiones denominados zumbadores, son dispositivos que generan un sonido de una frecuencia determinada y fija cuando son conectados a tensión.

Un buzzer activo incorpora un oscilador simple por lo que únicamente es necesario suministrar corriente al dispositivo para que emita sonido.

Al incorporar de forma interna la electrónica necesaria para hacer vibrar el altavoz un buzzer activo resulta muy sencillo de conectar y controlar. Además, no suponen carga para el procesador ya que no este no tiene que generar la onda eléctrica que se convertirá en sonido.

En contra posición, tienen la desventaja de que no podremos variar el tono del sonido emitido lo cual no es nuestra intención ya que solo se usará como un recordatorio para fechas que sean marcadas en el programa, sonando al comienzo del intervalo de tiempo correspondiente.

El código de activación es el siguiente.

```
1.  const int pin = 9;
2.
3.  bool isAlarmOn = 0;    //almacena el estado de la alarma
4.
5.  void setup() {
6.      pinMode(pin, OUTPUT); //definir pin como salida
7.  }
8.
9.  bool GetSystemState()
10. {
11.     return true; //cambiar en función del sensor usado
12. }
13.
14. void loop() {
15.     isAlarmOn = GetSystemState();
16.
17.     if(isAlarmOn)
18.     {
19.         digitalWrite(pin, HIGH); // poner el Pin en HIGH
20.         delay(5000);             // esperar 5 segundos
21.     }
22.     else
23.     {
24.         digitalWrite(pin, LOW);  // poner el Pin en LOW
25.     }
26.     delay(1000);
27. }
```

Las páginas de donde se ha obtenido esta información son:

<https://www.luisllamas.es/arduino-rfid-mifare-rc522/>

<https://www.luisllamas.es/encender-un-led-con-arduino/>

<https://www.luisllamas.es/arduino-buzzer-activo/>

