



Buscar



S/. 0,00

CATEGORÍAS

[All Blog News](#)[Tutorial Arduino y memoria SD y micro SD.](#)

## ARTÍCULOS POPULARES

[Tutorial sensor de flujo de agua](#) abr 13, 2016[Configuración del módulo bluetooth HC-05 usando comandos AT](#) ago 08, 2015[Tutorial transmisor de celda de carga HX711, Balanza Digital](#) ago 23, 2015

## ARTÍCULOS RECIENTES

[Usando ESP8266 con el IDE de Arduino](#) oct 21, 2016[Tutorial Sensor de Distancia SHARP](#) sep 28, 2016[Arduino y ESP8266 como cliente web](#) ago 24, 2016

## PROMOCIONES ESPECIALES



Cable Dupont hembra a hembra 20cm / 20Und  
Cable Dupont hembra a hembra 20cm x...

S/. 5,00 ~~S/. 6,00~~

Todas las promociones especiales >

## ETIQUETAS



sensor

LED

cnc

regulador

robot

driver

LCD

infrarrojo

Servo

RGB

## NOVEDADES



Raspberry Pi Zero W

El Raspberry Pi más pequeño y más económico pero con la misma utilidad...

S/. 105,00



Raspberry Pi Zero

El Raspberry Pi más pequeño y más económico pero con la misma utilidad...

S/. 70,00



Pulsioxímetro MAX30102

Construye tu propio pulsioxímetro con la ayuda de este magnífico sensor.

S/. 34,00



Voltímetro/Amperímetro DC 100A

Medidor de bajo costo de voltaje y corriente DC hasta 100 amperios

S/. 75,00

**Motor DC N20 12V 100 RPM (300:1)**

Este motor es ideal para proyectos de robótica móvil como robots...

S/. 20,00

**Válvula Solenoide 3/4" baja presión 220VAC**

Esta válvula solenoide permite controlar el paso de agua, no necesita...

S/. 135,00

**Sensor de nivel XKC Y25 T12V**

Con este sensor puedes detectar la presencia de liquido a través de...

S/. 35,00

**LED W52811**

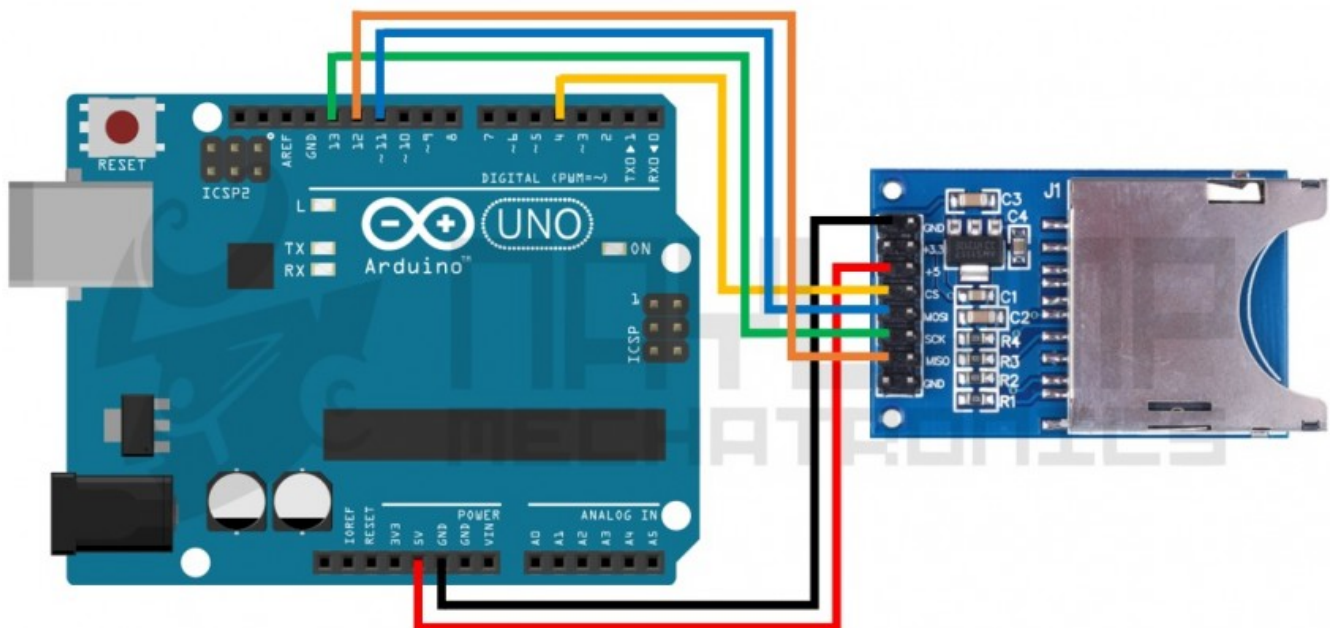
Agrega colores llamativos a tus proyectos con este led RGB digital de...

S/. 1,00

Todos los nuevos productos >

## Tutorial Arduino y memoria SD y micro SD.

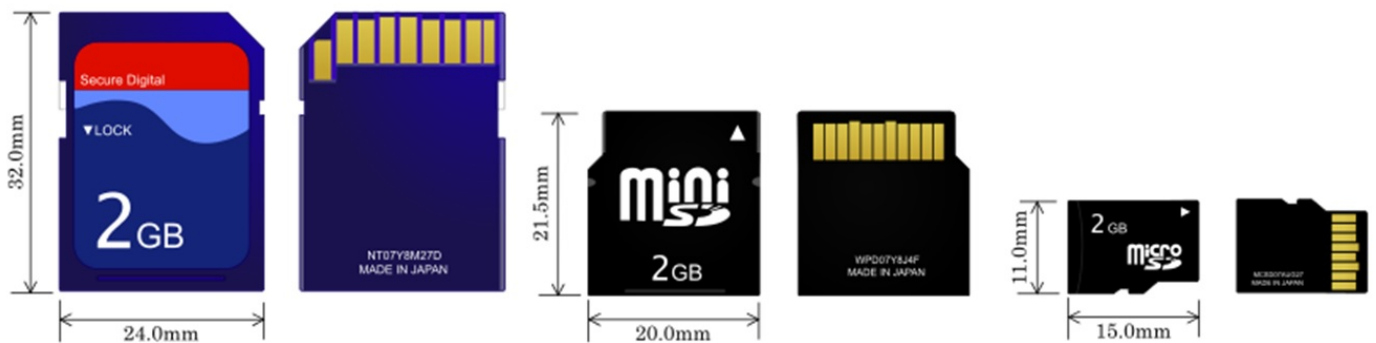
Posted by Tutoriales 52 Comments



Desarrollaremos ejemplos para trabajar con tarjetas SD o Micro SD, veremos las conexiones y características de estos dispositivos.

Las memorias SD son las más usadas por dispositivos portátiles, por su gran capacidad y su reducido tamaño, debido a su gran demanda son fáciles de conseguir en diferentes capacidades y precios. Estas características nos dan una buena alternativa de almacenamiento para usarlo en Arduino, sobretodo cuando necesitamos guarda gran cantidad de información.

Estas memorias vienen en tres tamaños, SD estándar, Mini SD y Micro SD, siendo este último el tamaño más común, funcionalmente son iguales, pudiéndose usar adaptadores para utilizarlos en sockets de diferente tamaño.



Con respecto al formato podemos encontrar 4 tipos, las tarjetas SD o SDSC (Standard Capacity), SDHC (High Capacity), SDXC (Extended Capacity) y las SDIO (Input/Output), permitiéndonos Arduino trabajar con los dos primeros tipos.

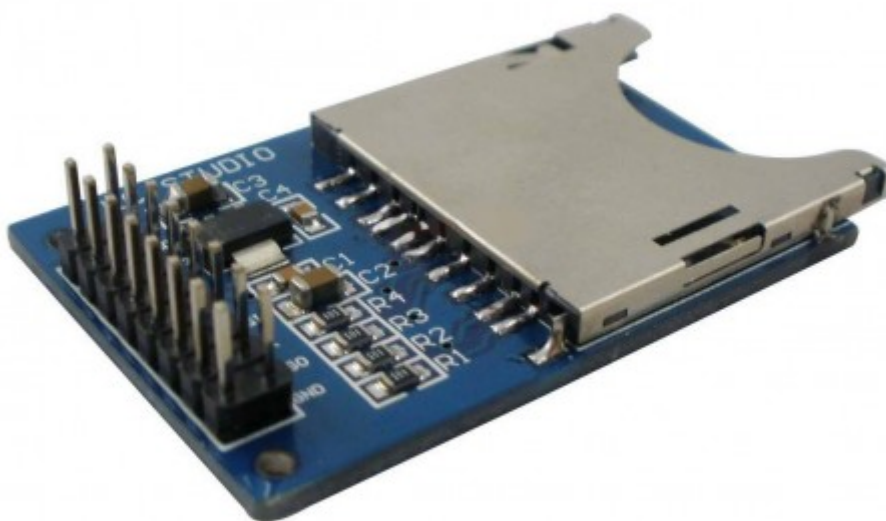
La comunicación de la memoria es por SPI pero trabajan con 3.3V, para utilizarlo con Arduino necesitamos módulos externos que aparte de tener el socket traen los componentes necesarios para adaptar los voltajes a TTL y poder conectarlo de forma fácil a nuestro Arduino.

Para este tutorial podemos usar cualquiera de estos dos módulos:

**Modulo Micro SD:** nos permite insertar una memoria Micro SD que son las más comunes en el mercado, el modulo se puede alimentar con 3.3V o 5V usando los pines respectivos.



**Módulo SD card:** Este módulo trae el socket grande para las memorias SD de tamaño estándar, pero usando un adaptador podemos usar también memorias micro SD



Arduino tiene una librería para usar estas memorias, que funciona con cualquiera de los módulos antes mencionados. La librería ya viene junto con el IDE de arduinos, por lo que no necesitamos instalar ni descargar nada.

Para poder usar la librería en nuestro Sketch es necesario incluir a la librería SD al inicio del código:

```
#include <SD.h>
```

A continuación explicamos las funciones principales de la librería SD, el cual es un resumen de la referencia proporcionada en la página oficial de Arduino: <https://www.arduino.cc/en/Reference/SD>

### **SD.begin(cspin)**

Inicializa la biblioteca SD y la tarjeta, como parámetro se le indica el pin CS al que está conectado el modulo, si no se especifica cspin , se usa el valor por defecto del CS por hardware. Los demás pines deben estar conectados al SPI por hardware del Arduino.

### **SD.exists(filename)**

Comprueba si existe el archivo especificado, filename es el nombre del archivo y/o directorio en la tarjeta SD si este existe la función nos retorna un true, de lo contrario retorna false.

### **SD.mkdir(directory)**

Crea el directorio especificado, si los subdirectorios no existen, también se crearan. Por ejemplo: SD.mkdir("Arduino/proyecto1/archivos), crea la carpeta "archivos" y si las carpetas Arduino y proyecto1 no existen, entonces también se crearan. La función retorna true si la creación del directorio fue exitosa de lo contrario nos retorna un false

### **SD.remove(filename)**

Elimina el archivo (filename) de la tarjeta SD, se debe de incluir el directorio. Solo elimina el archivo más no el directorio. Devuelve true se logra eliminar el archivo de lo contrario nos retorna un false.

### **SD.rmdir(dirname)**

Eliminar el directorio (dirname) de la tarjeta SD. El directorio debe estar vacío. Devuelve TRUE si la eliminación del directorio tuvo éxito o FALSE en caso contrario.

### **SD.open(filepath, mode)**

Abre el archivo especificado y se debe de incluir el directorio si el archivo está en carpetas. Si el archivo no existe, se creara un archivo con el nombre especificado, pero no será posible crear el directorio si este no existe. Se puede abrir un archivo como solo lectura (si mode es FILE\_READ) o como lectura y escritura (si mode es FILE\_WRITE), el modo por defecto en caso no se especifique es FILE\_READ

Ésta función nos retorna un objeto tipo FILE, el cual es necesario declararlo antes como una variable. Por ejemplo:

```
File myFile;  
myFile = SD.open("archivo.txt", FILE_WRITE);
```

### Funciones de la clase File:

#### **file.available()**

Compruebe si hay bytes disponibles para leer en el archivo y retorna el número de bytes disponibles

#### **file.read()**

Lee un byte de la variable File (archivo abierto anteriormente con SD.open())

#### **file.write(data)**

Escribe un byte en el archivo, el archivo debe estar abierto en modo lectura y escritura. Usando file.write(buf, len) se puede escribir un array de byte (buf) pero se debe especificar el tamaño (len).

#### **file.print(data)**

Esta función tiene las mismas características que un Serial.print(); data puede ser una variable o texto, el cual será enviado como caracteres. Si queremos agregar al final un salto o nueva línea se usa file.println(data)

#### **file.size()**

Retorna el tamaño en bytes del archivo

#### **file.position()**

Retorna la posición actual en donde se leerá o escribirá el siguiente byte.

#### **file.seek(pos)**

Nos ubicamos en una posición específica en el archivo. Pos debe ser un número entre 0 y el tamaño en bytes del archivo

#### **file.close()**

Cerramos el archivo, y recién en este momento los datos se guardan en la SD, pudiendo extraer de forma segura nuestra SD.

Explicado la librería empecemos viendo como conectar los módulos con nuestro Arduino:

## Conexión entre Arduino y modulo SD y micro SD

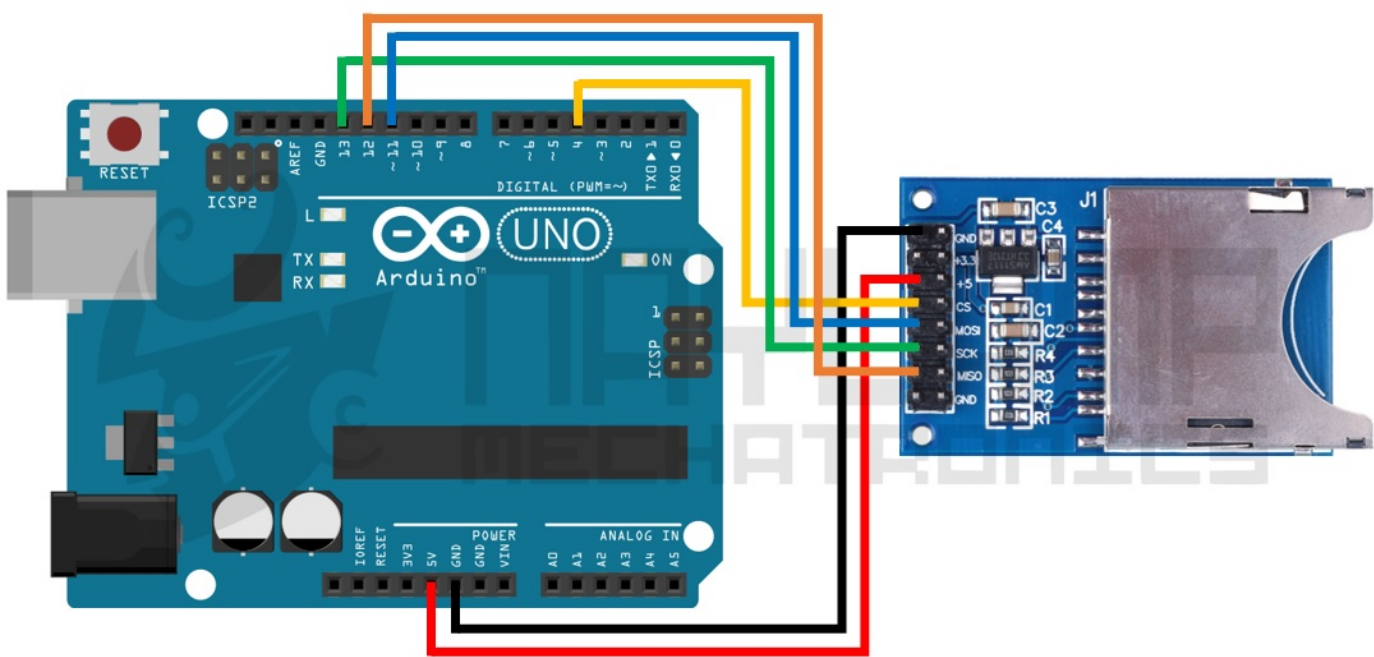
Las conexiones para el modulo SD son las siguientes;

Módulo SD	Arduino Uno, Nano	Arduino Mega
GND	GND	GND



Módulo SD	Arduino Uno, Nano	Arduino Mega
+3.3V	No conectado	No conectado
+5V	5V	5V
CS	4	4
MOSI	11	51
SCK	13	52
MISO	12	50
GND	GND	GND

\*Se puede alimentar con 5V o 3,3V usando los pines correspondientes, pero no se debe de alimentar por ambos pines a la vez



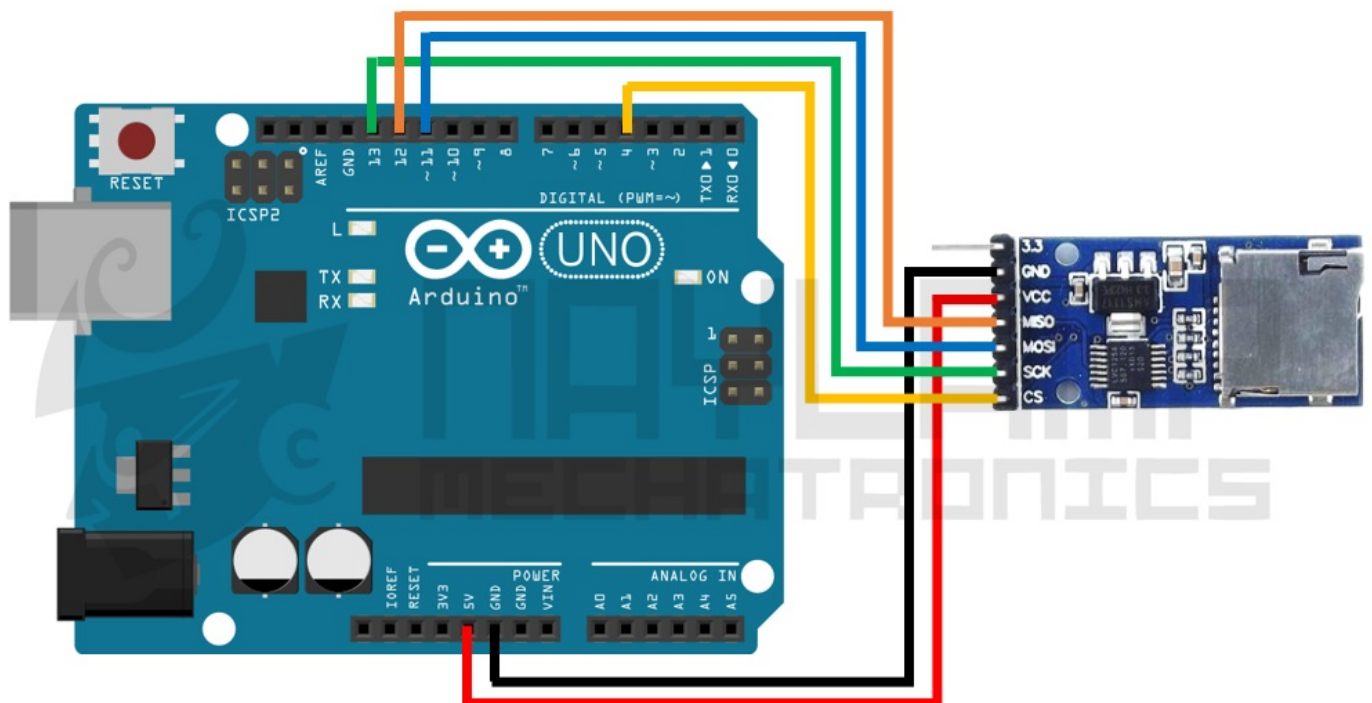
Y las conexiones si se trabajan con el módulo micro SD son:

Módulo SD	Arduino Uno, Nano	Arduino Mega
-----------	-------------------	--------------



Módulo SD	Arduino Uno, Nano	Arduino Mega
CS	4	4
SCK	13	52
MOSI	11	51
MISO	12	50
VCC	5V	5V
GND	GND	GND
3.3V	No conectado	No conectado

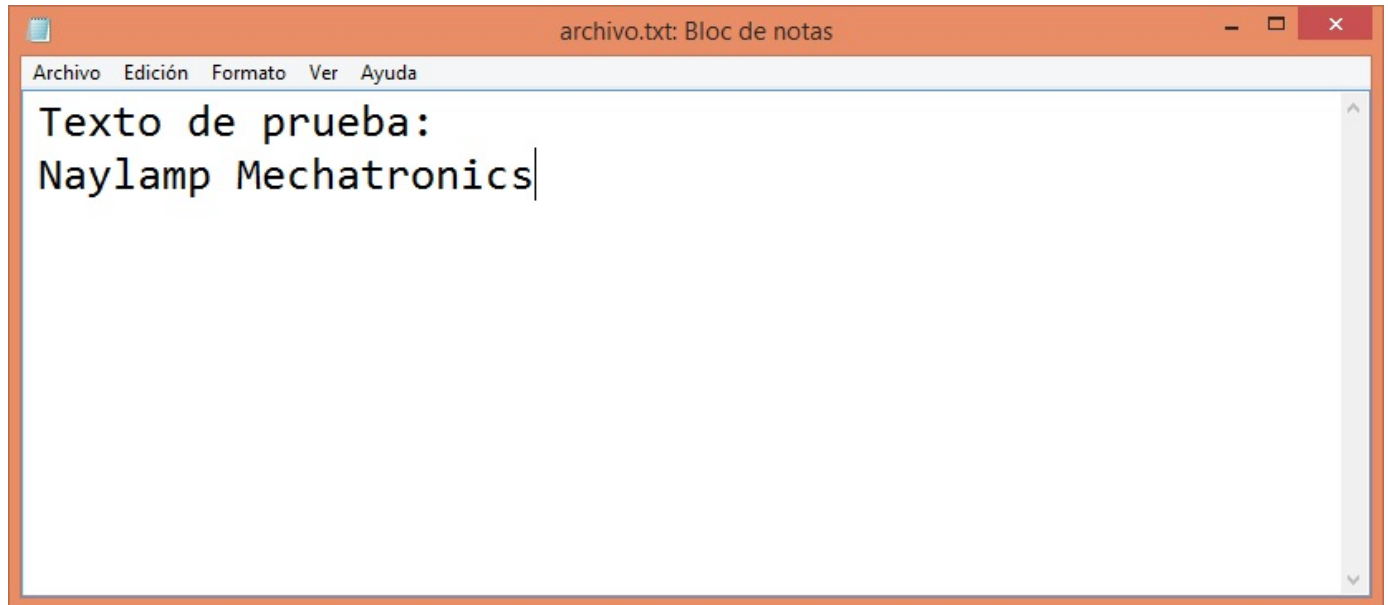
\*Se puede alimentar con 3,3V en lugar de 5V usando el pin correspondientes, pero no se debe de alimentar por ambos pines a la vez



## Ej1. Leer un archivo de la SD con nuestro Arduino

En este ejemplo vamos a leer un archivo de texto desde la SD.

Para esto insertamos nuestra SD a la PC, abrimos y creamos un archivo de hoja de texto, por ejemplo archivo.txt, e ingresamos el texto que posteriormente vamos a leer desde Arduino:



Seguidamente extraemos la SD de la PC, lo insertamos en el módulo SD. Y cargamos el siguiente sketch en nuestro Arduino:

```
#include <SD.h>

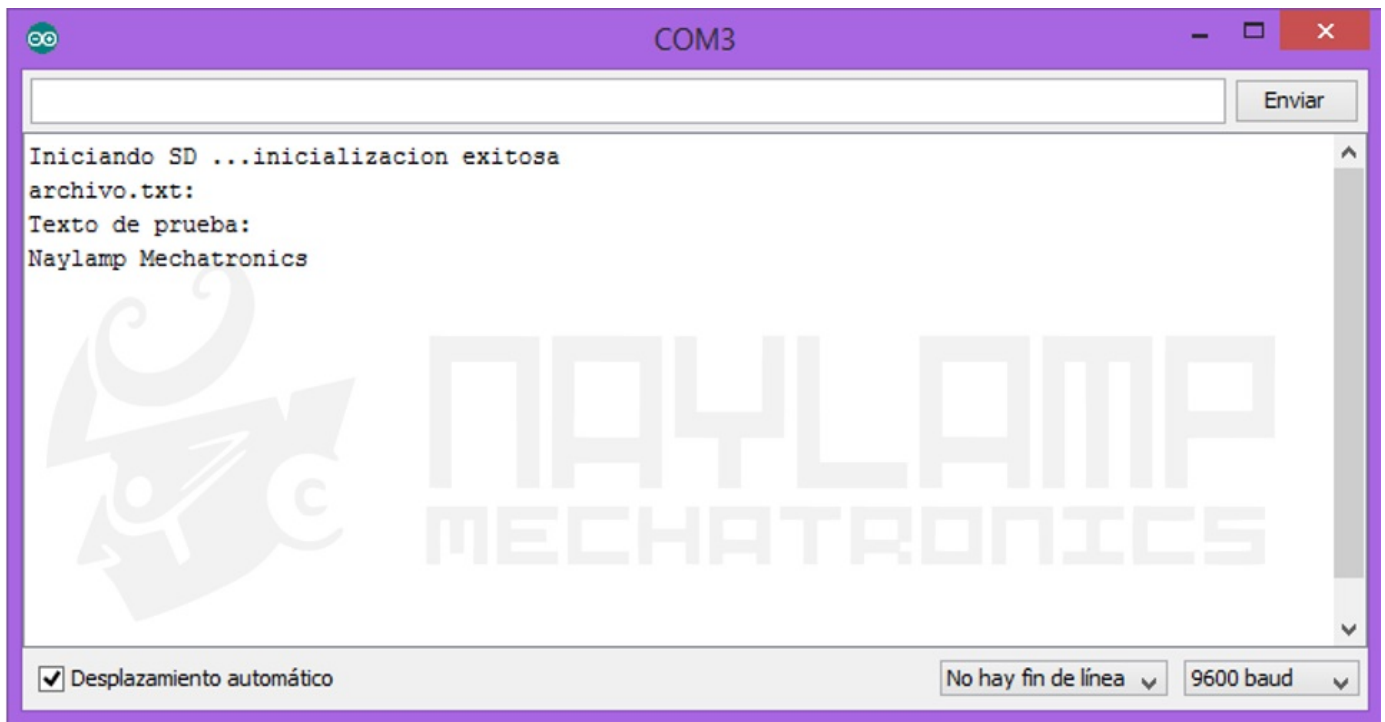
File myFile;

void setup()
{
  Serial.begin(9600);
  Serial.print("Iniciando SD ...");
  if (!SD.begin(4)) {
    Serial.println("No se pudo inicializar");
    return;
  }
  Serial.println("inicializacion exitosa");

  myFile = SD.open("archivo.txt");//abrimos el archivo
  if (myFile) {
    Serial.println("archivo.txt:");
    while (myFile.available()) {
      Serial.write(myFile.read());
    }
    myFile.close(); //cerramos el archivo
  } else {
    Serial.println("Error al abrir el archivo");
  }
}

void loop()
{
}
```

EL programa envía por el puerto serie todo el contenido del archivo guardado en la SD.



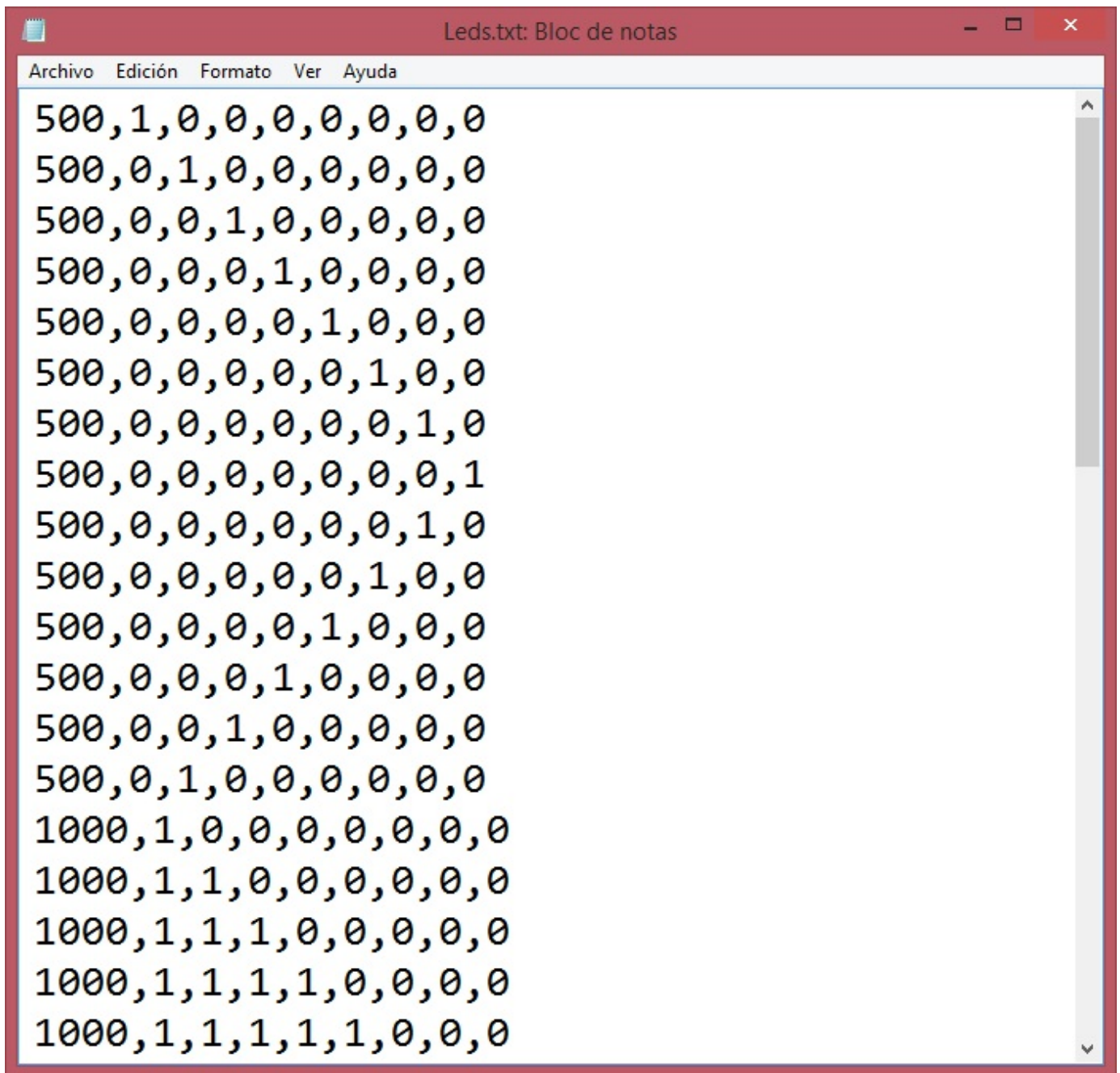
## Ej2. Leer datos desde la SD, Manipulando leds desde la SD

En este caso no solo vamos a leer el archivo desde la SD sino vamos a interpretar y realizar ciertas acciones con los datos.

En un archivo de texto vamos a guardar datos para encender y apagar leds y con esto ejecutar diferentes secuencias que estarán guardadas en la SD.

Conectamos ocho leds en los pines del 2 al 10, excluyendo el 4, con su resistencia respectiva.

Desde la PC en la SD en una hoja de texto guardamos las secuencias que deseemos realizar.



El primer número de cada línea corresponde a la pausa o tiempo que va a estar presente la secuencia, los siguientes ocho números corresponde a la secuencia o estado de cada uno de los ocho leds, ustedes pueden escribir cualquier secuencia o también pueden descargar el archivo de texto que usamos para este ejemplo: Leds.txt

Después de escribir nuestras secuencias de los leds en la hoja de texto guardamos con nombre Leds.txt e insertamos la SD en el módulo de nuestro Arduino

El Sketch es el siguiente.

```
#include <SD.h>

File myFile;
int UltimaPocicion=0;
int pausa=1000;
int PinLeds[8]={2,3,5,6,7,8,9,10}; //Pines de los Leds
void setup()
{
```

```

Serial.begin(9600);
Serial.print("Iniciando SD ...");
if (!SD.begin(4)) {
    Serial.println("No se pudo inicializar");
    return;
}
Serial.println("inicializacion exitosa");
for(int i=0;i<8;i++)
{
    pinMode(PinLeds[i],OUTPUT);
}

}

void loop()
{
    myFile = SD.open("Leds.txt");//abrimos el archivo
    int totalBytes=myFile.size();
    String cadena="";

    if (myFile) {
        if(UltimaPocicion>=totalBytes)    UltimaPocicion=0;
        myFile.seek(UltimaPocicion);

        //--Leemos una línea de la hoja de texto-----
        while (myFile.available()) {

            char character=myFile.read();
            cadena=cadena+character;
            UltimaPocicion=myFile.position();
            if(character==10)//ASCII de nueva de línea
            {
                break;
            }
        }
        //-------
        myFile.close(); //cerramos el archivo
        Serial.print("Cadena Leida:");
        Serial.print(cadena);
        //-------procesamos la cadena-----
        int index=0;
        char c=cadena[index++];
        pausa=0;
        while (c >= '0' && c <= '9')
        {
            pausa = 10*pausa + (c - '0');
            c = cadena[index++];
        }
        Serial.print("pausa=");
        Serial.print(pausa);
        Serial.print("    LEDS |");
        for(int i=0;i<8;i++)
        {
            if(cadena[index+i*2]=='1')
            {
                digitalWrite(PinLeds[i], HIGH);
                Serial.print(" 1 |");
            }
            else
            {

```

```

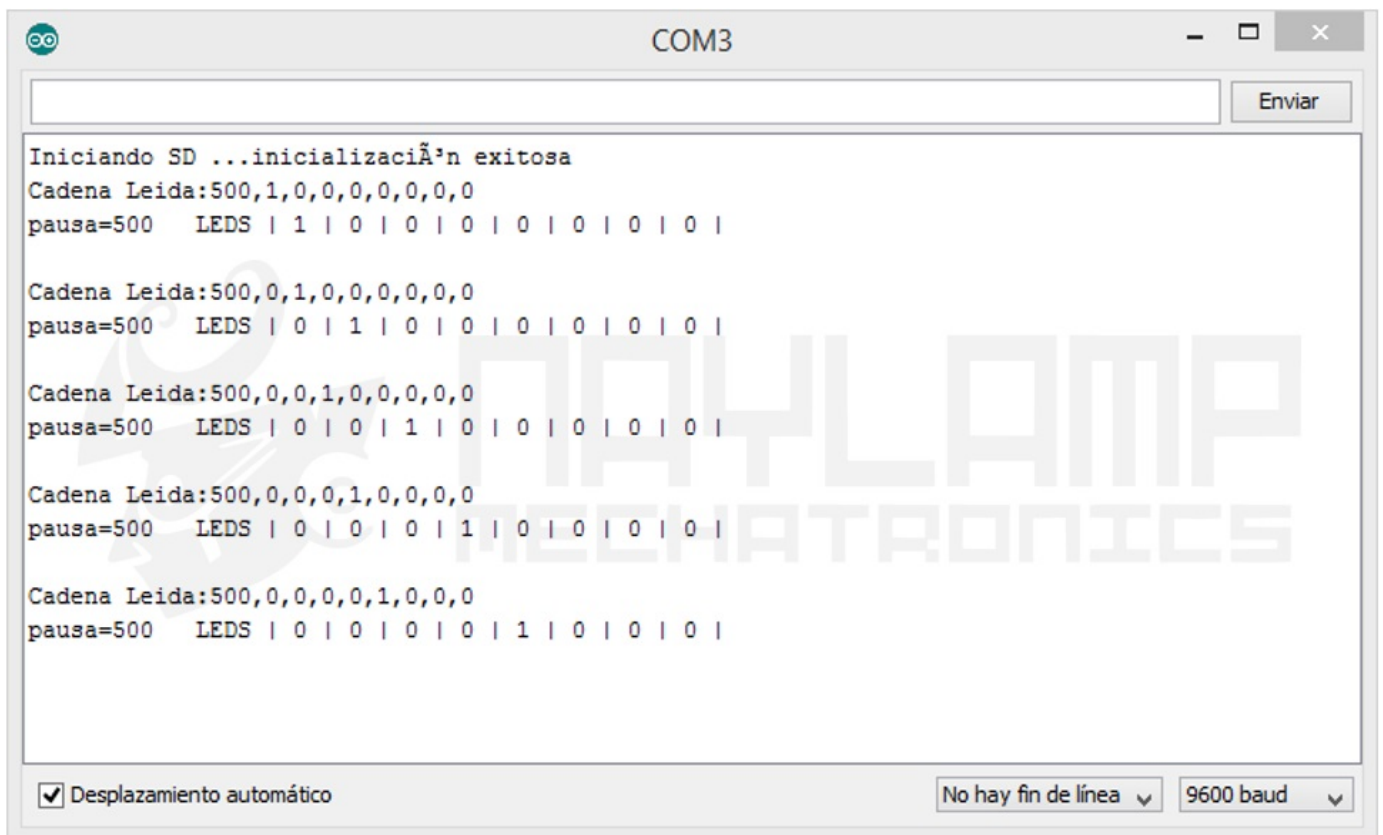
        digitalWrite(PinLeds[i], LOW);
        Serial.print(" 0 |");
    }
}
Serial.println();
Serial.println();

} else {
    Serial.println("Error al abrir el archivo");
}
delay(pausa);
}

```

Como se observa Leemos una línea de la SD la procesamos y mostramos la secuencia en los leds durante el tiempo que también obtenemos desde la SD, es necesario después de cada lectura guardar la última posición de memoria para que en la segunda lectura continuemos la lectura desde esa posición, esto porque después de cada lectura cerramos el archivo.

A continuación se muestra la salida del monitor serial, que es un reflejo de lo que se muestra en los leds:



## Ej3. Guardando datos en la SD (Datalogger)

En este ejemplo guardaremos información de variables y sensores en nuestra SD

Para simular los sensores utilizamos 3 potenciómetros que deben estar conectados a los pines A0, A1 y A2 del Arduino.

El sketch es el siguiente:

```
#include <SD.h>

File myFile;
void setup()
{
  Serial.begin(9600);
  Serial.print("Iniciando SD ...");
  if (!SD.begin(4)) {
    Serial.println("No se pudo inicializar");
    return;
  }
  Serial.println("inicializacion exitosa");
}

void loop()
{
  myFile = SD.open("datalog.txt", FILE_WRITE); //abrimos el archivo

  if (myFile) {
    Serial.print("Escribiendo SD: ");
    int sensor1 = analogRead(0);
    int sensor2 = analogRead(1);
    int sensor3 = analogRead(2);
    myFile.print("Tiempo(ms)=");
    myFile.print(millis());
    myFile.print(", sensor1=");
    myFile.print(sensor1);
    myFile.print(", sensor2=");
    myFile.print(sensor2);
    myFile.print(", sensor3=");
    myFile.println(sensor3);

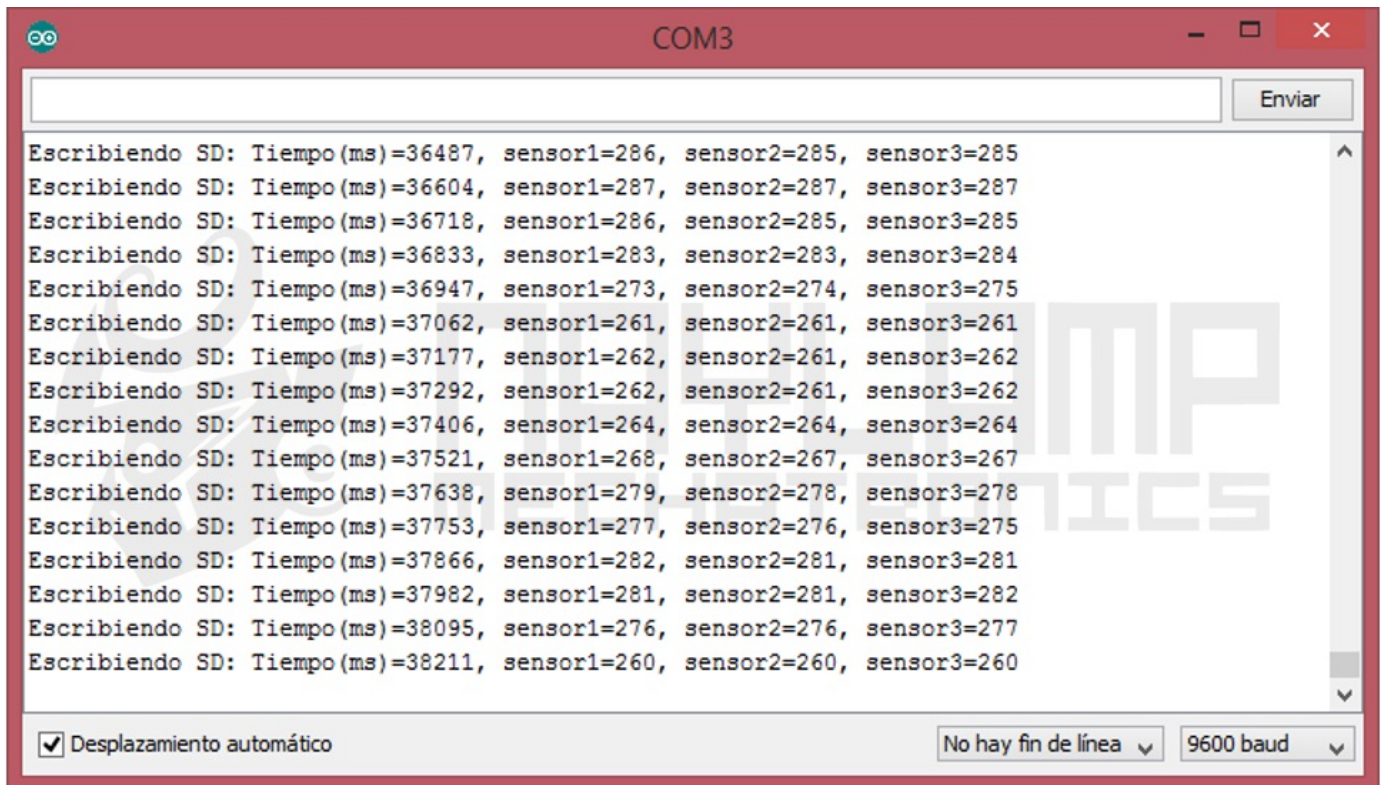
    myFile.close(); //cerramos el archivo

    Serial.print("Tiempo(ms)=");
    Serial.print(millis());
    Serial.print(", sensor1=");
    Serial.print(sensor1);
    Serial.print(", sensor2=");
    Serial.print(sensor2);
    Serial.print(", sensor3=");
    Serial.println(sensor3);

  } else {
    Serial.println("Error al abrir el archivo");
  }
  delay(1000);
}
```

En el monitor serial muestra los datos que se están escribiendo en la SD:





```
COM3

Escribiendo SD: Tiempo(ms)=36487, sensor1=286, sensor2=285, sensor3=285
Escribiendo SD: Tiempo(ms)=36604, sensor1=287, sensor2=287, sensor3=287
Escribiendo SD: Tiempo(ms)=36718, sensor1=286, sensor2=285, sensor3=285
Escribiendo SD: Tiempo(ms)=36833, sensor1=283, sensor2=283, sensor3=284
Escribiendo SD: Tiempo(ms)=36947, sensor1=273, sensor2=274, sensor3=275
Escribiendo SD: Tiempo(ms)=37062, sensor1=261, sensor2=261, sensor3=261
Escribiendo SD: Tiempo(ms)=37177, sensor1=262, sensor2=261, sensor3=262
Escribiendo SD: Tiempo(ms)=37292, sensor1=262, sensor2=261, sensor3=262
Escribiendo SD: Tiempo(ms)=37406, sensor1=264, sensor2=264, sensor3=264
Escribiendo SD: Tiempo(ms)=37521, sensor1=268, sensor2=267, sensor3=267
Escribiendo SD: Tiempo(ms)=37638, sensor1=279, sensor2=278, sensor3=278
Escribiendo SD: Tiempo(ms)=37753, sensor1=277, sensor2=276, sensor3=275
Escribiendo SD: Tiempo(ms)=37866, sensor1=282, sensor2=281, sensor3=281
Escribiendo SD: Tiempo(ms)=37982, sensor1=281, sensor2=281, sensor3=282
Escribiendo SD: Tiempo(ms)=38095, sensor1=276, sensor2=276, sensor3=277
Escribiendo SD: Tiempo(ms)=38211, sensor1=260, sensor2=260, sensor3=260

☒ Desplazamiento automático
No hay fin de línea 9600 baud
```

Y si insertamos la SD en nuestra PC podemos ver el archivo generado:



```
DATALOG.TXT: Bloc de notas

Tiempo(ms)=375, sensor1=450, sensor2=448, sensor3=446
Tiempo(ms)=487, sensor1=357, sensor2=356, sensor3=356
Tiempo(ms)=601, sensor1=343, sensor2=342, sensor3=341
Tiempo(ms)=712, sensor1=328, sensor2=328, sensor3=329
Tiempo(ms)=826, sensor1=302, sensor2=301, sensor3=302
Tiempo(ms)=937, sensor1=296, sensor2=295, sensor3=295
Tiempo(ms)=1051, sensor1=294, sensor2=293, sensor3=293
Tiempo(ms)=1163, sensor1=306, sensor2=306, sensor3=306
Tiempo(ms)=1276, sensor1=290, sensor2=290, sensor3=291
Tiempo(ms)=1388, sensor1=277, sensor2=276, sensor3=277
Tiempo(ms)=1505, sensor1=275, sensor2=274, sensor3=275
Tiempo(ms)=1617, sensor1=277, sensor2=276, sensor3=276
Tiempo(ms)=1730, sensor1=293, sensor2=292, sensor3=291
Tiempo(ms)=1843, sensor1=288, sensor2=288, sensor3=289
Tiempo(ms)=1955, sensor1=269, sensor2=269, sensor3=269
Tiempo(ms)=2068, sensor1=269, sensor2=269, sensor3=269
Tiempo(ms)=2181, sensor1=275, sensor2=275, sensor3=274
```

## Ej4. Exportando datos de nuestro datalogger a Excel.

En este caso los datos lo almacenaremos en forma de tabla pero no podremos guardarlos directamente en un archivo de Excel, sino los guardaremos en formato .csv el cual lo trabajaremos como si fuera una hoja de texto, las columnas lo separaremos por comas (u otro separador) y las filas por saltos de línea.

Para Excel también se podría trabajar en un archivo txt, pero en la extensión csv tenemos la posibilidad de importar no solo a Excel sino a otros programas como Calc de Open Office, Gnumeric, Matlab o sistemas web.

Tomaremos el ejemplo anterior con la diferencia que guardaremos los datos en el formato explicado anteriormente.

El sketch para el Arduino es el siguiente:

```
#include <SD.h>

File myFile;
void setup()
{
  Serial.begin(9600);
  Serial.print("Iniciando SD ...");
  if (!SD.begin(4)) {
    Serial.println("No se pudo inicializar");
    return;
  }
  Serial.println("inicializacion exitosa");

  if(!SD.exists("datalog.csv"))
  {
    myFile = SD.open("datalog.csv", FILE_WRITE);
    if (myFile) {
      Serial.println("Archivo nuevo, Escribiendo encabezado(fila 1)");
      myFile.println("Tiempo(ms),Sensor1,Sensor2,Sensor3");
      myFile.close();
    } else {
      Serial.println("Error creando el archivo datalog.csv");
    }
  }
}

void loop()
{
  myFile = SD.open("datalog.csv", FILE_WRITE);//abrimos el archivo

  if (myFile) {
    Serial.print("Escribiendo SD: ");
    int sensor1 = analogRead(0);
    int sensor2 = analogRead(1);
    int sensor3 = analogRead(2);
    myFile.print(millis());
    myFile.print(",");
    myFile.print(sensor1);
    myFile.print(",");
    myFile.print(sensor2);
    myFile.print(",");
    myFile.println(sensor3);
  }
```

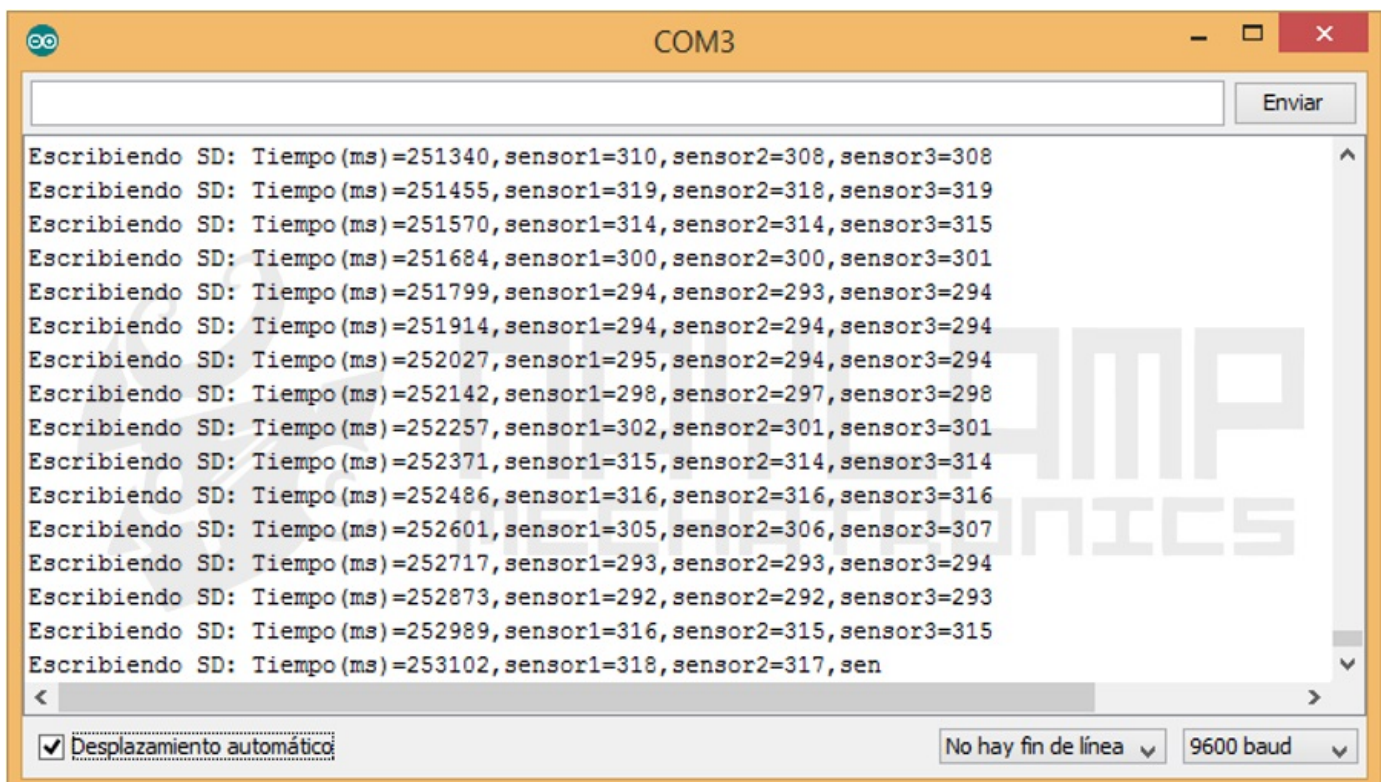
```
myFile.close(); //cerramos el archivo

Serial.print("Tiempo(ms)=");
Serial.print(millis());
Serial.print(",sensor1=");
Serial.print(sensor1);
Serial.print(",sensor2=");
Serial.print(sensor2);
Serial.print(",sensor3=");
Serial.println(sensor3);

} else {
    // if the file didn't open, print an error:
    Serial.println("Error al abrir el archivo");
}
delay(100);
}
```

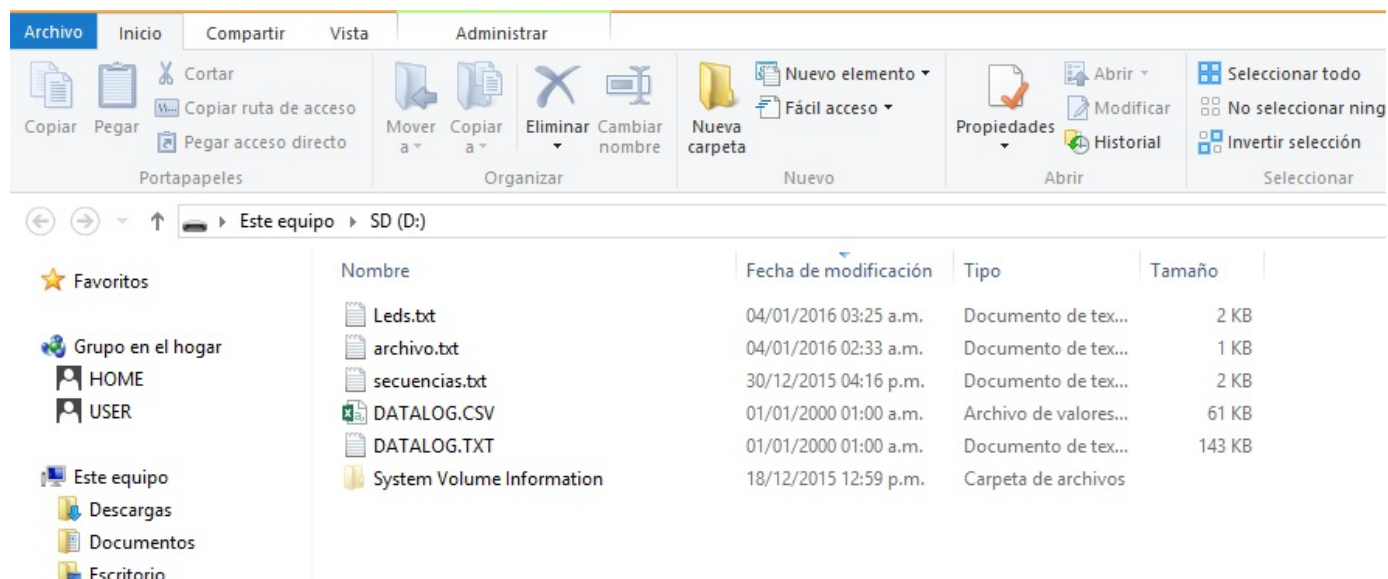
El programa es similar al del ejemplo anterior con la diferencia que al inicio del archivo necesitamos poner el nombre de los campos o encabezado, esto lo hacemos en setup() si el archivo no existe ,entonces creamos uno y seguidamente escribimos la primera fila, para posteriormente en el void loop() solo dedicarse a escribir los datos, como el archivo ya ha sido creado, cada vez que se reinicie o se encienda el Arduino continuara escribiendo desde la última posición, si desean reanudar todo, deberán manualmente desde la pc eleiminar el archivo o por código en setup() remover el archivo con SD.remove(filename).

La salida del monitor serial es igual al del ejemplo anterior, pues solo se ha cambiado la forma de guardar en la SD:

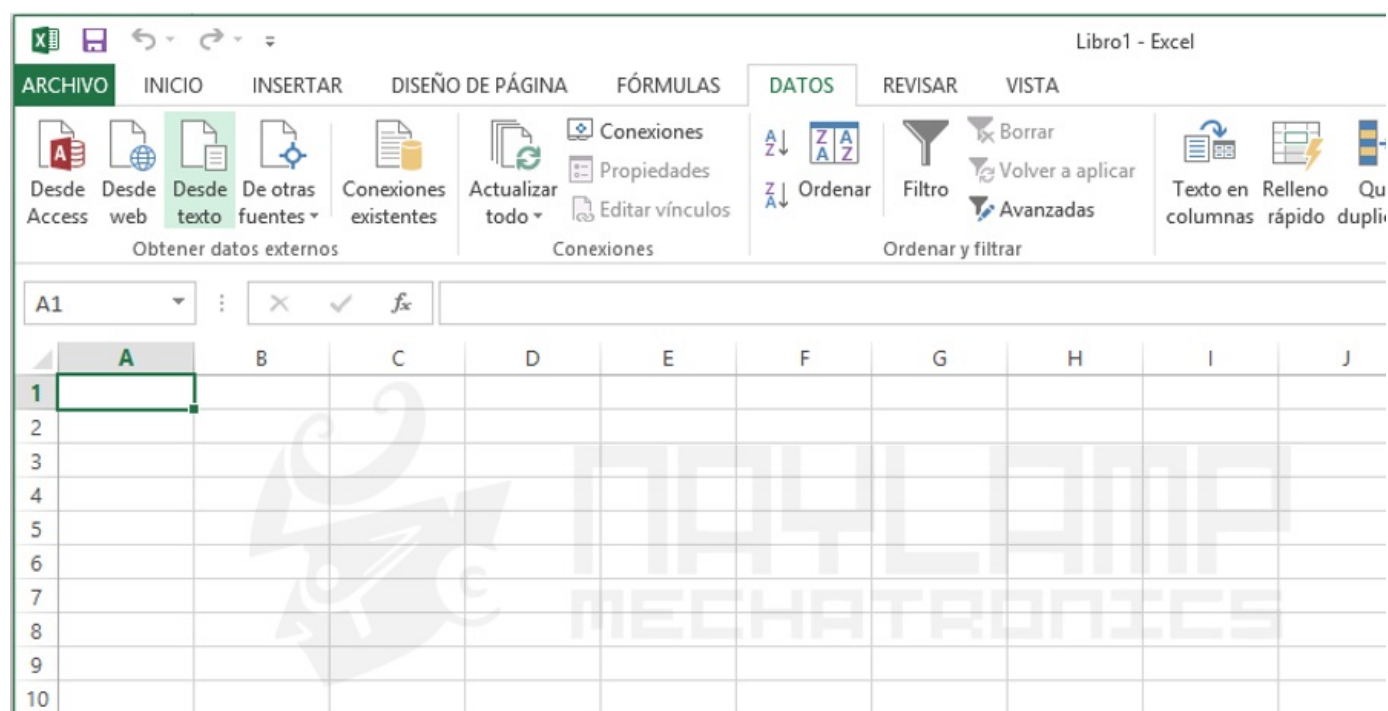




Después de obtener los datos que deseamos, el siguiente paso es ingresar la memoria SD a la PC ya con los datos guardados por nuestro Arduino.



Al ingresar a la SD deben encontrar el archivo DATALOG.CSV, posteriormente abrimos Excel y en la barra de menús vamos a Datos y luego a Obtener datos externos - Desde texto.



Seguidamente escogemos el archivo que deseamos importar, en nuestro caso datalog.csv y nos abrirá el asistente de importación, en el primer paso escogemos archivos delimitados

**Asistente para importar texto - paso 1 de 3**

El asistente estima que sus datos son Delimitados.  
Si esto es correcto, elija Siguiente, o bien elija el tipo de datos que mejor los describa.

**Tipo de los datos originales**

Elija el tipo de archivo que describa los datos con mayor precisión:

☒ **Delimitados** - Caracteres como comas o tabulaciones separan campos.

☐ **De ancho fijo** - Los campos están alineados en columnas con espacios entre uno y otro.

Comenzar a importar en la fila:  Origen del archivo:

☐ Mis datos tienen encabezados.

Vista previa del archivo D:\DATA\LOG.CSV.

1	Tiempo (ms)	Sensor1	Sensor2	Sensor3
2	409	462	459	457
3	521	417	416	416
4	634	394	394	394
5	745	358	357	357

En el paso 2 en separadores marcamos coma:

**Asistente para importar texto - paso 2 de 3**

Esta pantalla le permite establecer los separadores contenidos en los datos. Se puede ver cómo cambia el texto en la vista previa.

**Separadores**

☐ Tabulación

☐ Punto y coma

☒ **Coma**

☐ Espacio

☐ Otro:

☐ Considerar separadores consecutivos como uno solo

Calificador de texto:

Vista previa de los datos

Tiempo (ms)	Sensor1	Sensor2	Sensor3
409	462	459	457
521	417	416	416
634	394	394	394
745	358	357	357

En el paso 3 dejamos todas las columnas en general.

**Asistente para importar texto - paso 3 de 3**

Esta pantalla permite seleccionar cada columna y establecer el formato de los datos.

**Formato de los datos en columnas**

☒ General  
☐ Texto  
☐ Fecha: DMA  
☐ No importar columna (saltar)

'General' convierte los valores numéricos en números, los valores de fechas en fechas y todos los demás valores en texto.

[Avanzadas...](#)

**Vista previa de los datos**

General	General	General	General
Tiempo (ms)	Sensor1	Sensor2	Sensor3
409	462	459	457
521	417	416	416
634	394	394	394
745	358	357	357

[Cancelar](#)
[< Atrás](#)
[Siguiente >](#)
[Finalizar](#)

Después de dar en finalizar, deberán escoger el lugar en donde se va iniciar la tabla:

A1

**Importar datos**

Seleccione cómo desea ver estos datos en el libro.

☒ Tabla  
☐ Informe de tabla dinámica  
☐ Gráfico dinámico  
☐ Crear solo conexión

¿Dónde desea situar los datos?

☒ Hoja de cálculo existente:  
 =SAS1  
☐ Hoja de cálculo nueva

☐ Agregar estos datos al Modelo de datos

[Propiedades...](#)
[Aceptar](#)
[Cancelar](#)

[Cancelar](#)
[< Atrás](#)
[Siguiente >](#)
[Finalizar](#)

Finalmente tenemos los datos ya disponibles en Excel

Libro1 - Excel

ARCHIVO INICIO INSERTAR DISEÑO DE PÁGINA FÓRMULAS DATOS REVISAR VISTA Inicio

Obtener datos externos Actualizar todo Conexiones Ordenar y filtrar Ordenar Filtro Texto en columnas Herramientas de datos Relleno rápido Quitar duplicados Validación de datos Esquema

A1 : X ✓ fx Tiempo(ms)

	A	B	C	D	E	F	G	H
1	Tiempo(ms)	Sensor1	Sensor2	Sensor3				
2	409	462	459	457				
3	521	417	416	416				
4	634	394	394	394				
5	745	358	357	357				
6	860	346	346	346				
7	970	349	348	347				
8	1084	355	354	354				
9	1196	326	326	326				
10	1310	323	322	322				
11	1421	324	324	324				
12	1536	334	332	332				
13	1646	323	323	324				
14	1761	311	310	310				
15	1871	312	311	311				
16	1986	315	314	314				
17	2097	327	326	327				
18	2211	310	310	311				
19	2322	306	306	306				
20	2437	309	308	308				
21	2547	328	328	328				
22	2662	319	319	320				

Hoja1

LISTO 100 %

EL procedimiento para es similar para otras versiones de Excel y otros programas de hojas de cálculo.

**Pueden adquirir los materiales usados en este tutorial en nuestra tienda**

Módulo Micro SD Card

Módulo SD Card

Arduino Uno R3

Cable Dupont, hembra a macho 20cm x 20Und

Tags: sd



## 52 Comments



mauro alejandro zapata rozo

sep 27, 2016

hola buenos días por favor tu ayuda, no logro usar este programa sin desconectar el arduino del pc solo me funciona si esta conectado al computador

Reply



Naylamp

sep 30, 2016

Hola Mauro, cual ejemplo estas trabajando? Si funciona conectado a la PC también te debería funcionar con una fuente externa. Puede ser que tu fuente externa tenga ruido o no tenga el voltaje adecuado.

Reply



Jorge

ago 30, 2017

Hola!. Me pasa lo mismo. Tiene que ver con el puerto serial y el monitor serial . El código funciona perfectamente y guarda los datos existosamente en la memoria, pero SOLO si está conectado al puerto USB y en modo monitor serial. El código que usé fue el de Datalogger.Existe forma de hacer lo mismo, pero sin depender del PC?. En mi caso, necesito capturar datos de humedad en un campo,pero en ese lugar no puedo enchufar un PC y abrir el puerto serie XD.Gracias!.

Reply



Cementerio

oct 12, 2017

Pudieron sulucionar el problema de la conexon a PC? hago un proyecto similar al de Jorge y me gustaria saber que solucion han enncntrado para el problema. Gracias.

Reply



Luisa Fernanda Sanabria Perez

feb 6, 2018

Hola, tengo el mismo problema cuando alimento el arduino con una fuente externa, trabajo en un proyecto donde tomo medidas de voltajes y corrientes, guardo los datos en la micro SD, sin ningún problema pero únicamente cuando tengo conectado el arduino por el puerto USB. Con una fuente externa solo exporta el DATALOGER,

pero cuando lo reviso no contiene nada de datos. Podrían ayudarnos... Gracias.

[Reply](#)**Jandres**

oct 24, 2016

Eres el amo. Todo me fue de gran ayuda. Muchas gracias por tu aporte. Saludos

[Reply](#)**juan**

nov 3, 2016

hola!como puedo implementar un mensaje para cuando una vez se esten guardando informacion en la SD y en algun momento por alguna razon la SD se salio del slot entonces me diga que no hay SD insertada=

[Reply](#)**Naylamp**

nov 6, 2016

Estimado Juan, EL modulo SD o micro SD solo es una adaptador de voltaje para la tarjeta SD, sin la tarjeta, es como si no estuviera conectado nada a arduino y las funciones begin() y open() retornan un false. Si después de retirar se ingresa nuevamente es necesario volver a inicializar la SD.

[Reply](#)**David**

nov 3, 2016

Hola, un buen tutorial, mis felicitaciones. Despues de leerme toda la pagina me surgio una duda, sobre el ejemplo de los 8 led. Partiendo de los 8 led y de un archivo activa.txt y apaga.txt por ejemplo, y en esos archivos hubiera una serie de lienas con dia, mes, hora y minuto, como se haria para que el arduino partiendo de los datos reales de un rtc, leyera ese dia en particular la linea del archivo y activara o apagara los led señalados. No se si me explique bien, para ello pongo un ejemplo: activa.txt: linea1: 14121036-> 14/12 10:36 dd/MM hh:mm ..... linea 100: 2211812 suponiendo que hoy fuera 14 de diciembre y fueran las 10:36 como se haria para que leyera esa linea entre las 100 y encendiera todo los led por ejmplo??????? Un saludo y gracias

[Reply](#)**Naylamp**

nov 6, 2016

Estimado David, para realizar eso el procedimiento es similar, tendrías leer línea por línea y luego hacer comparaciones con la fecha y hora del rtc y evaluar si se debe prender o no los focos. Para leer una línea puedes usar la siguiente función en remplazo del file.read():  
String Linea = file.readStringUntil('\n');  
Y si quieres identificar el número de línea solo tendrías que agregar un contador incremental

[Reply](#)**David**

nov 7, 2016

Naylamp, muchas gracias por responder, pero ahora si se me han complicado las dudas. Me has hecho pensar con este ejemplo y con tu nueva respuesta todo el dia de ayer y parte de hoy. Se podria hacer por ejemplo guardar la fecha, hora y dia en una variable (long dia) y luego compararla con las lineas de cada archivo.txt. ???? Te agradeceria que pusieras un ejemplo en caso afirmativo, porque le he dado muuuuuchas vueltas a la cabeza con esta duda. un saludo y gracias por todo.

[Reply](#)**David**

nov 7, 2016

Hola, no tengo un IDE a mano, asique no se si funcionara correctamente, pero mas o menos creo que se entiende la idea, solo que la deje para un solo led.

```
#include //incluimos la libreria para la
Tarjeta SD
#include #include DS1307 rtc(20, 21); // Indica en que pines esta conectado el RTC.
Time t_temp, t; const int chipSelect = 4; int index=0; char claveSD[15]; File myFile; char c; int
LedRojo=9; // Variable que controla o tempo para grabar los dos parametros no cartÃo SD //
Para utilizar o Sd card do LCD altere para 53
void setup() { rtc.halt(false); // Inicia o funcionamiento
do RTC. pinMode(LedRojo, OUTPUT); Serial.begin(9600); //iniciamos el puerto serie a 9600
baudios //Inicializamos la tarjeta SD Serial.println("Iniciando Tarjeta SD..."); // Pin 53 lo
definimos como salida aun que no lo usemos pinMode(53, OUTPUT); // Si hay tarjeta se inicializa if
(!SD.begin(chipSelect)) { Serial.println("La tarjeta no esta o a fallado"); return; }
Serial.println("Tarjeta inicializada."); } void loop() { t = rtc.getTime(); // Actualiza las variables que usa
el RTC. Serial.println("t"); if ((t.hour > 00) && (t.hour < 12)) { myFile = SD.open("Activa.txt"); bool
encontrado = false; if (myFile) { do { c = myFile.read(); if (c == 'n' || !myFile.available()) {
claveSD[index] = ""; index = 0; if (t == atol(claveSD)) { // compara la fecha (ddmm) y hora (hhmm)
actuales con las lineas encontrado = true; break; } } else { claveSD[index] = c; index++; } } while
(myFile.available()); myFile.close(); if (encontrado) { Serial.println("encontrado");
digitalWrite(LedRojo, HIGH); delay(1500); } else { Serial.println("No existe");
digitalWrite(LedRojo, LOW); } } }
```

[Reply](#)**Victor Olguin**

nov 13, 2016

Hola buen dia, una duda como puedo leer valores por columna de excel a arduino? se puede o podra? en caso de si como hacerlo?

[Reply](#)**Naylamp**

nov 16, 2016

Hola Víctor, no es posible abrir un archivo Excel en arduino. Lo que puedes es exportar tu tabla de Excel en CSV separado por comas, y desde arduino hacer una rutina para que realice la lectura línea por línea parecido al ejemplo 2.

[Reply](#)

**Enrique**

nov 20, 2016

Hola como estas, muy buen post. Tengo un problema, estoy trabajando en un proyecto donde leo valores de un csv linea por linea y comando salidas del arduino como el ejemplo de los leds, el problema es que solo me lee las primeras 65 lineas del archivo. Hay alguna función que defina la cantidad de lineas que voy a leer del archivo? El programa tiene varias hojas pego aqui la parte de la SD. Gracias por su ayuda.

```
#define GOUP 1
#define DIRECTORIO 2
#define ARCHIVO 3
#define DESCONOCIDO 4

struct _disparo {
    byte caja;
    byte salida;
    unsigned long delay_ms;
    struct _disparo * sig;
```

Reply

**Naylamp**

nov 30, 2016

Hola Enrique, si funciona bien con 65 líneas puede ser que alguna de tus variables se esté desbordando, también asegúrate que tú .csv finalice la línea con n, o prueba con 0x0D o 0x0A. También puede ser que tu csv tenga en alguna línea un carácter NULL y esto cause que abandone el while según la lógica que usas.

Reply

**uriel**

dic 3, 2016

Un tutorial bien explicado para reproducir sonidos con un modulo de micro porfavooooor!!!!

Reply

**julio**

ene 30, 2017

Buen día, como podría guardar los datos y que me indique la hora y fecha de cada dato guardado

Reply

**Naylamp**

feb 1, 2017

Hola, Julio para tener hora y fecha debes agregar un módulo RTC, como el DS1307 o DS3231, son fácil de implementar, lo explicamos en este tutorial:

[http://www.naylampmechatronics.com/blog/52\\_tutorial-rtc-ds1307-y-eeeprom-at24c32.html](http://www.naylampmechatronics.com/blog/52_tutorial-rtc-ds1307-y-eeeprom-at24c32.html)

Reply

**Matias Gallo**

feb 8, 2017

Hola, muy buen articulo. mi consulta es... como hago para leer solo la ultima linea del archivo de texto? esto es para que ante un corte de energia, el arduino me levante la variable que previamente guarde. ahora lo que estoy haciendo es medio feo, pero funciona, y equivale a leer, borrar, escribir. asi de esta manera del archivo de texto me queda siempre con una sola linea de texto. o alguna otra idea para levantar el ultimo valor guardado en la tarjeta SD cuando el arduino se reinicia? muchas gracias.

[Reply](#)**Naylamp**

feb 17, 2017

Hola Matías, Si la variable es de tamaño fijo podrías inicialmente leer el tamaño del archivo, luego restar el tamaño de la variable y realizar la lectura a partir de dicha posición. Pero la forma que lo has solucionado también es correcto, si no borras el archivo crecerá indeterminadamente y será más pesado. No olvides también que puedes usar la eeprom interna del arduino.

[Reply](#)**Hector Inbacuan**

abr 29, 2017

Muy buen tutorial amigo. Felicitaciones. Mi pregunta es si es posible en caso de que exista mas de un archivo en la sd, ver los nombres de estos

[Reply](#)**Naylamp**

abr 30, 2017

Hola Hector, con `file.name()` puedes consultar el nombre del archivo o carpeta, y con `openNextFile()` puedes abrir otro archivo del mismo directorio, con estas dos funciones podrías hacer un barrido por toda la SD. Puedes encontrar un ejemplo en Archivo>Ejemplos>SD>ListFiles

[Reply](#)**manuel**

may 26, 2017

hola buenos días quisiera saber si podrias ayudar con la tarjeta sd, quiero conectar dos sensores dth22 y que la tarjeta guarde los datos me podrias ayudar por favorgracias

[Reply](#)**Alex**

jun 6, 2017

Que tal amigo mira disculpa tengo una inquietud. Almaceno datos de sensores en un archivo de texto, y necesito leer solo las 3 ultimas lineas del archivo, estuve analizando el ejemplo con los leds donde vas leyendo el archivo en forma ascendente por asi decirlo, como podria hacerlo para

que lea desde la ultima linea, yo se que la respuesta esta en tu codigo pero ya he intentado todo y no consigo resultados, de antemano gracias x tu ayuda. Saludos.

[Reply](#)**Naylamp**

jun 23, 2017

Hola Alex, Si el tamaño por línea es fijo seria fácil de implementar. Analízalo como si el archivo es fuese una cadena de texto. Con `myFile.size()` obtienes el tamaño y con `myFile.seek(x)` te ubicas en una posición para proceder a leer. Dicha posición seria el tamaño total menos el tamaño de las ultimas 3 líneas.

[Reply](#)**Daniel**

jun 26, 2017

Hola, disculpa, soy muy nuevo en esto, pero quisiera realizar la lectura para encender los led's sin embargo el código no me compila, apareciendo un `#include expects` y eso, solo agarré el sketch como tal

[Reply](#)**carlos**

jun 30, 2017

hola inge. tengo una pregunta estoy haciendo un proyecto con un teclado matricial 4 x 4 y un arduino nano. la idea es que quiero almacenar en una micro sd los datos ingresados atravez de mi teclado matricial al precionar la tecla A. por ejemplo. si presiono el numero 1234 y despues la tecla A, quiero que ese numero 1234 me quede almacenado en mi memoria micro sd y asi sucesivamente con todos mis registros.  
hasta el momento lo unico q he e logrado es que me reconozca las filas y columnas . pero no se como lograr almacenar las pulsaciones. adjunto mi codigo y si ud lo puede modificar para q me almacene en la micro sd le agradeceria.

```
#include  
// empieza el codigo necesario
```

[Reply](#)**jro1952**

jul 5, 2017

Hola, excelente la información brindada. Estoy en una pequeña tarea donde debo leer datos de un GPS y una IMU (serial2 y serial1) presentarlos en el monitor y luego almacenarlos en microSD. Tienes algún material de lectura que me puedas recomendar o algun ejemplo realizado. desde ya muchas gracias.

[Reply](#)**soldado**

ago 5, 2017

hello ingeniero!!tengo una duda. se podria guardar un solo dato alli en los ejemplos o definir los datos que se van a ingresar al txt o al excel???pues hay cada segundo menos va guardando los

datos en la sd.se podria hacer un programa que guardara la informacion en la sd solo cuando ocurra un cambio y nada mas?? por ejemplo si un led se prende y el otro se apaga solo guardar la informacion de cuando se prendió o se apago pero un solo dato.asi como el ejemplo de los sensores guardaría muchos datos en el txt o excel de la memoria creo que los guarda es cada segundo o menos los datos. pues la pregunta era esa!! si hay alguna instrucción que defina la cantidad de datos que se guardan o que se guarde un dato cada minuto u hora o cuando detecte algún cambio!!

[Reply](#)**Naylamp**

ago 29, 2017

Hola, la cantidad de datos y el tiempo o momento en que se guarden va a depender de tu algoritmo, en nuestro caso lo guardamos en cada iteración del bucle principal. Puedes agregar if para agregar las condiciones que mencionas. O programar una función para guardar en la SD y usarla en el momento que desees.

[Reply](#)**Pablo**

ago 8, 2017

Hola muy buen tutorial y gracias de antemano. Pero tengo una pregunta....Estoy haciendo un proyecto con Arduino y MIDI. Me gustaría saber si se pueden grabar en la SD este tipo de datos. En caso afirmativo, como podría hacerlo? Estoy bloqueado en este punto y ya llevo un tiempo.Muchas gracias por todo y disculpa las molestias.Saludos

[Reply](#)**Naylamp**

ago 29, 2017

Hola Pablo, con la Librería que tenemos, solo podemos guardar texto, al usar `myFile.print(variable)`, la variable es convertida a una cadena de caracteres, igual que en un `Serial.print()`; cualquier otro tipo de dato que la función print no soporte tiene que convertirse a una cadena de texto o String para poder guardar en la SD. Ten en cuenta que en una SD se gestiona por Archivos a diferencia de un eeprom que es por direcciones.

[Reply](#)**Raul**

ago 23, 2017

hola disculpa una pregunta como puedo para guardar datos en la SD, pero para que funcione en diferentes tiempos por ejemplo, para que un led o x aparato funcione lunes, miercoles, viernes un promedio de 5 min

[Reply](#)**Naylamp**

ago 29, 2017

Hola Raul, tienes que agregar un RTC para que tu arduino pueda leer la fecha y hora. Puedes ayudarte de nuestro Tutorial RTC



[Reply](#)**Jorge**

ago 29, 2017

Hola! Excelente tutorial. Es extremadamente útil. Una consulta: Como hago para leer y guardar los datos SIN abrir el monitor serial?. Gracias

[Reply](#)**Julio Cesar**

ago 31, 2017

Hola estoy intentado grabar datos en una shield de esos y me falla mas que mono con escopeta hasta una el 90% de las veces me dice que no tengo tarjeta..... y la tarjeta es de 4 gigas

[Reply](#)**Jorge**

sep 1, 2017

Alguna ayuda para poder guardar datos sin depender de la comunicación serial del pc :/ ?. Si conecto el arduino al usb del computador, funciona perfectamente. Si dejo el sistema con una batería, no guarda datos ya que falta la comunicación serie.gracias

[Reply](#)**soldado**

sep 3, 2017

hellow de nuevo ingeniero!!hay una duda y es estaSCK 13MOSI 11MISO 12que instrucciones hay para cambiar las posiciones del SCK MOSI Y MISO por ejemplo guardarSCK 10MOSI 8MISO 9

[Reply](#)**soldado**

sep 3, 2017

hellow de nuevo ingeniero!!hay una duda y es estaSCK 13MOSI 11MISO 12que instrucciones hay para cambiar las posiciones del SCK MOSI Y MISO por ejemplo guardarSCK 10MOSI 8MISO 9

[Reply](#)**soldado**

sep 3, 2017

hellow de nuevo ingeniero!!hay una duda y es estaSCK 13MOSI 11MISO 12que instrucciones hay para cambiar las posiciones del SCK MOSI Y MISO por ejemplo guardarSCK 10MOSI 8MISO 9

[Reply](#)**Edwin**

sep 27, 2017

hola buenos dias, quisiera saber porque en el ejemplo de escritura de la sd, cierras el archivo y despues vuelves a imprimir, estoy utilizando ese ejemplo y despues de un tiempo me genera error en la escritura de la sd

[Reply](#)**Oswaldo**

oct 11, 2017

Hola ¿Alguien sabe Como hago para que los archivos que creo tomen la fecha y hora de mi modulo RTC? Se guardan con fecha del 1 de enero del 200 las 0 hrs y 0 min Saludos

[Reply](#)**Chris**

ene 22, 2018

Estimado como puedo leer los datos de una conexión Ethernet??? o sea de un PLC Micrologix 1100. saludos

[Reply](#)**angel**

ene 23, 2018

tengo un dilema al abrir moitor serie es exsitosa la coneccion sd pero marca error al abrir el archivo que puedo hacer??

[Reply](#)**Joaquin**

feb 8, 2018

Hola, me podrías decir por favor, ¿por qué me sale que la tarjeta SD no se pudo inicializar?. Sería de gran ayuda que me lo explicaras

[Reply](#)**Alexander**

feb 12, 2018

lograste resolver el problema, estoy teniendo el mismo.

[Reply](#)**carlos**

mar 1, 2018

Estoy trabajando un proyecto de acceso por RFID y quiero que los registros de acceso en tiempo real se guarden en la micro sd alguien que me pueda ayudar :(

[Reply](#)**Byron hurtado**

mar 18, 2018

Hola amigo quiero hacerme una bateria electrica pero te soy sincero tengo años que no he programado es mas ya me olvidado y queria hacerla con una tarjeta de memoria para no usarla con la computadora me puedes ayudar con una opcion de algun link donde este ya los codigos de programacion o si tu lo tienes me puedes ayudar con eso gracias de antemano y bendiciones

[Reply](#)**Caled**

abr 8, 2018

Hola amigo tendrás un ejemplo u orientarme para obtener datos de una SD como por ejemplo NOMBRE: Joss y TELÉFONO: 55443321 e imprimirlos por monitor serial de dos en dos ya que en el otro ejemplo me arroja todos los datos que tengo. Gracias.

[Reply](#)

## Leave a Reply

\* **Name:**

\* **E-mail:**

(Not Published)

**Website:**

(Site url with http://)

\* **Comment:**

## Boletín


Introduzca su dirección de correo electrónico




## Información

[Contacte con nosotros](#)[Entrega](#)[Condiciones de uso](#)[Nosotros](#)

**Tutoriales y Proyectos con Arduino****Mapa del sitio****Mi cuenta****Mis compras****Mis vales descuento****Mis direcciones****Mis datos personales****Mis vales****Información sobre la tienda**

 Naylamp Mechatronics SAC., Almacén: Mz. E Lote 31 Urb. Ingeniería, Trujillo - Perú //// Horario de atención:  
Lun. a Sab. de 9am. a 7pm.

 Llámanos ahora: 997646858

 Email: [naylamp.mechatronics@gmail.com](mailto:naylamp.mechatronics@gmail.com)