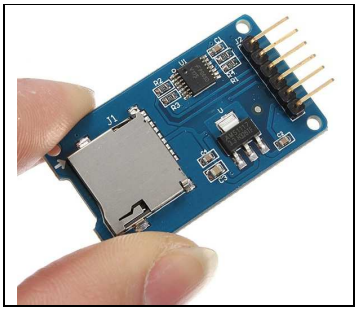
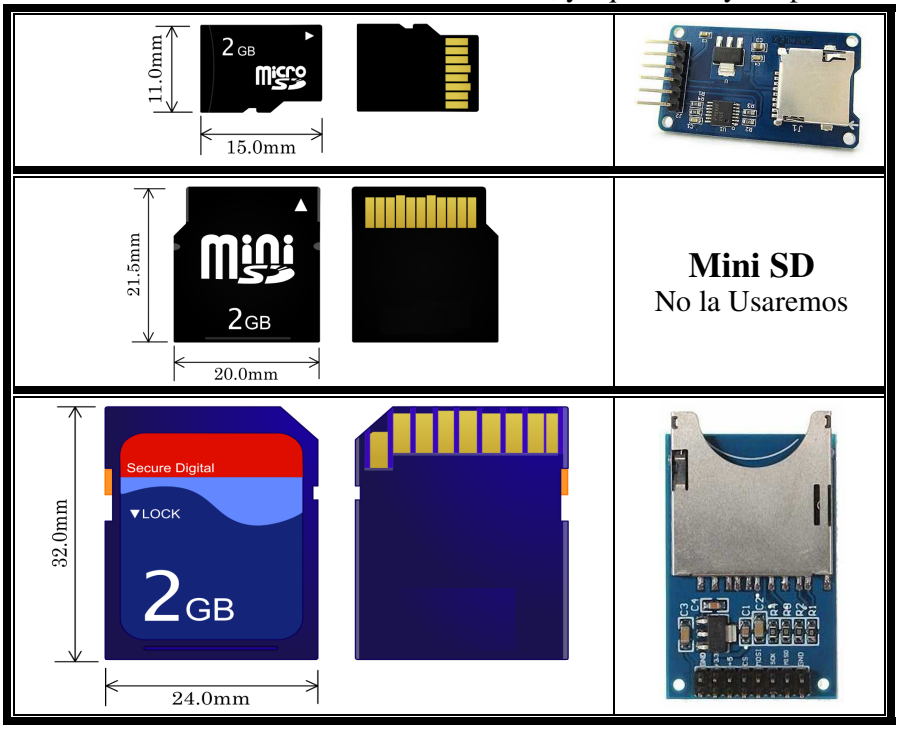
Lector / Grabador de Tarjetas SD y Micro SD



Los Módulos de memorias SD, son dispositivos periféricos que nos permiten almacenar y recuperar (leer y escribir) gran cantidad de datos en Tarjetas SD.

Actualmente son utilizadas en cámaras de fotos, Teléfonos digitales y Memorias USBs, etc. La tecnología FLASH ha tenido un éxito rotundo y una aceptación masiva, lo que ha generado una gran baja en los precios y prácticamente han acabado con CDs, DVDs y demás formatos ópticos para la transferencia de información. Esta Maravillosa y muy barata tecnología también beneficia a los amantes de Arduino, ya que es muy simple su implementación.



Podremos utilizar una Lectora Grabadora de tarjetas SD o Micro SD, cuando necesitamos una Memoria permanente y/o No Volátil, o la memoria que tiene incorporada la tarjeta Arduino que estamos usando, no es suficiente para almacenar los datos que generan aplicaciones como estaciones meteorológicas, datos de usuarios, claves de acceso, registro de clientes, automatización, reproductores de MP3, procesamiento digital de señales, etc. Incluso la ruta de robots precalculadas y como Broche de oro, podremos escribir y leerla en la PC.
Estos dispositivos, tienen generalmente un regulador de voltaje para poder trabajar con niveles de 3.3V y 5V. Con una Velocidad de hasta 10 Mbytes de Lectura/Escritura, usando cuatro líneas de datos. Actualmente tienen incorporado un Reloj de 0 – 25 MHz. Y un muy bajo consumo, que no supera los requerimientos recomendados par Arduino.

En Nuestros proyectos usaremos el Módulo Lector/grabador Sd Card - Pic Aa114

Conexión Lector/Grabador de Tarjetas SD y Micro SD a nuestro Arduino

(Según modelo que esté usando)

La conexión de una Lecto/grabadora de Tarjetas SD o Micro SD, es muy simple, solo hay que buscar los pines adecuados en nuestra placa Arduino.

IMPORTANTE: En la Imagen se muestra que se debe conectar a 5V, sin embargo *Se puede alimentar con 5V o 3,3V* ***usando los pines correspondientes****, pero no se debe alimentar por ambos pines a la vez, o quemará el Lector/Grabador y posiblemente también Arduino (no creo que quiera saberlo). Otra cosa a tener en cuenta, es que* El Pin "**CS**" representa a cualquier Pin digital que podamos usar para controlar, En particular, la imagen hace referencia al PIN Digital 9, y en el programa Usaremos para trabajar el PIN digital 4 o 7.

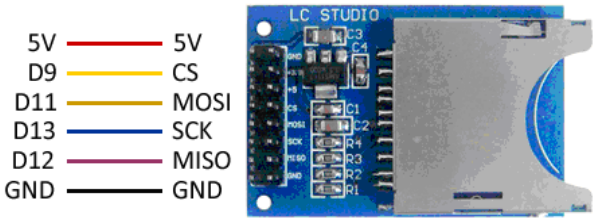
Por conexiones con algún modelo Arduino, que no encuentres listado en la presente guía, puedes consultar el PinOut específico de su placa, en nuestra la página.

Conectando a una Placa Modelo**:** **Uno, Nano y Mini Pro**

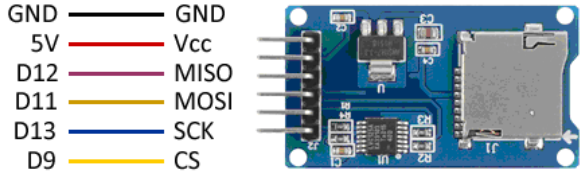
- CS - Pin 9
- MOSI - Pin 11
- SCK - Pin 13
- MISO - Pin 12
- GND - Pin GND

Los pines SPI, donde conectaras con los modelos Arduino Uno, Nano y Mini Pro. Son los siguientes:

(D9 hace referencia al pin 9 y D11 Hace referencia al Pin 11 y así sucesivamente)



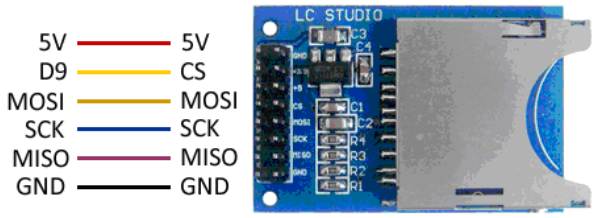
Conexión Tarjeta SD con Modelos Arduino Uno – Nano - Mini Pro



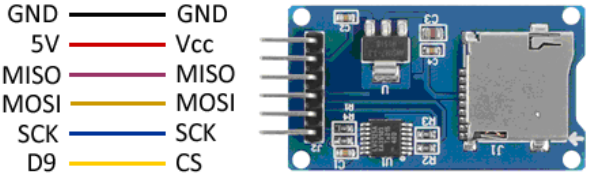
Conexión Micro SD con Modelos Arduino Uno – Nano - Mini Pro

Conectando a una Placa Modelo Mega

La Conexión en la Tarjeta (GND) es muy simple y si alguna duda se presenta solo hay que leer la descripción de cada pin y asunto solucionado. Sin embargo, a la hora de conectar los cables en nuestra placa Arduino Mega, disponemos de al menos, dos grupos de Pines con la misma funcionalidad, en dos lugares distintos, así que, ante la menor duda, debemos recurrir al PinOut de la Placa, donde los encontraremos fácilmente.



Conexión Tarjeta SD con Modelos Arduino MEGA

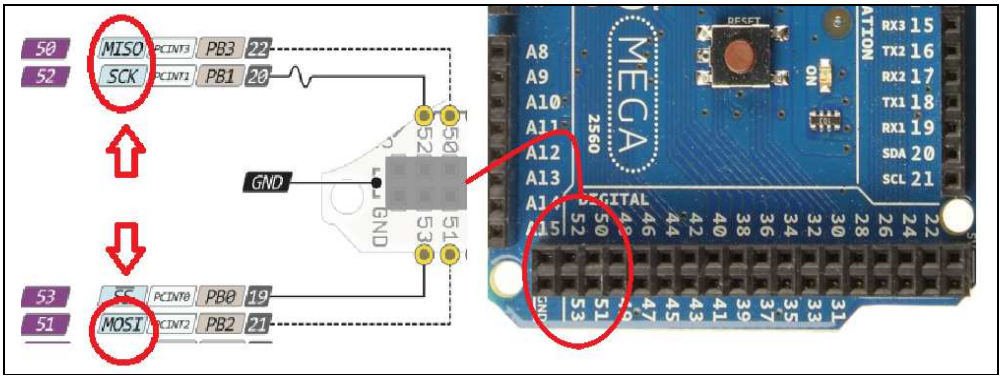


Conexión Micro SD con Modelos Arduino MEGA

IMPORTANTE: Recuerda que "**CS**" representa a cualquier PIN digital que podamos usar, En particular, la imagen hace referencia al PIN Digital 9, y en el programa **Usaremos para trabajar el PIN digital 7**.

El Primer grupo de pines, es fácil de ubicar, si vemos el PinOut de la Placa, encontraremos que:

Pin 50 → MISO
Pin 51 → MOSI
Pin 52 → SCK
Pin GND → GND



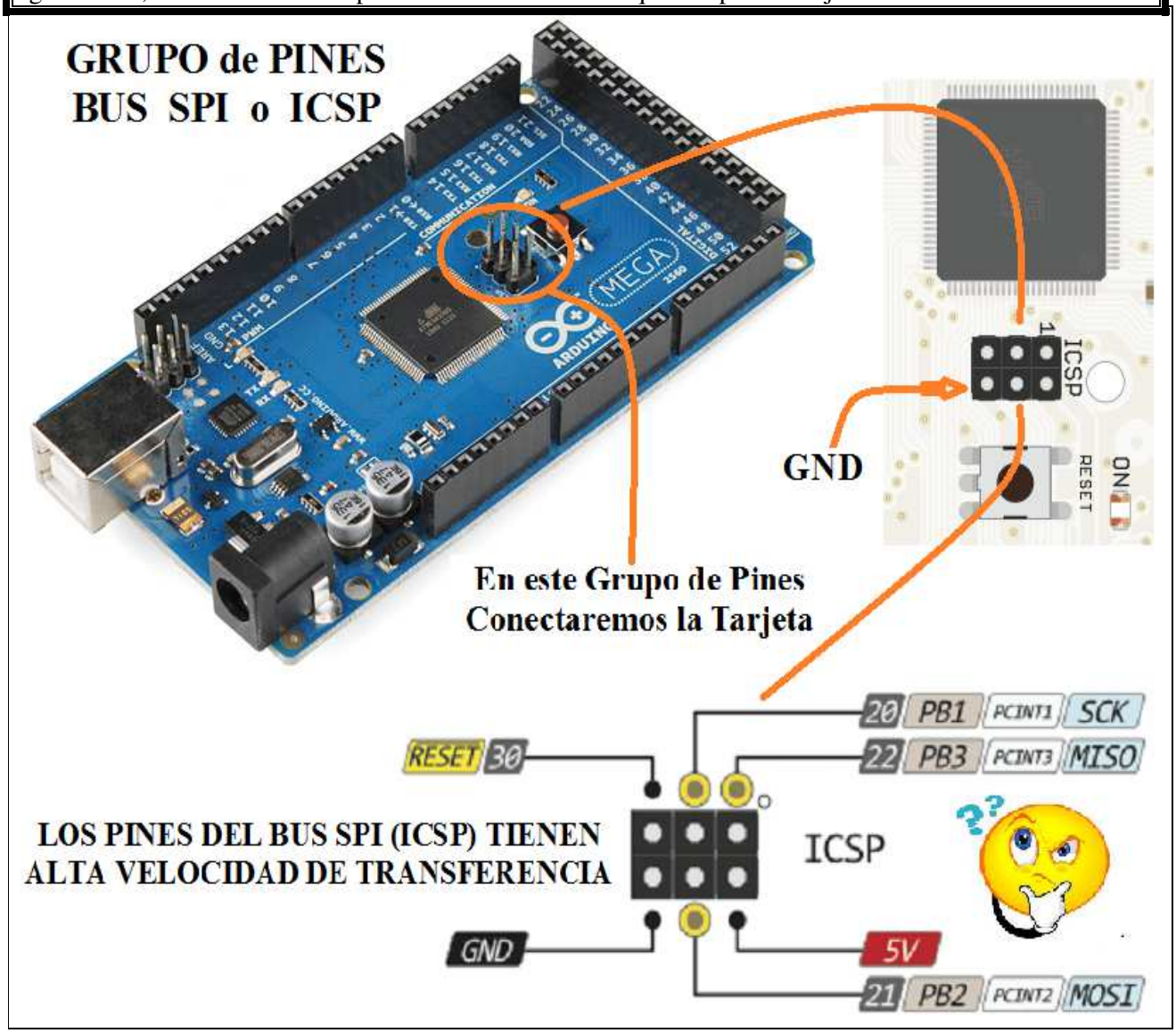
Tenemos un GND y solo nos Falta el Pin de **5V**, pero podemos usar cualquiera de los Pines de 5V de los que Arduino Dispone. Busca, y encontraras más de uno..!!

IMPORTANTE: En la Imagen se muestra que se debe conectar a 5V, sin embargo, podemos *alimentar con 5V o 3,3V* ***usando los pines correspondientes****, pero no se debe de alimentar por ambos pines a la vez, o se quemará el Lector/Grabador y Arduino. Otra cosa a tener en cuenta, es que* El Pin "**CS**"

representa a cualquier Pin digital que podamos usar. En particular, la imagen hace referencia al PIN Digital 9, pero en nuestros programas usaremos para trabajar el PIN digital 4 o 7.

El segundo grupo de Pines se encuentra en el medio de la placa, junto al Botón de "Reset". Veamos entonces como los identificamos y cuales usaremos.

Sería muy bueno que, los trate de identificar usted solo en el PinOut completo de la placa que dispone en nuestra página. Lo ayudará a manejarse independientemente y con mayor fluidez en futuros proyectos. Igualmente, a continuación dispone de la información requerida para trabajar.



Tarjetas SD o Micro SD

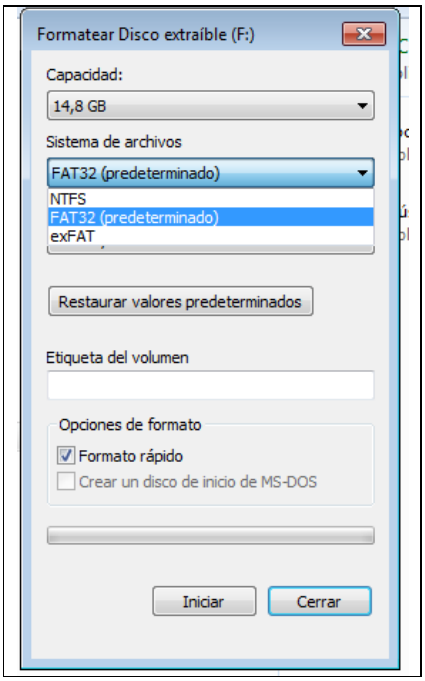
Las tarjetas que podremos usar, son las siguientes: SD o SDSC (Standard Capacity), SDHC (High Capacity), **pero no SDXC (Extended Capacity).** Arduino puede leer y escribir en estas tarjetas, pero no formatearlas.

El Formateo de las tarjetas, debe hacerse desde una PC. No es complicado, **pero es así**. Y deben estar formateadas en sistema de archivos FAT16 (hasta 2GB) o FAT32 (hasta 32GB). De lo contrario No podremos escribir o leer en ellas desde Arduino.
El proceso de Formateo es muy Simple:

- ✓ Para comenzar, pones la Tarjeta SD o micro SD en tu PC, y en cuanto sea reconocida, veras que aparece una nueva unidad, que será reconocida como Disco Extraíble.

- ✓ Inmediatamente después, te dará opciones de lo que puedes hacer con esta nueva unidad extraíble. (hasta acá, todo es igual, a cuando insertas un Pendrive). y si ya esta formateada alejes **abrir carpeta para ver Archivos.** Pero si no puede ser leída, es posible que tarde unos segundos y luego pregunte si quieres formatearla, y acá es donde debes prestar mucha atención.

- ✓ Si la PC no puede leerla, o la acabas de comprar, o desconoces el origen, debes asegurarte que este correctamente formateada con sistema de archivos FAT16 (hasta 2GB) o FAT32 (hasta 32GB). De lo contrario No podremos escribir o leer en ellas desde Arduino.
- ✓ Para esto haces un clic en Disco Extraíble y a continuación el botón derecho del Mouse. De inmediato te aparecerá un menú de opciones entre las cuales esta Formatear.



- ✓ Apenas veas esta ventana de formateo, (presta especial atención, que efectivamente sea la Unidad Extraíble la que estés por formatear, o eliminaras la información de alguna unidad de tu PC) Selecciona el sistema de archivos correcto. Si tu Tarjeta es de hasta 2GB selecciona FAT16. Pero si tu Tarjeta es de hasta 32GB deberás seleccionar FAT32.
- ✓ A continuación los otros parámetros de formateo, no tienen mucha importancia, sin embrago, puedes poner una etiqueta o nombre a tu Tarjeta, y hacer un formateo rápido, o lento y minucioso (esto ultimo es recomendado cundo la tarjeta ya fue usada por otro dispositivo).
- ✓ Ahora solo queda Iniciar el formateo y esperar que termine. y ya esta lista tu tarjeta para ser usada con Arduino

Nombres de los Archivos

En cuanto a los nombres de archivos, estamos limitados al viejo formato de 8+3, es decir que el nombre, solo puede tener hasta 8 caracteres, más una extensión de tres caracteres (nombre más extensión). Y nada de acentos, eÑes, ni espacios en blanco, y por demás esta decir un NO rotundo a los signos raros. Resumiendo, solamente podemos usar letras, mineros y cualquiera de los dos guiones (Medio y Bajo).

Cantidad de Archivos

Puedes leer y escribir datos en todos los archivos que necesites, a condición que cierres el que estabas utilizando antes de abrir el siguiente (Solo uno a la vez). Usar múltiples archivos te permitirá separar la información de diferentes sensores, configuraciones, etc.

Como Ver los Archivos de la Tarjeta en la PC

Es muy simple, solo debe tener una Lector/grabador de Tarjetas en la PC, Son muy Baratas, y cuestan menos que el Lecto/Grabador que usaremos con Arduino. Se inserta la tarjeta en el lector, y lo visualiza igual que lo hace con un CD o un Pendrive. Y si no podía, ahora también podrá ver y/o copiar los archivos que grabo con el celular.

Podemos Usar la Tarjeta del Celular?

La respuesta es Si. Podemos usar la tarjeta de nuestro celular, pero no lo recomiendo, ya que por ahora estamos experimentado, y puedes llegar a perder (borrar) toda la información. Es mejor que uses alguna que ya te resulte pequeña, o compra una pequeña y destínala a jugar, son muy baratas. Los Adaptadores para usar una tarjeta Micro en un Lecto/Grabador de SD, generalmente te los regalan cuando compras una Tarjeta Micro SD.

Precauciones a tener en cuenta

Es importante saber que, existe un Buffer de datos, donde los datos, se almacenan temporalmente antes de grabarlos, evitando perdidas de tiempo, y solo se grabarán en el archivo, cuando el Buffer se llene o cierres el archivo. Es por este motivo, que en ocasiones, no se guarden inmediatamente en el archivo los datos que le envías a la SD. Para asegurarse de que SI están guardados, debes hacer una llamada a **flush** (Función que vacía el Buffer guardando todo) y/o **close** (Función que guarda y cierra). Siempre puedes consultar al final de la guía, el Apéndice **A - Librería "SD.h" Clases y Métodos disponibles**

RECORDATORIO: **La librería** "**SD.h**" está incluida entre otras librerías, cuando instalamos el IDE para desarrollo de Arduino, pero no se carga por defecto, ya que no siempre la usaremos (evitamos usar recursos que se les pueden asignar a otro propósito). Por este motivo, es necesario incluirla (cargarla) antes de poder utilizarla. Escribiremos #include<SD.h> en una de las primeras líneas del código de nuestro Programa, y listo, ya puede ser usada en nuestro programa.

1) Grabar un Archivos Con 10 Valores tomados de un Sensor Ficticio.

Generar un Archivo con 10 lecturas de un censor ficticio. La función que simula leer el sensor solo retorna cero. Cada línea grabada contendrá, el número de línea, el tiempo (expresado en milésimas de segundo), desde que se activo el programa, y el valor del sensor, en nuestro programa será cero. El nombre del archivo es "**datalog.txt**". Puede consultar el "**Apéndice A - Librería "SD.h" Clases y Métodos Disponibles**".

(Programa "700_Lector_Grabador_Sd_01_Escritura")

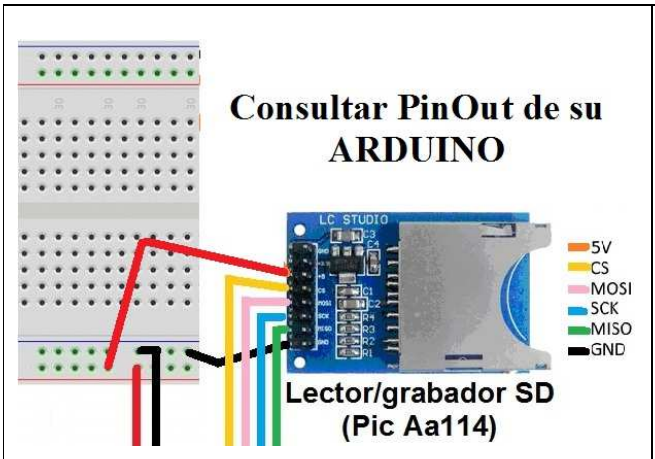
```
1   #include<SD.h> // Esta Librería se encuentra en el IDE de Arduino
2   #define PinLectora_SD   7  // Este es el Pin denominado SC, en el PinOut del Dispositivo
-
3   File ArchivoEjemplo; // Variable del tipo Archivo
4   long C;                    // Usada para Contar Numero de Líneas que Graba en Archivo
5   int  ValorSensado;      // Valor recibido de un sensor Ficticio
-
6   void setup( ){
7     Serial.begin(9600); // Abro el Puerto Serie
8     C = 0;                   // Inicializo Numero de Línea
9   /******** Controlando que inicie bien la Lecto/Grabadora de Tarjetas ********/
10    Serial.print( F("Iniciando SD ...") );
11    if (!SD.begin( PinLectora_SD )){
12      Serial.println( F("Error al iniciar") );
13      return;
14    }
15    Serial.println(F("Iniciado correctamente"));
16  /*************************************************************/
17  }
```

Programo las Acciones que se deben hacer por única vez en la Función **Setup( )**.

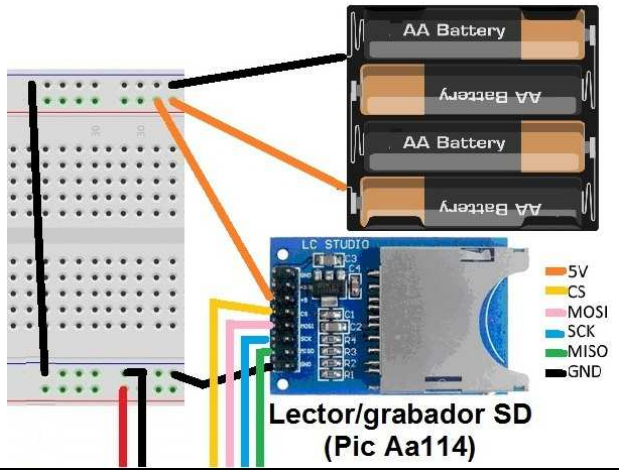
```
-
int LeeSensor( ){ // Función que simula la lectura de un sensor
  return( 0 );
}
-
void loop( ){
  if(C < 10){   // Controla la Cantidad de líneas que se grabarán
    ArchivoEjemplo = SD.open("datalog.txt", FILE_WRITE); // Abriendo para Escribir
    if (ArchivoEjemplo != NULL) { // Si NO pudo abrirse, la Variable Tiene Valor NULL
      C++;                // Cuento la Línea que Estoy Grabando
      ValorSensado = LeeSensor( ); // Leo Sensor Ficticio
      ArchivoEjemplo.print(C);   // Grabo Número de línea
      ArchivoEjemplo.print(" - Tiempo(ms)= ");
      ArchivoEjemplo.print(millis( ));
      ArchivoEjemplo.print(" - Valor Sensor= ");
      ArchivoEjemplo.println( ValorSensado );
      ArchivoEjemplo.close( ); // Cerrando archivo
    }else{
      Serial.println("Error al abrir el archivo");
    }
  }else{
    Serial.println("Listo"); // En este punto avisa que, ya se grabaron 15 Líneas
  }                          // en el Archivo "datalog.txt"
  delay(500); // Debería eliminarse. Esta solo para que funciones más lento
} // Llave que Finaliza la función loop
```

CIRCUITO PARA NUESTRO PROYECTO

Lista de Materiales: 1 Porta Pilas para 4 pilas comunes AA (1,5V) - 1 Módulo Lector/grabador De Sd Card - Pic Aa114 – 8 a 11 Cables de conexión (dependiendo del modelo de arduino serán Hembra/Hembra o Macho/Hembra) - 1 Tarjeta Micro SD con adaptador (Máx. 16Gb) - 1 Protoboard - Placa Arduino y 1 Cable USB.



Dispositivo Alimentado desde Placa Arduino



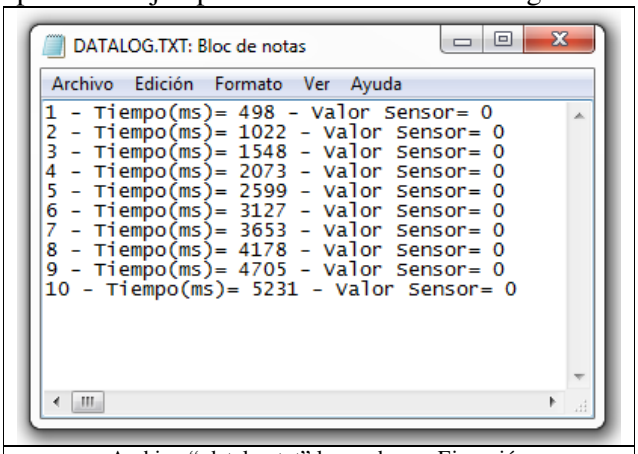
Dispositivo con Alimentación externa

-o-

<u>Los cables deberán ser conectados</u> (abajo) de izquierda a derecha, deberán ser conectado a la placa según corresponda al Modelo Arduino que uses:

Cables	Rojo	Negro	Amarillo	Rosa	Celeste	Verde
Lecto Grabadora SD	**5V**	**GND**	**CS**	**MOSI**	**SCK**	**MISO**
UNO - Nano y Mini Pro	**5V**	**GND**	**Pin 9**	**Pin 11**	**Pin 13**	**Pin 12**
Mega	**5V**	**GND**	**Pin 7**	**Pin 51**	**Pin 52**	**Pin 50**

Si opta por alimentación Externa, recuerde siempre unir el negativo de la alimentación externa con GND de Arduino. Para cualquiera de los dos casos, el programa es el mismo, ya que el control siempre lo tiene Arduino. Una ves que hayas terminado con este ejemplo, no desarmes el dispositivo, ya que los próximos ejemplos usaran la misma configuración.



Archivo " datalog.txt" luego de una Ejecución

Habiendo ejecutado el programa una vez, retira la Tarjeta del dispositivo y visualiza en al computadora, usa para abrir el archivo el Bloc de Notas, y a continuación deberías ver lo que se muestra en la Imagen.

Si pones la Tarjeta otra vez en el dispositivo y ejecutas nuevamente el programa, entonces encontraras que todo se repite dos veces. Es decir que nuestro dispositivo abrirá el archivo y continuará escribiendo a continuación de lo último que escribió. Esto es muy practico, ya que nos permitirá grabar muchas cosas, sin importar que se escribiera antes.

ALGO PARA TENER EN CUENTA: Debes tener mucho cuidado al conectar cada cable en su lugar, no los cruces o podrías quemar la Placa Arduino o la Lecto/Grabadora.

Sin embargo cuando alguno de los cables no hace buen contacto, es común encontrar fallas de grabación, que nada tienen que ver con la lógica del programa.

Una practica muy buena y segura (No romperás nada) seria **desconectar** el PIN correspondiente al Cable de 5V y hacer funcionar el dispositivo, vea que pasa, luego de ver que se grabo, conéctalo otra vez.

Otra linda y jugosa prueba, será **desconectar** el cable correspondiente al PIN "**CS**" y hacer funcionar el dispositivo, luego vea en la PC lo que pasó en la tarjeta (Formateando la Tarjeta SD, se soluciona el problema).

Tienes para divertirse un buen rato. Es muy bueno y práctico, reconocer que genera cada error, de forma tal que, cuando suceda realmente, saber que y donde buscar, pero sobre todo como arreglarlo.

Archivo " datalog.txt" luego de dos Ejecuciones

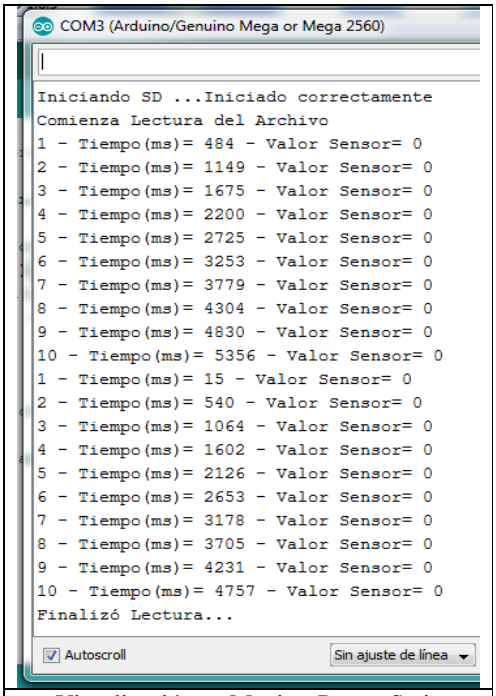
2) Leer el Archivo generado anteriormente y mostrar los datos por el monitor del puerto Serie.

Leer el archivo "**datalog.txt**" que generamos en el programa anterior. Mostrar los datos leídos en la PC, por el Monitor del Puerto Serie.
En este ejercicio, solo se muestra como leer un archivo, y el contenido sale por el monitor del puerto serie, pero podríamos haber leído la configuración de trabajo de algún dispositivo, por ejemplo, un Brazo Robot, un móvil, o una estación meteorológica, etc. Puede consultar el "**Apéndice A - Librería "SD.h" Clases y Métodos Disponibles**".

(Programa "700_Lector_Grabador_Sd_02_Lectura_A")

```
1   #include <SD.h> // Esta Librería se encuentra en el IDE Arduino
-
2   #define PinLectora_SD 7
```

```
-
3   File ArchivoEjemplo;
4   char CaracterLeido;
-
5   void setup( ){
6     Serial.begin(9600);
7     pinMode(PinLectora_SD,INPUT); // Opcional
-
8     while (!Serial) {
9       ; // Esperamos que el puerto serie este abierto.
10      ;  //Este ejemplo es para ver lo que se grabó.
11    }
-
12    Serial.print(F("Iniciando SD ..."));
13    if ( !SD.begin( PinLectora_SD ) ){
14      Serial.println( F("Error al iniciar") );
15      return(0);  // Ver explicación
16    }
-
17  Serial.println( F("Iniciado correctamente") );
-
18   ArchivoEjemplo = SD.open("datalog.txt");
19   if(ArchivoEjemplo != NULL) {
20     Serial.println("Comienza Lectura del Archivo");
21     while (ArchivoEjemplo.available( )){
22       CaracterLeido=ArchivoEjemplo.read( ); // lee 1 Caractér
23       Serial.write(CaracterLeido);
24       //delay(500);  // Hace mas Lento el Programa
25     }
26     ArchivoEjemplo.close( ); //Al terminar, Cierro Archivo
27     Serial.println("Finalizó Lectura...");
28   }else{
29     Serial.println(F("Error al abrir el archivo"));
30   }
31  }
-
32  void loop( ) {
33    // Esta vez acá hacemos NADA
34  }
```



Visualización en Monitor Puerto Serie
(Esto es lo Grabado en Ejercicio Anterior)

Programo Acciones y comandos en la Función **Setup( )** para que se ejecuten solo una vez.

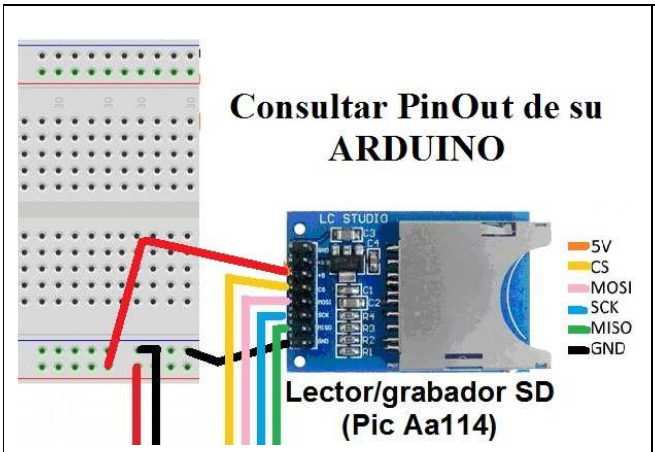
Explicación Líneas de Interés

7	El comando “pinMode( )”, no es obligatorio usarlo en el Pin de control del Archivo. Por otro lado, no sabríamos que poner cuando el archivo lo usemos de Lectura/Escritura
8 a 11	El programa queda esperando (haciendo NADA) que puerto serie este abierto y recibiendo información.. Porque el propósito del ejemplo es ver lo que se grabó, y si no lo vemos… no tendría sentido
12	Comienza Inicialización de la Tarjeta SD
15	**Termina el programa si no se pudo inicializar la Lectora/Grabadora**
17	Termina Inicialización de la Tarjeta SD
18	Apertura del Archivo “datalog.txt”
19	Si el archivo Pudo abrirse….
20	Mensaje por puerto serie
21	Mientras el archivo este accesible, es decir mientras haya algo para leer…

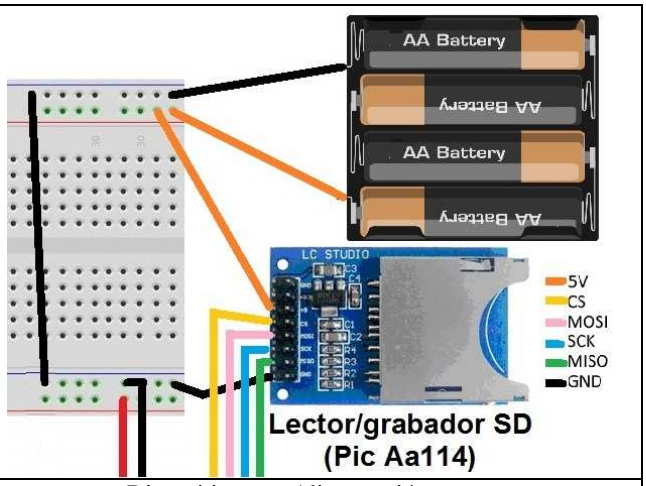
24	El "**delay( )**" esta puesto, Solo para que la lectura sea más lenta, y se visualice por el monitor.

CIRCUITO PARA NUESTRO PROYECTO

Lista de Materiales: 1 Porta Pilas para 4 pilas comunes AA (1,5V) - 1 Módulo Lector/grabador De Sd Card - Pic Aa114 – 8 a 11 Cables de conexión (dependiendo del modelo de arduino serán Hembra/Hembra o Macho/Hembra) - 1 Tarjeta Micro SD con adaptador (Máx. 16Gb) - 1 Protoboard - Placa Arduino y 1 Cable USB.



Dispositivo Alimentado desde Placa Arduino



Dispositivo con Alimentación externa

-o-

Los cables deberán ser conectados (abajo) de izquierda a derecha, deberán ser conectado a la placa según corresponda al Modelo Arduino que uses:

Cables	Rojo	Negro	Amarillo	Rosa	Celeste	Verde
Lecto Grabadora SD	**5V**	**GND**	**CS**	**MOSI**	**SCK**	**MISO**
UNO - Nano y Mini Pro	**5V**	**GND**	**Pin 9**	**Pin 11**	**Pin 13**	**Pin 12**
Mega	**5V**	**GND**	**Pin 7**	**Pin 51**	**Pin 52**	**Pin 50**

Los cables deberán ser conectados: Abajo de izquierda a derecha, Cable Rojo a 5V y Cable Negro a GND, luego los siguientes cuatro cables deberán ser conectado a la placa Arduino según corresponda al Modelo de placa Arduino que usted Tenga. Por lo tanto, en estos casos, deberás releer un poco más arriba, donde se explica la conexión para cada modelo de placa Arduino.

Si opta por alimentación Externa, recuerde siempre unir el negativo de la alimentación externa con GND de Arduino. Para cualquiera de los dos casos, el programa es el mismo, ya que el control siempre lo tiene Arduino. Una ves que hayas terminado con este ejemplo, no desarmes el dispositivo, ya que los próximos ejemplos usaran la misma configuración.

3) Programa que muestra por monitor del Puerto Serie, el estado y características de la Tarjeta. También mostrar información de los archivos contenidos.

Este ejemplo es para quienes le interesa profundizar en medios de almacenamientos. Datos muy importantes desde el comienzo de la informática, y que actualmente el usuario común desconoce desde la aparición de Windows.

En la Imagen se muestra lo que se podrá visualizar por el monitor del puerto serie cuando se ejecute el programa y todo este bien. Le recomiendo realizar pruebas, ejecutando el programa sin la tarjeta, con la tarjeta sin formatear, o desconectando algún cable, etc.

Por otro lado, a modo de prueba, he grabado un archivo en la PC, para destacar la fecha de grabación de un Archivo grabado por arduino y un archivo grabado por la PC. . Puede consultar el "**Apéndice A - Librería "SD.h" Clases y Métodos Disponibles**".

Este programa es una traducción, basada en el programa ejemplo provisto por la pagina oficial de arduino "www.arduino.cc/".

(Programa "700_Lector_Grabador_Sd_03_Informacion_Tarjeta_A")

```
1   #include <SPI.h>
2   #include <SD.h>
-
3   Sd2Card Tarjeta_SD;
4   SdVolume Datos_Tarjeta;
5   SdFile root;
-
6   const int PinLectora_SD = 7;
-
7   void setup( ) {
8    Serial.begin(9600);
-
9    while (!Serial) {
10     ; // Esperando se abra el Puerto Serie
11     ; // Y así podremos ver los mensajes
12     ; // que enviamos. Esto nos permitirá
13     ; // entender cada acción
14   }
-
15   Serial.print("\nInicializando Tarjeta SD...");
-
16    if (!Tarjeta_SD.init(SPI_HALF_SPEED, PinLectora_SD)) {
-
17     Serial.println("Falla de inicialización. Debe Verificar:");
18     Serial.println("* Tarjeta esta Insertada ?");
19     Serial.println("* El Cableado es Correcto?");
20     Serial.println("* Cambió el pin PinLectora_SD para que coincida con su módulo?");
21     while (1);
22   } else {
23     Serial.println("Cableado Correcto y Tarjeta Presente.");
24   }
-
25   Serial.println( );
26   Serial.print("Tipo de Tarjeta:   ");
27   switch (Tarjeta_SD.type( )) {  // Muestra el Tipo de Tarjeta
28     case SD_CARD_TYPE_SD1:
29       Serial.println("SD1");
30       break;
31     case SD_CARD_TYPE_SD2:
32       Serial.println("SD2");
33       break;
34     case SD_CARD_TYPE_SDHC:
35       Serial.println("SDHC (High Capacity)");
36       break;
36     default:
37       Serial.println("Desconocida");
38   }
```

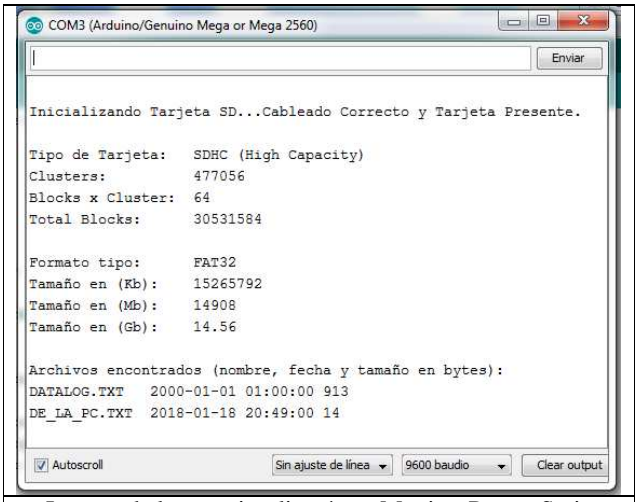


Imagen de lo que visualizará por Monitor Puerto Serie

Programo Acciones y comandos en la Función **Setup( )** para que se ejecuten solo una vez.

```
-
39    if (!Datos_Tarjeta.init(Tarjeta_SD)) {
40     Serial.println("Tarjeta Sin Formato o Desconocido. Formatear con FAT16/FAT32");
41     while (1);
42    }
-
43    Serial.print("Clusters:          ");
44    Serial.println(Datos_Tarjeta.clusterCount( ));
45    Serial.print("Blocks x Cluster:  ");
46    Serial.println(Datos_Tarjeta.blocksPerCluster( ));
-
47    Serial.print("Total Blocks:      ");
48    Serial.println(Datos_Tarjeta.blocksPerCluster( ) * Datos_Tarjeta.clusterCount( ));
49    Serial.println( );
-
50    uint32_t volumesize;
51    Serial.print("Formato tipo:      FAT");
52    Serial.println(Datos_Tarjeta.fatType( ), DEC);
-
53    volumesize = Datos_Tarjeta.blocksPerCluster( );
54    volumesize *= Datos_Tarjeta.clusterCount( );
55    volumesize /= 2;   // Un blocks de la Tarjeta SD contiene 512 bytes (2 blocks son 1KB)
56    Serial.print("Tamaño en (Kb):    ");
57    Serial.println(volumesize);
58    Serial.print("Tamaño en (Mb):    ");
59    volumesize /= 1024;
60    Serial.println(volumesize);
61    Serial.print("Tamaño en (Gb):    ");
62    Serial.println((float)volumesize / 1024.0);
-
63    Serial.println("\nArchivos encontrados (nombre, fecha y tamaño en bytes): ");
64    root.openRoot(Datos_Tarjeta);
-
65    root.ls(LS_R | LS_DATE | LS_SIZE);
66   }
-
67   void loop(void){ // Sin Tareas a Realizar
68   }
```

Explicación Líneas de Interés

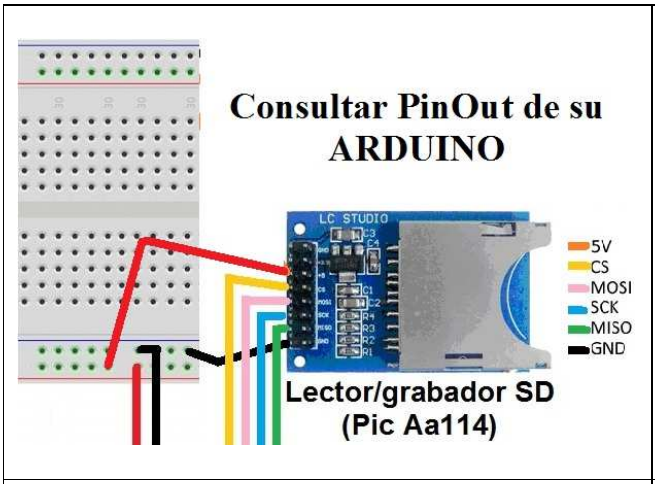
1 y 2	Inclusión de Librerías
3 a 5	Declaración de variables (Instanciando objetos) - usando las clases y funciones de librería SD
9 a 11	Espero que el monitor del puerto serie se habilite
12 a 21	Inicialización de Tarjeta SD y detección de errores
64	Enumera los archivos en tarjeta con fecha y tamaño. **Importante:** la Fecha de archivos grabados el la Lecto/Grabadora por medio de Arduino, Por ahora será 2000-01-01 debido a que Ninguno de los Dispositivos (Arduino o Lecto/grabadora) Dispone de un Reloj Incorporado que aporte esa información.

CIRCUITO PARA NUESTRO PROYECTO

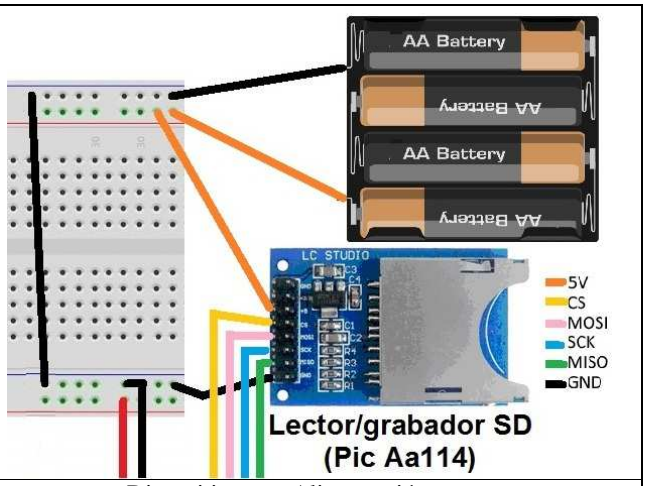
Lista de Materiales: 1 Porta Pilas para 4 pilas comunes AA (1,5V) - 1 Módulo Lector/grabador De Sd Card - Pic Aa114 – 8 a 11 Cables de conexión (dependiendo del modelo de arduino serán

Hembra/Hembra o Macho/Hembra) - 1 Tarjeta Micro SD con adaptador (Máx. 16Gb) - 1 Protoboard - Placa Arduino y 1 Cable USB.

Los cables deberán ser conectados: Abajo de izquierda a derecha, Cable Rojo a 5V y Cable Negro a GND, luego los siguientes cuatro cables deberán ser conectado a la placa Arduino según corresponda al Modelo de placa Arduino que usted Tenga. Por lo tanto, en estos casos, deberás releer un poco más arriba, donde se explica la conexión para cada modelo de placa Arduino.



Dispositivo Alimentado desde Placa Arduino



Dispositivo con Alimentación externa

-o-

Los cables deberán ser conectados (abajo) de izquierda a derecha, deberán ser conectado a la placa según corresponda al Modelo Arduino que uses:

Cables	Rojo	Negro	Amarillo	Rosa	Celeste	Verde
Lecto Grabadora SD	**5V**	**GND**	**CS**	**MOSI**	**SCK**	**MISO**
UNO - Nano y Mini Pro	**5V**	**GND**	**Pin 9**	**Pin 11**	**Pin 13**	**Pin 12**
Mega	**5V**	**GND**	**Pin 7**	**Pin 51**	**Pin 52**	**Pin 50**

Si opta por alimentación Externa, recuerde siempre unir el negativo de la alimentación externa con GND de Arduino. Para cualquiera de los dos casos, el programa es el mismo, ya que el control siempre lo tiene Arduino.

4) Grabar en archivo los valores entregados por 4 Potenciómetros. Usar Alimentación Externa para la Lecto/Grabadora, y Una variable que contenga el Nombre de Archivo.

En este ejercicio guardaremos en un Archivo los valores arrojados por 4 potenciómetros y mientras aprenderemos como poner el nombre del Archivo dentro de una variable, esto es muy importante para futuros proyectos.

Es importante destacar: al grabar el archivo, cada línea debe tener cuatro valores enteros (uno por cada potenciómetro), que termina con una coma (Separador). Y todas las líneas deben terminar con una coma. Este formato será el esperado para leer en el siguiente ejercicio. **Ver formato de cada línea.**

```
468,426,359,309,
0,1161,0,190,
472,440,378,336,
49,46,38,9,
479,447,391,354,
```

Formato de cada línea del Archivo "**datalog.txt**"

Lo importante de este formato de archivo es que puede ser generado y exportado desde Excel (en cualquiera de sus versiones), o viceversa, generado por Arduino y luego importado y analizado en Excel (en cualquiera de sus versiones). ***Y algo que ahora solo mencionare, y más adelante usaremos. Los Archivos de Configuración inicial de dispositivos como Brazos Robot, o rutas que deben seguir los móviles para desplazarse.***

(Programa "700_Lector_Grabador_Sd_04_Valores_4_Potenciometros_Graba")

Puede consultar el "**Apéndice A - Librería "SD.h" Clases y Métodos Disponibles**".

```
1  #include <SPI.h>
2  #include <SD.h>
-
3  const int PinLectoGrabadora = 7;
-
4  String NombreArchivo = "datalog.txt",
5                         DatosString;
6  File    Archivo;
7  int    CantidadPotenciometros = 4,
8         ValorPotenciometro,
9         PinAnalogico,
10        Cantidad = 0;
-
11 void setup( ) {
12   Serial.begin(9600);
13   while (!Serial) {
14     ; //Espera que se establezca conexión con el puerto Serie
15   }
16  Serial.print("Inicializando Lecto/Grabadora SD...");
17   if (!SD.begin(PinLectoGrabadora)) {
18     Serial.println("Falla en Tarjeta SD, o No está Presente");
19     while(1); //Detiene Programa si no puede Inicializar Lecto/Grabadora
20   }
21   Serial.println("Tarjeta Inicializada Correctamente.");
22 }
-
23 void loop( ) {
24 //  if( Cantidad < 5 ){  // Activar para Grabar Solo 5 Líneas
25      DatosString  = "";
26      for(PinAnalogico = 0; PinAnalogico < CantidadPotenciometros; PinAnalogico++) {
27        ValorPotenciometro = analogRead(PinAnalogico);
28        DatosString += String(ValorPotenciometro);
29        if(PinAnalogico < CantidadPotenciometros) {
30           DatosString += ",";
31        }
32      }
-
33      Archivo = SD.open( NombreArchivo, FILE_WRITE);
-
34      if(Archivo != NULL){ // Si Pudo Abrir Archivo
35           Archivo.println(DatosString);
36           Archivo.close( );
36           Serial.println(DatosString);
37      }else{
38         Serial.println("Error Abriendo Archivo " + NombreArchivo);
39      }
40 //     Cantidad++;  // Activar para Grabar Solo 5 Líneas
41 //   }  // Activar para Grabar Solo 5 Líneas
42  }
```

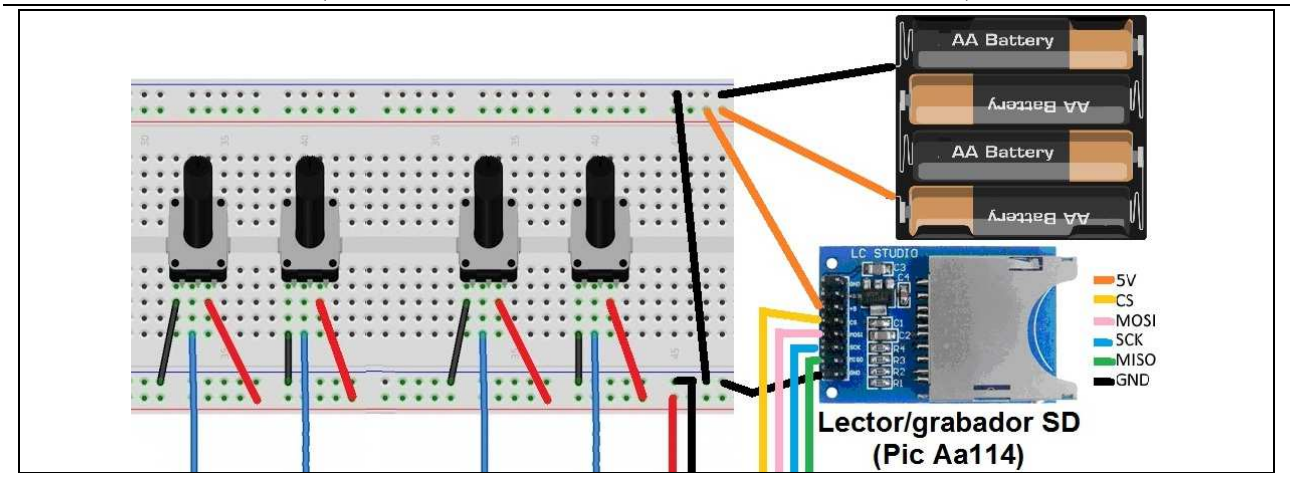
Explicación Líneas de Interés

1 y 2	Inclusión de Librerías
3	Declaración de la Constante "**PinLectoGrabadora**" que identificara el PIN de control de la Lecto/Grabadora

4	Declaración de la Variable "**NombreArchivo**" que contendrá el Nombre del Archivo (Nombre Físico del Archivo) – Es el nombre que veremos cuando examinemos la Tarjeta.
5	Declaración de la variable "**DatosString**" donde pondremos los datos (de un renglón) que grabaremos el archivo.
6	Declaración de la Variable "**Archivo**" que nos permitirá manejar el Archivo (Nombre Lógico del Archivo) - Nombre Lógico es el nombre de la Variable y Nombre Físico lo que estará dentro de la variable y es el nombre que realmente tendrá el archivo.
7	Declaración de la Variable "**CantidadPotenciometros**" que usaremos para saber cuantos potenciómetros estamos usando
8	Declaración de la Variable "**ValorPotenciometro**" que usaremos para guardar cada uno de los valores entregados por los potenciómetros
9	Declaración de la Variable "**PinAnalogico**" que usaremos para posicionar donde este conectado cada uno de los Potenciómetros. A0, A1, A2 y A3.
10	Declaración de la Variable "**Cantidad**" que usaremos para Controlar la Cantidad de lineas que grabaremos en el Archivo. Si Activa las Líneas 24, 40 y 41, que dicen **"// Activar para Grabar Solo 5 Líneas**". Activará el control
12	Abre el Puerto Serie
13 a 15	Espero que realice la conexión con el puerto serie,
16 a 22	Inicializo y controlo estado de la Lecto/grabadora y tarjetas SD
23	Comienzo de la Función "**loop( )**" – Función repetitiva de Arduino
24	Esta es una de las tres líneas que debe activar en el programa para grabar solo 5 líneas en el archivo. Las otras dos líneas son la 40 y 41
25	Inicialización de la Variable "**DatosString**" explicada en la línea 5 – De esta forma, en cada ciclo o vuelta del programa, la variable se limpia y podremos armar el renglón que grabaremos en el archivo.
26	Comienzo ciclo "**for**" – Ciclo repetitivo que nos permitirá recorrer los cuatro Potenciómetros y tomar el valor de cada uno. Este "**for**" contiene en su interior, las líneas 27 a 31 que se explican a continuación.
27 a 31	En este conjunto de líneas se lee el valor del potenciómetro identificado por la variable "**PinAnalogico**" que ira tomando los valores desde cero a tres, correspondientes a los Pines analógicos A0, A1,A2 y A3. Una vez leído el potenciómetro, el valor se almacena en la variable "**ValorPotenciometro**" y a continuación se concatenan en la variable "**DatosString**", separados por una coma. Las comas se insertan en la variable "**DatosString**" en las líneas 29, 30 y 31. **Esta Coma, es la que usaremos para identificar el final de cada numero cuando se lea el archivo**
32	Finaliza ciclo "**for**" explicado en línea 26.
33	Una vez realizada la lectura de todos los potenciómetros y ensamblada la línea, se procede a Abrir el Archivo
34	Se pregunta si el archivo se abrió adecuadamente
35	Si el archivo fue abierto, se graba el renglón con la información
36	Se cierra el archivo
37	Se muestra por el monitor del puerto Serie la información que se grabo en el archivo
40 y 41	Estas son dos de las tres líneas que deberá activar para que el programa solo grabe 5 líneas en el archivo. La otra línea es la 24.

CIRCUITO PARA NUESTRO PROYECTO

Lista de Materiales: 1 Porta Pilas para 4 pilas comunes AA (1,5V) - 4 Potenciómetros – 1 Porta pilas - 1 Módulo Lector/grabador De Sd Card - Pic Aa114 – 15 Cables Macho/Macho – 6 Cables de conexión (dependiendo del modelo de arduino serán Hembra/Hembra o Macho/Hembra) - 1 Tarjeta Micro SD con adaptador (Máx. 16Gb) - 1 Protoboard - Placa Arduino y 1 Cable USB.



Los cables deberán ser conectados: Abajo de izquierda a derecha, **los cuatro cables azules**, deben conectarse a los 4 pines analógicos de control (**A0, A1, A2, A3**) correspondientes a los cuatro potenciómetros. **Cable Rojo** a 5V y **Cable Negro** a GND, luego los siguientes cuatro cables (los últimos 4 de abajo a la derecha) deberán ser conectado a la placa Arduino según corresponda al Modelo de placa Arduino que usted Tenga. Por lo tanto, en estos casos, deberás releer un poco más arriba, donde se explica la conexión para cada modelo de placa Arduino.

Cuando Usamos alimentación Externa, recuerde siempre unir el negativo de la alimentación externa con GND de Arduino.

No desarme este circuito ya que lo usaremos en el próximo ejercicio, cuando leamos el archivo.

5) Leer el archivo generado con los Valores entregados por 4 Potenciómetros (ejercicio anterior) y mostrarlo por el monitor del Puerto Serie. Usar Alimentación Externa para la Lecto/Grabadora, y Una variable que contenga el Nombre de Archivo.

En este ejercicio recuperamos (leemos) de en un Archivo, los valores (Números enteros separados por coma) grabados en el ejemplo anterior. Usamos el nombre del Archivo dentro de una variable, esto es muy importante para futuros proyectos.

Es importante destacar: **al grabar el archivo,** cada línea debe tener cuatro valores enteros (uno por cada potenciómetro), que termina con una coma (Separador). Y todas las líneas deben terminar con una coma. Este formato será el esperado para leer el presente archivo. **Ver formato de cada línea.**

468,426,359,309, 0,1161,0,190, 472,440,378,336, 49,46,38,9, 479,447,391,354,
Formato de cada línea del Archivo "**datalog.txt**"

Lo importante de este formato de archivo es que puede ser generado y exportado desde Excel (en cualquiera de sus versiones), o viceversa, generado por Arduino y luego importado y analizado en Excel (en cualquiera de sus versiones). ***Y algo que ahora solo mencionare, y más adelante usaremos. Los Archivos de Configuración inicial de dispositivos como Brazos Robot, o rutas que deben seguir los móviles para desplazarse.***

(Programa "700_Lector_Grabador_Sd_04_Valores_4_Potenciometros_Lee")

Puede consultar el "**Apéndice A - Librería "SD.h" Clases y Métodos Disponibles**".

```
1  #include <SPI.h>
2  #include <SD.h>
-
3  #define CantidadPotenciometros 4
-
4  const int PinLectoGrabadora = 7;
-
5  String NombreArchivo = "datalog.txt",
6         DatosString;
```

```
7   char    Caracter;
8   File     Archivo;
9   int       Valor[CantidadPotenciometros],
10        C = 0;
-
11  void setup( ) {
12    Serial.begin(9600);
13    while (!Serial) {
14      ; // Esperando se conecte el Puerto serie
15    }
-
16   Serial.print("Inicializando Tarjeta SD...");
-
17    if ( !SD.begin( PinLectoGrabadora ) ) {
18      Serial.println("Falla en Tarjeta o NO Presente");
19      while (1);
20    }
21    Serial.println("Tarjeta Inicializada.");
-
22    File Archivo = SD.open( NombreArchivo );
-
23   if ( Archivo != NULL ) {
24      Serial.println("El Archivo: " + NombreArchivo + "Esta OK..");
25      DatosString  = "";
26      int Mostrar = 0;
27      while(Archivo.available( )) {
28        Caracter = Archivo.read( );
29        if(Caracter >= '0' && Caracter <= '9' || Caracter == ','){
30          if(Caracter != ','){
31            DatosString.concat(Caracter); // Armo caractér a caractér el número
32          }else{
33            Valor[C] = DatosString.toInt( ); // Transformo String a Numero entero
34            C++;
35            DatosString  = "";
36          }
37          Mostrar = 1;
38        }else{
39          if(Mostrar == 1){
40            Serial.print(Valor[0]);
41            Serial.print(" - ");
42            Serial.print(Valor[1]);
43            Serial.print(" - ");
44            Serial.print(Valor[2]);
45            Serial.print(" - ");
46            Serial.println(Valor[3]);
47            C = 0;
48            DatosString  = "";
49            Mostrar = 0;
50          }
51          delay(500);
52        } // Fin del armado, transformación y envió de valores de cada línea del archivo.
53      } // Fin While
54      Archivo.close( );
55    }else{
56      Serial.println("No Se pudo Abrir Archivo:" + NombreArchivo);
```

Programo Acciones y comandos en la Función **Setup( )** para que se ejecuten solo una vez.

```
57      }
58  }
59  void loop( ){
60      // NADA que Hacer
61   }
```

Explicación Líneas de Interés

1 y 2	Inclusión de Librerías
3	Declaro la constante "**CantidadPotenciometros**" que indica la cantidad de potenciómetros, este numero esta relacionada con la cantidad de valores que encontrare en cada linea del archivo, y como siempre cada valor separado con una coma.
4	Declaración de la Constante "**PinLectoGrabadora**" que identificara el PIN de control de la Lecto/Grabadora
5	Declaración de la Variable "**NombreArchivo**" que contendrá el Nombre del Archivo (Nombre Físico del Archivo) – Es el nombre que veremos cuando examinemos la Tarjeta.
6	Declaración de la Variable "**DatosString**", que usaremos para armar cada numero que se leerá caractér a caractér.
7	Declaración de la Variable llamada "**Caracter**" que contendrá el carácter que se lee del archivo. Recordar que el archivo se lee caractér a carácter.
8	Declaración de la Variable "**Archivo**" que nos permitirá manejar el Archivo (Nombre Lógico del Archivo) - Nombre Lógico es el nombre de la Variable y Nombre Físico lo que estará dentro de la variable y es el nombre que realmente tendrá el archivo.
9	Declaración de un vector llamado "**Valor[CantidadPotenciometros]**", de tantos elementos como indica la constante "**CantidadPotenciometros**", que usare para guardar el numero ya convertido, recién leído del archivo.
10	Declaración de la Variable "C", que será usada como índice del vector.
23	En este línea se controla que se haya podido abrir el archivo correctamente, de lo contrario (que no se haya podido abrir el archivo por cualquier motivo) se saltara a la línea 54, donde se mostrara el mensaje de error
24	Envió mensaje por el Puerto Serie, informando que el Archivo "**datalog.txt**" se pudo abrir Apropiadamente
25	Inicialización de la Variable "**DatosString**" la que usaremos para Armar el numero que se leerá carácter a carácter. Se declaro y explico en la línea 6.
26	Declaración e inicialización de la Variable "Mostrar" esta la usaremos y explicare un poco más abajo
27	Comienza la estructura "**While**" que repetirá mientras haya caracteres para leer del archivo "**datalog.txt**". El bloque de acciones que se repetirá es el comprendido entre las líneas 29 y 52
28	Lee un carácter del Archivo y lo guarda en la variable "**Carácter**".
En este segmento del programa, que abarca desde la línea 29 a la 51, es que Arman los valores, se transforman nuevamente en números y se envían por el puerto seria para visualizarlos en la PC	
29	Pregunta si es un Número o una Coma. Si esto es verdad …
30	Si el carácter leído es distinto de una Coma, entonces continua concatenando los caracteres en la variable "**DatosString**".
32	Por el contrario,
33 a 36	si el carácter leído es una coma, es que ya se termino de armar el numero, entonces transformo los caracteres en un numero entero, incremento en uno el índice "**C**" y limpio en contenido de la variable "**DatosString**", dejándola lista para armar el próximo numero si lo hubiera.
37	Habilito en uno la señal "**Mostrar**" que luego, me permitirá mostrar la línea leída, con los valores ya transformados en números.
38	Comienza la Parte falsa del "if" (comenzó en la línea 29), que pregunta si el carácter leído es un Número o una Coma. Entonces, **de lo contrario**
39	Pregunto y si la variable "**Mostrar**" que uso como seña esta habilitada y entonces proceso a

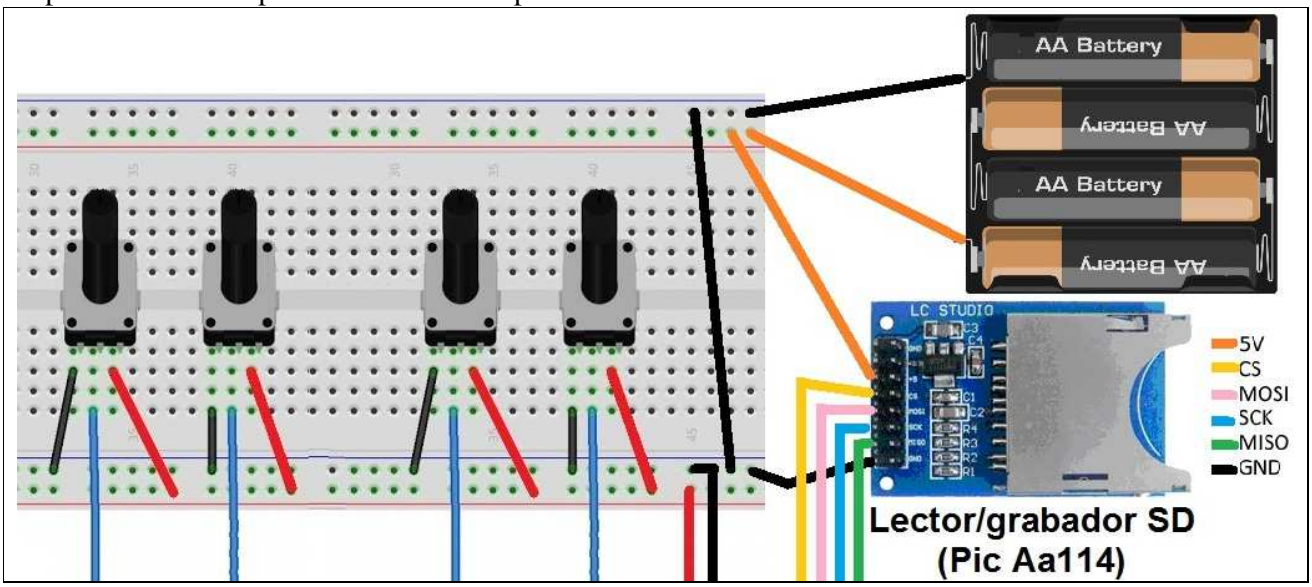
	mostrar la línea leída, con los valores ya transformados en números. Pongo señal y contador en cero, dejando todo listo para precisar próxima línea que se leerá del archivo
40 a 50	Muestro línea leída, con los valores ya transformados en números. Pongo señal y contador en cero, dejando todo listo para precisar próxima línea que se leerá del archivo
54	Si todo Salio bien, cierro el archivo

CIRCUITO PARA NUESTRO PROYECTO

Este ejercicio, es continuación del anterior, solo leeremos los valores recién grabados. Para minimizar el trabajo, se presenta el mismo circuito, aunque no usaremos los 4 potenciómetros. A menos que quiera grabar y leer varias veces y analizar las diferencias.

Lista de Materiales: 1 Porta Pilas para 4 pilas comunes AA (1,5V) - 4 Potenciómetros – 1 Módulo Lector/grabador De Sd Card - Pic Aa114 – 15 Cables Macho/Macho – 6 Cables de conexión (dependiendo del modelo de arduino serán Hembra/Hembra o Macho/Hembra) - 1 Tarjeta Micro SD con adaptador (Máx. 16Gb) - 1 Protoboard - Placa Arduino y 1 Cable USB.

Los cables deberán ser conectados: Abajo de izquierda a derecha, **los cuatro cables azules**, deben conectarse a los 4 pines analógicos de control (**A0, A1, A2, A3**) correspondientes a los cuatro potenciómetros. **Cable Rojo** a 5V y **Cable Negro** a GND, luego los siguientes cuatro cables (los últimos 4 de abajo a la derecha) deberán ser conectado a la placa Arduino según corresponda al Modelo de placa Arduino que usted Tenga. Por lo tanto, en estos casos, deberás releer un poco más arriba, donde se explica la conexión para cada modelo de placa Arduino.



6) Este Programa Verifica La Existencia de un Archivo, luego intenta Crearlo y finalmente intenta Borrarlo. Emitiendo en todos los casos los mensajes que corresponden.

La finalidad de este ejemplo es mostrar la simplicidad con la que podemos trabajar archivos, que luego nos resultara muy útil a la hora de manejar dispositivos complejos.

(Programa "700_Lector_Grabador_Sd_05_Acciones_Con_Archivos_01")

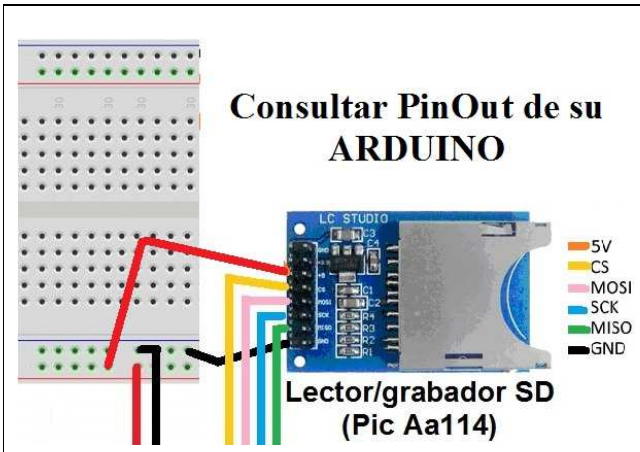
Puede consultar el "**Apéndice A - Librería "SD.h" Clases y Métodos Disponibles**".

```
1  #include <SPI.h>
2  #include <SD.h>
-
3  const int PinLectoGrabadora = 7;
-
4  String NombreArchivo = "Ejemplo.txt";
```

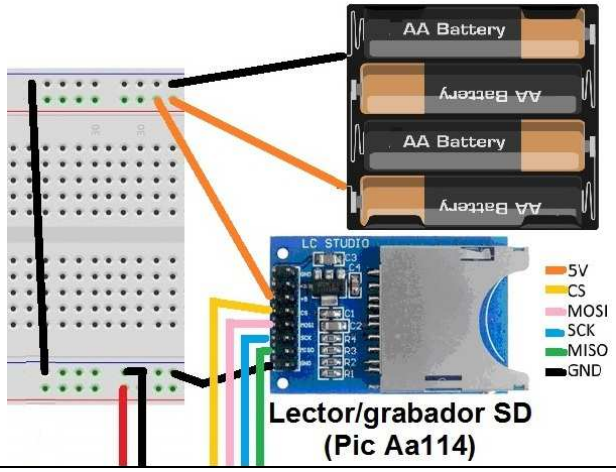
```
-
5   File MiArchivo;
-
6   void setup( ) {
7     Serial.begin(9600);
8     while (!Serial) {
9       ; // Esperando se conecte el Puerto serie
10    }
-
11    Serial.print("Inicializando Tarjeta SD...");
12    if (!SD.begin(PinLectoGrabadora)) {
13      Serial.println("Falla de Inicialización!");
14      while (1);
15    }
16    Serial.println("inicializacion OK!.");
17    Serial.println(" ");
-
18    if ( SD.exists( NombreArchivo ) ){
19      Serial.println("Archivo " + NombreArchivo + " SI EXISTE");
20    } else {
21      Serial.println("Archivo " + NombreArchivo + " NO EXISTE");
22    }
23    Serial.println(" ");
-
24    Serial.println("Creando Archivo " + NombreArchivo);
25    if ( !SD.exists( NombreArchivo ) ){
26      MiArchivo = SD.open( NombreArchivo, FILE_WRITE);
27      MiArchivo.close( );
28      Serial.println("Archivo " + NombreArchivo + " CREADO Exitosamente.!");
29    }else{
30      Serial.println("ERROR. El Archivo " + NombreArchivo + " Ya Existía.!");
31    }
32    Serial.println(" ");
-
33    Serial.println("Borrando Archivo " + NombreArchivo );
34    if ( SD.exists( NombreArchivo ) ) {
35      SD.remove( NombreArchivo );
36      Serial.println(NombreArchivo + " Borrado Exitosamente!.");
37    }else{
38      Serial.println("Error...No se Encontró el Archivo");
39    }
40    Serial.println(" ");
41  }
42  void loop( ) {
43    // Ninguna acción por ahora.
44  }
```

Programo Acciones y comandos en la Función **Setup( )** para que se ejecuten solo una vez.

CIRCUITO PARA NUESTRO PROYECTO



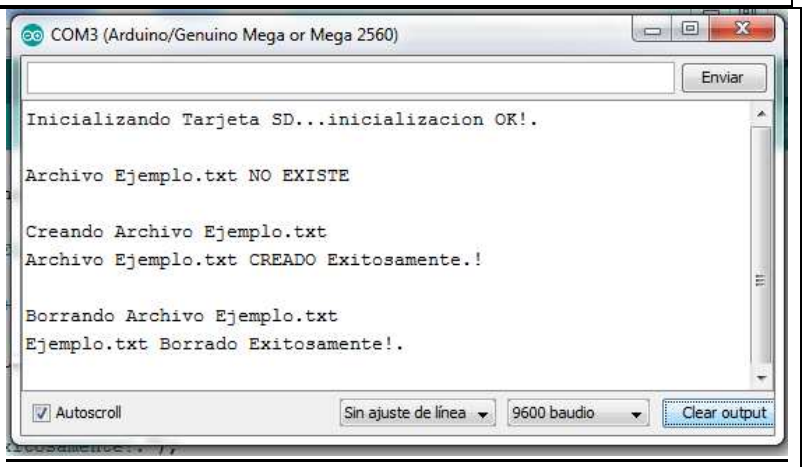
Dispositivo Alimentado desde Placa Arduino



Dispositivo con Alimentación externa

Lista de Materiales: 1 Porta Pilas para 4 pilas comunes AA (1,5V) - 1 Módulo Lector/grabador De Sd Card - Pic Aa114 – 8 a 11 Cables de conexión (dependiendo del modelo de arduino serán Hembra/Hembra o Macho/Hembra) - 1 Tarjeta Micro SD con adaptador (Máx. 16Gb) - 1 Protoboard - Placa Arduino y 1 Cable USB.

Los cables deberán ser conectados: Abajo de izquierda a derecha, Cable Rojo a 5V y Cable Negro a GND, luego los siguientes cuatro cables deberán ser conectado a la placa Arduino según corresponda al Modelo de placa Arduino que usted Tenga. Por lo tanto, en estos casos, deberás releer un poco más arriba, donde se explica la conexión para cada modelo de placa Arduino.



Visualización del Puerto Serie al ejecutar el programa

Si opta por alimentación Externa, recuerde siempre unir el negativo de la alimentación externa con GND de Arduino. Para cualquiera de los dos casos, el programa es el mismo, ya que el control siempre lo tiene Arduino. Una ves que hayas terminado con este ejemplo, no desarmes el dispositivo, ya que los próximos ejemplos usaran la misma configuración.

7) Creación y Uso de Carpetas (Directorios) dentro da la Tarjeta SD. Manejo de Archivos y listado completo de Archivos y carpetas..

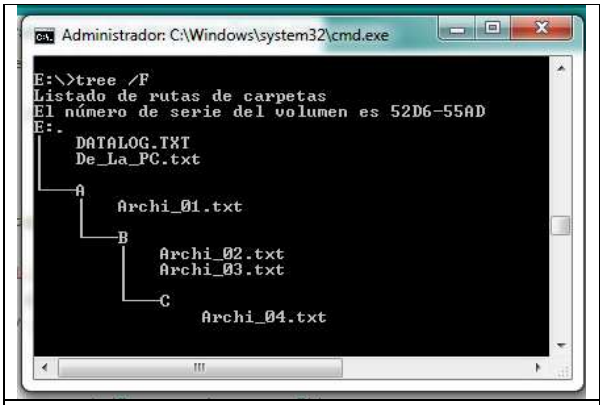
En este Proyecto, será realizado en tres etapas (A, B, C), la primera, creamos el ambiente de trabajo, además de guiarlos en la creación y borrado de Carpetas (Directorios) y profundizamos en la creación de Archivos. En la segunda etapa, jugaremos un poco en el recorrido de la estructura del la Tarjeta. Y finalmente en la tercera nos adentramos en algoritmos recursivos, algo prácticamente indispensable para recorrer completamente nuestra tarjeta. **NOTA:** El circuito es el mismo para los programas de las tres etapas.

(Programa "700_Lector_Grabador_Sd_06_Lista_Directorio_01_Crea")

Etapa A

```
1  #include <SPI.h>
2  #include <SD.h>
-
3  const int PinLectoGrabadora = 7;
```

```
4   int Errores = 0;
-
5   String  NomArchi_01 = "DATALOG.TXT",
6           NomArchi_02 = "De_La_PC.txt",
7           NomArchi_03 = "Archi_01.txt",
8           NomArchi_04 = "Archi_02.txt",
9           NomArchi_05 = "Archi_03.txt",
10          NomArchi_06 = "Archi_04.txt",
11          NomArchi_Aux = "";
-
12  String  Carpeta_R = "/",
13        Carpeta_A = "/a",
14        Carpeta_AB = "/a/b",
15        Carpeta_ABC = "/a/b/c",
16        Carpeta_Z = "/z";
-
17  File MiArchivo;
-
18  void setup( ) {
19  //*************************************************************************
20    Serial.begin(9600);
21    while (!Serial) {
22      ; // Esperando se conecte el Puerto serie
23    }
24  //*************************************************************************
25  Serial.print("Inicializando Tarjeta SD...");
26   if (!SD.begin(PinLectoGrabadora)) {
27     Serial.println("Falla de Inicialización!");
28     while (1);
29    }
30    Serial.println("inicialización OK!.");
31    Serial.println(" ");
32  //*************************************************************************
33    Serial.println("Creando Carpeta (Directorios) '" + Carpeta_A + "'");
34    delay(1000);
35    if(SD.mkdir(Carpeta_A)){
36      Serial.println("Carpeta '" + Carpeta_A + "' Creada");
37    }else{
38      Serial.println("Error Creando Carpeta '" + Carpeta_A + "'");
39      Errores++;
40    }
-
41    Serial.println("Creando Carpeta (Directorios) '" + Carpeta_AB + "'");
42    if(SD.mkdir(Carpeta_AB)){
43      Serial.println("Carpeta '" + Carpeta_AB + "' Creada");
44    }else{
45      Serial.println("Error Creando Carpeta '" + Carpeta_AB + "'");
46      Errores++;
47    }
-
48    Serial.println("Creando Carpeta (Directorios) '" + Carpeta_ABC + "'");
49    if(SD.mkdir(Carpeta_ABC)){
50      Serial.println("Carpeta '" + Carpeta_ABC + "' Creada");
51    }else{
52      Serial.println("Error Creando Carpeta '" + Carpeta_ABC + "'");
```



Vista en la PC, de la Estructura que crearemos en la Tarjeta SD. Para Que usted también pueda visualizar esto en la PC, consulte **Apéndice B** al final de esta guía

Puede consultar el "**Apéndice A - Librería "SD.h" Clases y Métodos Disponibles**".

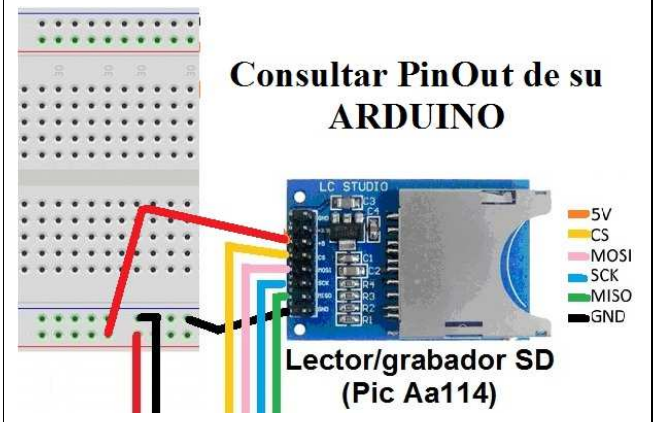
```
53        Errores++;
54      }
55      Serial.println(" ");
-
56    // En esta acción se genera un error Apropósito, pero no se contempla
57      Serial.println("Borrando Carpeta (Directorios) '" + Carpeta_Z + "'");
58      if(SD.rmdir(Carpeta_Z)){
59        Serial.println("Carpeta '" + Carpeta_Z + "' Borrada");
60      }else{
61        Serial.println("Error Borramdo Carpeta '" + Carpeta_Z + "'");
62      }
63      Serial.println(" ");
64    //*************************************************************************
65      Serial.println("\nCreando Archivos de Trabajo");
66      delay(1000);
67      Errores += CreaArchivos( NomArchi_01 );
68      Errores += CreaArchivos( NomArchi_02 );
-
69      NomArchi_Aux = ArmaNombreArchivo(Carpeta_A, NomArchi_03);
70      Errores += CreaArchivos( NomArchi_Aux );
71      NomArchi_Aux = ArmaNombreArchivo(Carpeta_AB, NomArchi_04);
72      Errores += CreaArchivos( NomArchi_Aux );
-
73      NomArchi_Aux = ArmaNombreArchivo(Carpeta_AB, NomArchi_05);
74      Errores += CreaArchivos( NomArchi_Aux );
-
75      NomArchi_Aux = ArmaNombreArchivo(Carpeta_ABC, NomArchi_06);
76      Errores += CreaArchivos( NomArchi_Aux );
77      Serial.println(" ");
78    //*************************************************************************
79      if(Errores){
80        Serial.print("DEBE REVISAR - Hubo '");
81        Serial.print(Errores);
82        Serial.println("' ERRORES en el Proceso");
83      }else{
84        Serial.println("Preparación de Entorno... EXITOSO!");
85      }
86    }
-
87    String ArmaNombreArchivo(String carpeta , String archivo){
88      String Nombre = "";
89      Nombre.concat(carpeta);
90      Nombre.concat("/");
91      Nombre.concat(archivo);
92      return(Nombre);
93    }
-
94    int CreaArchivos(String Archi){
95      File MiArchivo;
96      int Error = 0;
97      if ( SD.exists( Archi ) ){
98        Serial.println("ERROR: Archivo '" + Archi + "' Ya Existe");
99        Error = 1;
100     } else {
101       MiArchivo = SD.open( Archi, FILE_WRITE);
```

```
102     MiArchivo.close( );
103     if ( SD.exists( Archi ) ){
104       Serial.println("Archivo '" + Archi + "' Creado Exitosamente..!");
105     }else{
106       Serial.println("Error: Archivo '" + Archi + "' No pudo ser Creado..!");
107       Error = 1;
108     }
109   }
110   return(Error);
111 }
-
112 void loop( ) {
113   // Ninguna acción por ahora.
114 }
```

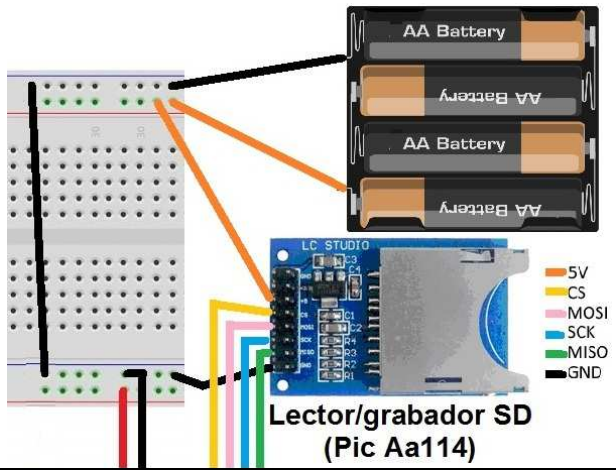
Programo comandos y llamo a las funciones desde la Función **Setup( )** para que se ejecuten solo una vez.

CIRCUITO PARA NUESTRO PROYECTO

Lista de Materiales: 1 Porta Pilas para 4 pilas comunes AA (1,5V) - 1 Módulo Lector/grabador De Sd Card - Pic Aa114 – 8 a 11 Cables de conexión (dependiendo del modelo de arduino serán Hembra/Hembra o Macho/Hembra) - 1 Tarjeta Micro SD con adaptador (Máx. 16Gb) - 1 Protoboard - Placa Arduino y 1 Cable USB.



Dispositivo Alimentado desde Placa Arduino



Dispositivo con Alimentación externa

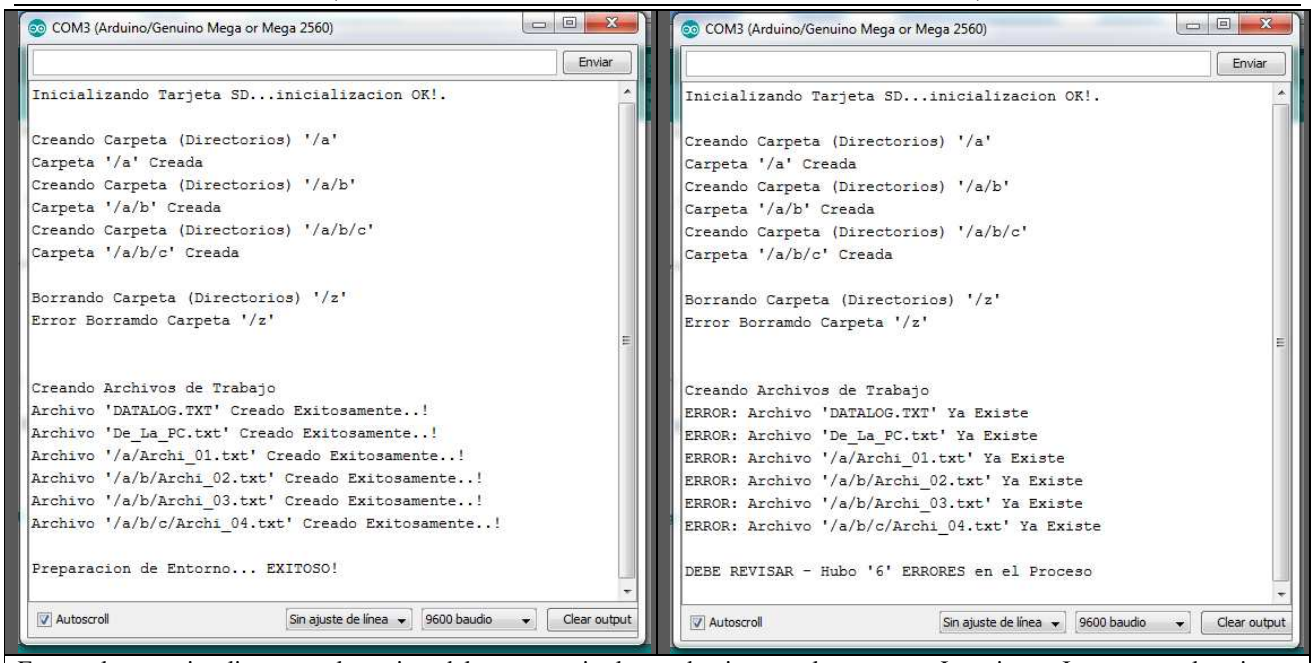
-o-

<u>Los cables deberán ser conectados</u> (abajo) de izquierda a derecha, deberán ser conectado a la placa según corresponda al Modelo Arduino que uses:

Cables	Rojo	Negro	Amarillo	Rosa	Celeste	Verde
Lecto Grabadora SD	**5V**	**GND**	**CS**	**MOSI**	**SCK**	**MISO**
UNO - Nano y Mini Pro	**5V**	**GND**	**Pin 9**	**Pin 11**	**Pin 13**	**Pin 12**
Mega	**5V**	**GND**	**Pin 7**	**Pin 51**	**Pin 52**	**Pin 50**

Si opta por alimentación Externa, recuerde siempre unir el negativo de la alimentación externa con GND de Arduino. Para cualquiera de los dos casos, el programa es el mismo, ya que el control siempre lo tiene Arduino. Una ves que hayas terminado con este ejemplo, no desarmes el dispositivo, ya que los próximos ejemplos usaran la misma configuración.

Ejecuta dos veces el programa, y la primera vez, todo debería resultar exitoso, sin embrago la segunda vez, podrás detectar errores en el proceso. Experimenta y capitalizaras experiencias invaluables para el futuro.



Esto es lo que visualizara en el monitor del puerto serie, luego de ejecutar el programa. La primera Imagen con la primera ejecución. La segunda imagen con la segunda vez, marcando los errores.

En este momento, ya tenemos el entorno de trabajo creado en la Tarjeta SD. Ahora podemos comenzar la el programa de la etapa 2. Recuerde que el circuito es el mismo, así que solo debe cambiar el programa.

Ahora Veremos si la estructura creada por nuestro primer programa. El segundo programa ingresa y detecta el primer nivel de carpetas, o **Directorio Raíz** de la unidad de almacenamiento, nuestra **Tarjeta SD** (Identificado en nuestro programa como "**root**"). Pasando desapercibido el contenido de las otras carpetas anidadas. Pueda que con esto nos alcance para saber lo que nos hace falta.

(Programa "700_Lector_Grabador_Sd_06_Lista_Directorio_02_Lista")

Etapa B

```
1  #include <SPI.h>
2  #include <SD.h>
-
3  #define PinLectora_SD 7
-
4  File root;
5  File archivo;
-
6  void setup( ){
7    Serial.begin(9600);
8    while (!Serial) {
9      ; // Esperamos que el puerto serie este abierto.
10   }
11 //*****************************************************************
12   Serial.print(F("Iniciando SD ..."));
13   if ( !SD.begin( PinLectora_SD ) ){
14     Serial.println( "Error al iniciar" );
15     return(0);  // Ver explicación
16   }
17   Serial.println( "Iniciado correctamente \n" );
18 //*****************************************************************
19   root = SD.open("/");
20
21
```

Programo Acciones y comandos

```
22    do{                                                    se ejecuten solo una vez.
23      archivo = (root.openNextFile( ));
24      if(!archivo){
25        // Carpeta Vacía o no hay archivo siguiente
26      }else{
27        Serial.print(archivo.name( )); //Imprimo el nombre
28        if (archivo.isDirectory( )){ //Si es un directorio
29          Serial.print("\t \t directorio");
30        }else{  //Si es un archivo
31          Serial.print(" ");
32          Serial.print(archivo.size( ), DEC);
33          Serial.print(" Bytes");
34          Serial.print(" \t archivo");
35        }
36        Serial.print("\n");
37      }
38    }while(archivo);
39    root.close( );
40    Serial.println("\n Proceso Terminado!");
41  }
42  void loop( ){
      // Ninguna acción por ahora.
    }
```

Para Analizar la tercera etapa, deberíamos introducirnos en la Recursividad, un tema un poco avanzado, sin embargo para no excluirlo, realizaré una rápida explicación (ver "**Apéndice C – Recursividad**" – Al final de la Guía). Si le interesa aprender más de este tema, consúlteme, que con gusto ampliare tanto como le haga falta.

(Programa "700_Lector_Grabador_Sd_06_Lista_Directorio_03_Recursion")

```
1   #include <SPI.h>                                          Etapa C
2   #include <SD.h>
-
3   #define PinLectora_SD 7
4   int C;
-
5   void setup( ){
6     Serial.begin(9600);                  Programo Acciones y comandos
7     while (!Serial) {                    en la Función Setup( ) para que
8       ; // Esperamos que el puerto serie este abierto.   se ejecuten solo una vez.
9     }
10  //*********************************************************************
11    Serial.print(F("Iniciando SD ..."));
12    if ( !SD.begin( PinLectora_SD ) ){
13      Serial.println( "Error al iniciar" );
14      return(0);  // Ver explicación
15    }
16    Serial.println( "Iniciado correctamente \n" );
17  //*********************************************************************
18    File root = SD.open("/");
19    Directorio(root, 0);
20    root.close( );
21  //*********************************************************************
22    Serial.println("\n Proceso Terminado!");
```

```
23  }
-
24  void Directorio(File dir, int Corrimientos){
25   File archivo;
26   do{
27     archivo = dir.openNextFile( );
28     if(archivo){
29       for(C=0; C < Corrimientos; C++){
30         Serial.print(".");
31       }
32       Serial.print(archivo.name( )); //Imprimo el nombre
33       if(archivo.isDirectory( )){    //Si es un directorio
34         Serial.println("\t \t directorio");
35         Directorio(archivo, Corrimientos + 4);
36       }else{ //Si No es Directorio entonces es un archivo
37         Serial.print(" ");
38         Serial.print(archivo.size( ), DEC);
39         Serial.println(" Bytes  \t archivo");
40       }
41     }else{
42       //No Hay un archivo siguiente o Carpeta Vacía
43     }
44   }while(archivo);
45   archivo.close( );
46  }
-
47  void loop( ){
48    // Ninguna acción por ahora.
49  }
```

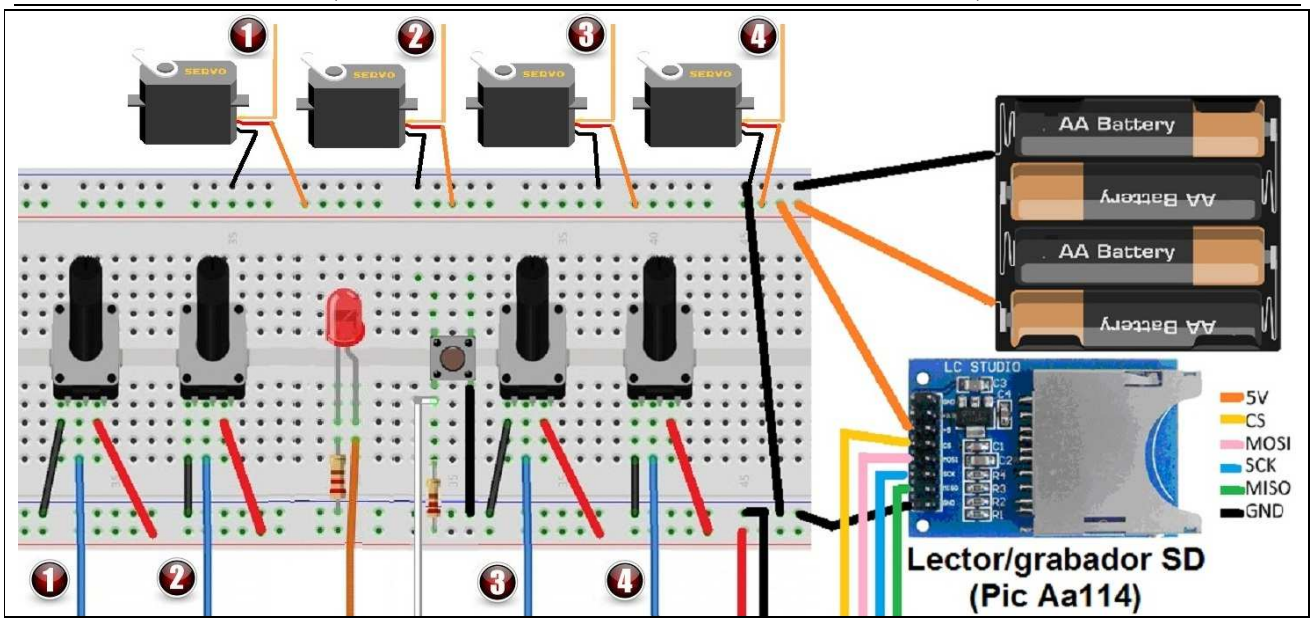
Por consultas, ver "**Apéndice C – Recursividad**" – Al final de la Guía

8) Controlar los Servos desde los potenciómetros y cada vez que se presione el Botón, Grabar en un Archivo, las posiciones de cada uno de los Servo Motores (Controlados por los Potenciómetros).

Este ejercicio queda como tarea para los alumnos. Cada uno deberá ejercitar la imaginación, ya que no es más que una combinación de lo estudiado en guías anteriores. Esto es muy practico para calibrar dispositivos. Tarea que un poco más adelante será prácticamente indispensable realizar

CIRCUITO PARA NUESTRO PROYECTO

Lista de Materiales: 1 Porta Pilas para 4 pilas comunes AA (1,5V) - 1 Módulo Lector/grabador De Sd Card - Pic Aa114 – 1 Led (Comunes de 5 milímetros) – 1 Top Swich (Boton Pulsador) - 1 Resistencia de 470 Ω (De las que compramos para usar con los LED) – 1 Resistencias de 5K Ω (de las que compramos para usar con los botones) – 4 Potenciómetros - 4 Servomotores - 38 Cables de conexión (dependiendo del modelo de arduino serán Hembra/Hembra o Macho/Hembra) - 1 Tarjeta Micro SD con adaptador (Máx. 16Gb) - 1 Protoboard - Placa Arduino y 1 Cable USB.



Apéndice A - Librería "SD.h" Clases y Métodos disponibles

(Apéndice duplicado en guías: Lectora/Grabadora Tarjetas SD - Ethernet Shield W5100)

La Librería "SD.h" contiene un objeto llamado SD, y nos pone a nuestra disposición un conjunto de Métodos (Funciones), las que trabajan conjuntamente con un Objeto de la Clase File, que también contiene métodos (Funciones). A continuación las funciones (Métodos) y una rápida explicación de lo que hace cada una.

SD.begin(PinTarjetaSD)	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 ***PinTarjetaSD*** , se usa el valor por defecto del CS por hardware. Los demás pines deben estar conectados al SPI por hardware del Arduino.
SD.exists(NombreArchivo)	Comprueba si existe el archivo especificado, **NombreArchivo** 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(NombreDirectorio)	Crea el directorio (carpeta) especificado, si los subdirectorios no existen, también se crearan. Por ejemplo: SD.mkdir("**Arduino/Trabajo/EJ_01**), crea la carpeta "**EJ_01**" y si las carpetas "**Arduino**" y "**Trabajo**" no existen, entonces también serán creadas. La función retorna "**true**" si la creación del directorio fue exitosa de lo contrario nos retorna un "**false**"
SD.remove(NombreArchivo)	Elimina el archivo "**NombreArchivo"** de la tarjeta SD, se debe incluir el directorio si es que esta dentro de uno. Solo elimina el archivo y no el directorio. Devuelve "**true**" se logra eliminar el archivo de lo contrario nos retorna un "**false**".
SD.rmdir(NombreDirectorio)	Eliminar el directorio "**NombreDirectorio"** 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(NombreArchivo, Modo)	Abre el archivo especificado en "**NombreArchivo"** y se debe incluir el nombre del directorio, si el archivo está en uno (carpeta). Si el archivo no existe, se creara un archivo con el nombre especificado, pero no será posible crear el directorio si este no existe. Ésta función nos retorna un objeto tipo FILE, el cual es necesario declararlo antes, igual que se declara una variable. Por ejemplo:
Veamos un Ejemplo:	

```
File MiArchivo;
MiArchivo = SD.open("Trabajo/Archivo.txt", FILE_WRITE);
```

Siempre viene bien recordar: La Clase es "**File**", el Objeto es "**MiArchivo**".

Modos de Aperturas. Modos de apertura del archivo en la Función SD.open( ):

FILE_READ	archivo como solo lectura
FILE_WRITE	lectura y escritura (si mode es

En caso no se especifique el modo de apertura, el modo, por defecto será FILE_READ

Suponiendo Que hemos declarado una variable (Instanciamos un Objeto) de la Clase "File" (según vemos en el siguiente código) `File MiArchivo;` Entonces las siguientes funciones (métodos) se usarán:

FUNCIONES (Métodos) DE LA CLASE "File"	
MiArchivo.available( )	Comprueba si hay bytes disponibles para leer en el archivo y retorna el número de bytes que falta por leer.
MiArchivo.close( )	Cierra el archivo, y recién en este momento, los datos se terminan de guardar en la tarjeta SD, pudiendo extraerla en forma segura.
MiArchivo.flush( )	Con este comando, se asegura de que se escribe en la SD todo. Algo que generalmente se hace automáticamente al hacer "**close( )**". Este comando puede usarse en Situaciones críticas en que pudiera suceder algo. De esta forma aseguramos la información.
MiArchivo.isDirectory( )	Indica si el fichero abierto es (retorna "**true**") o no (retorna "**false**") un directorio.
MiArchivo.peek( )	Lee un char del archivo, en la posición que marca el apuntador y no avanza.
MiArchivo.position( )	Retorna la posición actual dentro del archivo, en donde se leerá o escribirá el siguiente byte.
MiArchivo.print(datos)	Esta función tiene las mismas características que un Serial.print( ); "**datos**" puede ser una variable o texto, el cual será enviado como caracteres. Si queremos agregar al final un salto o nueva línea se usa "**MiArchivo.println(datos)**"
MiArchivo.println(datos)	Idéntica a "**MiArchivo.print(datos)**" solo que agrega al final un salto de línea.
MiArchivo.read( )	Lee un byte de la variable File (archivo abierto anteriormente con SD.open( )) y avanza a la próxima posición
MiArchivo.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.size( )"
MiArchivo.size( )	Retorna el tamaño en bytes del archivo.
MiArchivo.write(dato)	Escribe un byte en el archivo, el archivo debe estar abierto en modo lectura y escritura. Usando **MiArchivo**.write(datos, Tamaño) se puede escribir un array de byte (**datos**) pero se debe especificar el tamaño "**Tamaño**".
	https://www.arduino.cc/en/Reference/SD

Apéndice B - Ventana de Comandos o Símbolo del Sistema

(Apéndice duplicado en guías: Lectora/Grabadora Tarjetas SD - Ethernet Shield W5100)

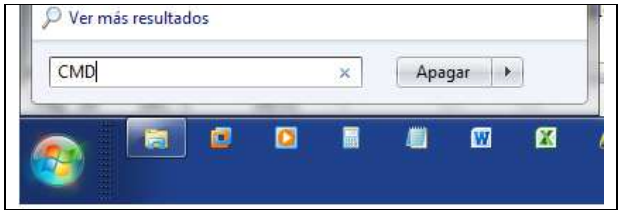
Si bien acá disponemos de muchos comandos, nos centraremos en los que necesitamos para poner en funcionamiento y usar nuestro dispositivo.

La consola de comandos o Símbolo del sistema "**CMD**" es uno de los recursos más utilizados wn Windows desde hace algunos años, sobre todo por los administradores de sistemas para realizar múltiples tareas tanto a nivel de información como a nivel de gestión y soporte.

Si en tu PC Tienes Windows 10, deberás darle a esa ventana los Privilegios de Administrador. Si es tu caso, lee este artículo: "www.solvetic.com/tutoriales/article/2341-formas-de-abrir-consola-de-comandos-en-windows-10/".

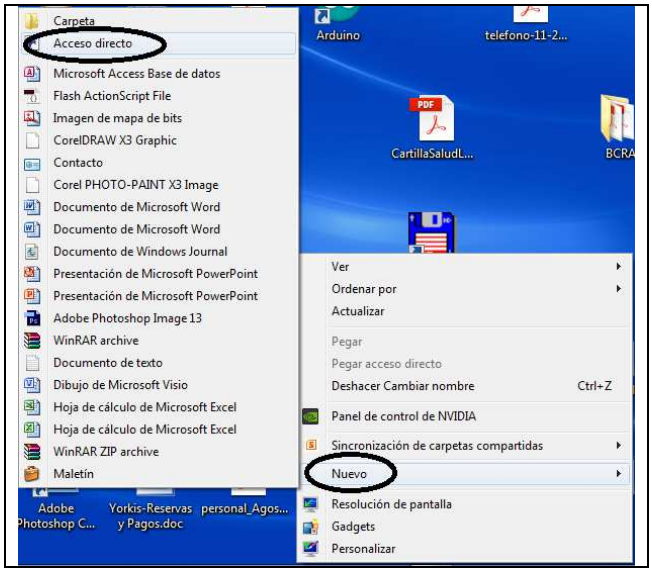
Existen Muchas formas de Ingresar a la consola de comandos, incluso cambian las formas de ingresar con las versiones de Windows, por algunas persisten invariablemente en el tiempo. Acá te muestro dos de ellas.

1- Con el Mouse hacemos Clic en el Botón Inicio de Windows (Abajo a la Izquierda) entonces Windows abrirá el menú, y bien abajo, casi pegado al botón de inicio (que acaban de hacer clic) Dependiendo de la Versión de Windows dirá "Ejecutar" o "Buscar Programas y Archivos". Justo ahí escriben "**CMD**" y "**ENTER**". Instantáneamente se abriera la Ventana de comandos.



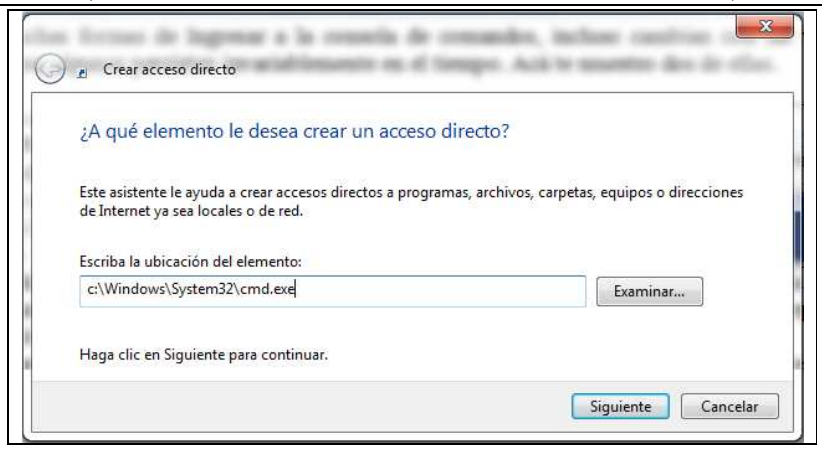
Cada Vez que nos haga Falta Trabajar con el símbolo del sistema, repetimos esta acción y listo.

2- Y para quienes deban entrar repetidas veces y prefieran algo más rápido, pueden crear un acceso directo (Igual que se crea uno para cualquier aplicación).



Entonces, con el Mouse sobre el escritorio de Windows, hacemos clic derecho, y seleccionamos "Nuevo" y a continuación "Acceso Directo".

Inmediatamente después aparecerá una ventana de dialogo donde deberá buscar la aplicación correspondiente. En este caso "**CMD.exe**". Para Facilitarle las cosas, simplemente escriba en ella "**c:\Windows\System32\cmd.exe**" y presione el Botón Siguiente.



Y en la próxima ventana de dialogo, debe poner en nombre que tendrá el icono de su nuevo acceso directo. Listo ya puede acceder cuando guste y rápidamente.

Desde acá podrá ejecutar infinidad de comandos. Por ahora solo veremos unos pocos comandos.

Comando "**tree**".

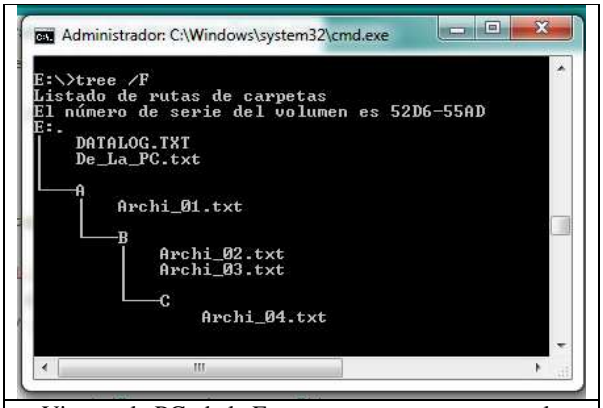
En su PC, escriba en la ventana de comandos:

tree y Presione tecla "Enter"

Todos los comandos se escriben sin las comillas ☺

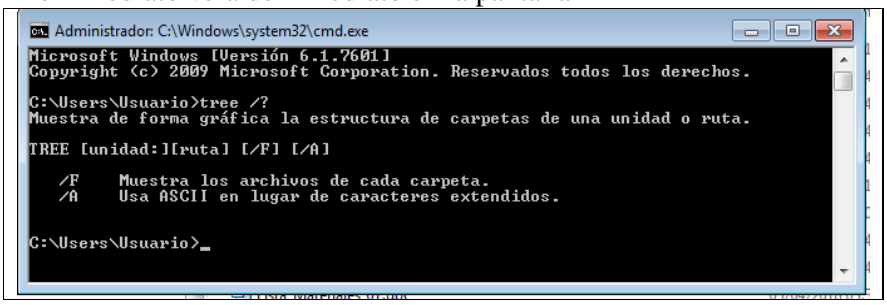
Si le interesa conocer más opciones de este comando escriba:

tree /? y Presione tecla "Enter"



Vista en la PC, de la Estructura que crearemos en la Tarjeta SD.

De inmediato verá de inmediato en la pantalla



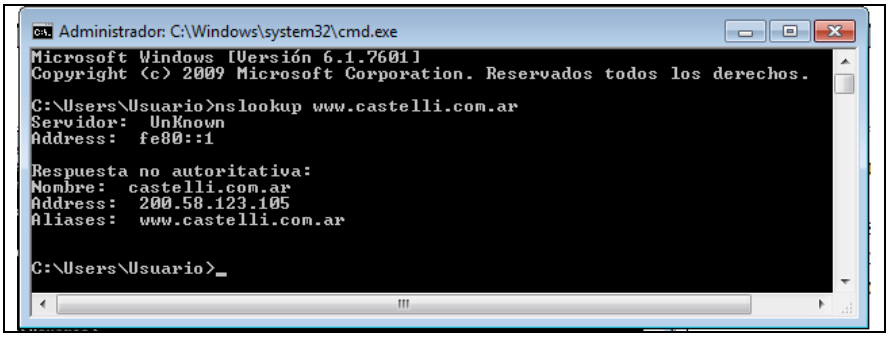
Y pensar que hubo una época en que, lo que hoy llamamos Símbolo del sistema o consola de comandos, era la única forma de interactuar con el ordenador.

Y una carita feliz se veía así :-)

Comando **nslookup.** (para obtener la dirección IP de una URL)

Con este comando obtendrá la dirección IP de una determinada URL en Internet. Por ejemplo Escriba en su consola de comandos:

nslookup www.castelli.com.ar y presione Enter.



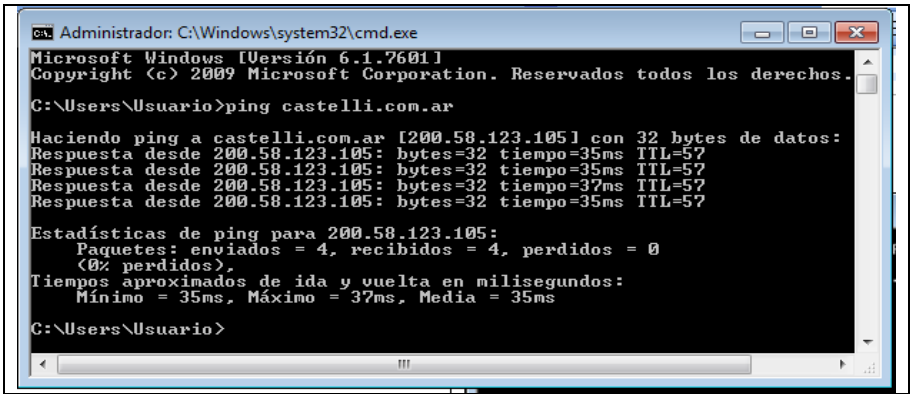
De inmediato vera el IP

correspondiente a la URL (dirección) que solicito, en este caso, "www.castelli.com.ar".

Si le interesa conocer más opciones de este comando escriba: **nslookup /?**
De inmediato verá de inmediato en la pantalla todas las opciones.

Comando **ping.**

Con este comando podemos saber si una URL esta o no Activa (si una dirección de Internet esta funcionando), el tiempo medio de respuesta y también sabremos su Dirección IP.



Por ejemplo Escriba en su consola de comandos:

ping www.castelli.com.ar

Y de inmediato visualizara en su pantalla toda la información requerida

Comando **help.**

Si quisiera profundizar en este modo de trabajo con Windows, entonces puede usar el comando "**help**" y de inmediato, le serán listados todos los comandos disponibles, junto a una breve descripción. Sin embargo puede profundizar cada comando escribiendo, el comando seguido por "**/?**".

Escriba en la ventana de comandos **Help** y presione Enter

Apéndice C - Recursividad

Se dice que una función es recursiva cuando se llama a si misma; algo así como si una persona necesitara ayuda para levantar un paquete y llamara un duplicado suyo para que lo ayude, y si todavía no pudieran levantar el paquete (la persona y su duplicado), el duplicado llamaría a un duplicado suyo; esto de las llamadas a los duplicados continuará repitiéndose hasta que entre todos, puedan levantar el paquete. Y en el momento de irse todos a descansar, primero se irán los últimos en llegar, es decir; el primero en irse será el ultimo en ser llamado.

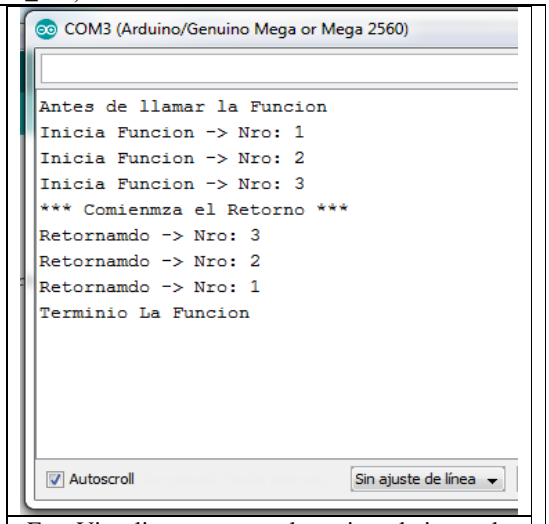
Seguidamente, y habiendo comprendido la idea anterior, una explicación mejor: Cuando una la función es llamada, esta reserva espacio para todas las variables e indicadores internos que le son necesarios para funcionar, **en alguna parte de la memoria**.
Luego si la función se llama a si misma (recursión) se genera en memoria un duplicado del original (comenzamos otra vez); repitiendo este proceso tantas veces como sea necesario. Y cada vez que se llame a si misma, realiza la misma operación, volviendo a crear otro bloque con información distinta, aunque siempre con formato idéntico, por ser una nueva copia.

Algo que debe tener presente es, cuando use funciones recursivas, recuerde establecer condiciones de salida (finalización) a estos bucles, ya que si no lo hiciera, comenzaría un bucle infinito, y esto produciría un desbordamiento del Snack (Se acaba el especio de memoria), con el consiguiente error que pararía el programa de manera brusca, perdiendo todo.
Veamos un ejemplo.

A- Realizar un programa en el que una función recursiva cuente desde cero al cuatro, mostrando por monitor los valores tomados durante la ida y la vuelta de la recursión.

(Programa "900_Recursividad_01")

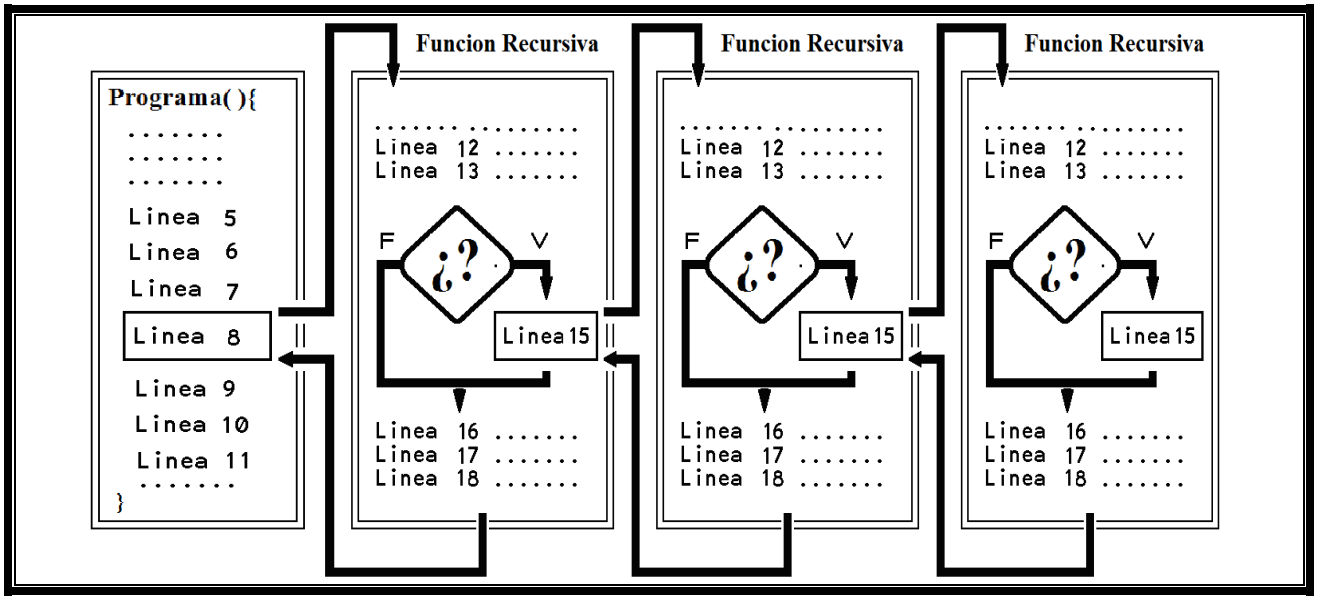
```
1   int Valor_Inicial = 1;
-
2   void setup( ){
3     Serial.begin(9600);
4     while (!Serial) {
5       ; // Esperamos que el puerto serie este abierto.
6     }
7     Serial.println("Antes de llamar la Función");
8     Cuenta( Valor_Inicial );
9     Serial.println("Terminó La Función");
10  }
-
11  void Cuenta(int Nro){
12    Serial.print("Inicia Función -> Nro: ");
13    Serial.println(Nro);
14    if(Nro < 3){
15      Cuenta( Nro + 1 );
16    }else{
17      Serial.println("*** Comienza el Retorno *** ");  // Esta línea Podría eliminarse
18    }
19    Serial.print("Retornamdo -> Nro: ");
20    Serial.println(Nro);
21    delay(1000);
22  }
-
23  void loop( ){
24    // Ninguna acción por ahora.
25  }
```



Este Visualizaremos por el monitor al ejecutarlo

Programo Acciones y comandos en la Función **Setup( )** para que se ejecuten solo una vez.

A continuación el Esquema de los pasos o secuencia de ejecución de la Función Recursiva (Note que hay correspondencia entre los números de línea del esquema y el programa)



B- Realizar un programa en el que una función recursiva Realice la Sumatoria de todos los números enteros positivos comprendidos entre 1 y 5 (Ambos inclusive).

A continuación dos versiones del mismo ejercicio. La diferencia reside en la forma de realizar la llamada a la recursión. También hay una diferencia el los resultados visualizados, puede apreciarlo en la imágenes a continuación de código

(Programa "900_Recursividad_02_Suma_A")

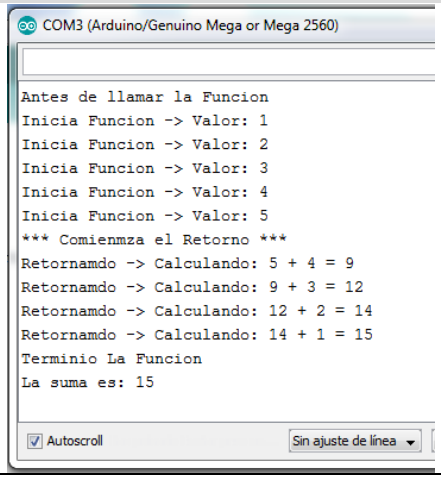
```
1   int Valor_Inicial = 1,
-       Resultado;
2   void setup( ){
3    Serial.begin(9600);
4    while (!Serial) {
5      ; // Esperamos que el puerto serie este abierto.
6    }
7    Serial.println("Antes de llamar la Función");
8    Resultado = Suma(Valor_Inicial);
9    Serial.println("Terminó La Función");
10   Serial.print("La suma es: ");
11   Serial.println(Resultado);
12  }
-
13  int Suma(int Valor){
14   int R = 0,
15       S;
16   Serial.print("Inicia Función -> Valor: ");
17   Serial.println(Valor);
18   delay(700);
19   if(Valor < 5){
20     R = Suma( Valor + 1 );
21     S = Valor + R;
22     Serial.print("Retornando -> Calculando: ");
23     Serial.print(R);
-      Serial.print(" + ");
24     Serial.print(Valor);
25     Serial.print(" = ");
26     Serial.println(S);
27   }else{
28     S = Valor;  // Este es el ultimo valor
29     Serial.println("*** Comienza el Retorno *** ");
30   }
31   delay(700);
32   return(S);
33  }
34
35  void loop( ){
36    // Ninguna acción por ahora.
37  }
```

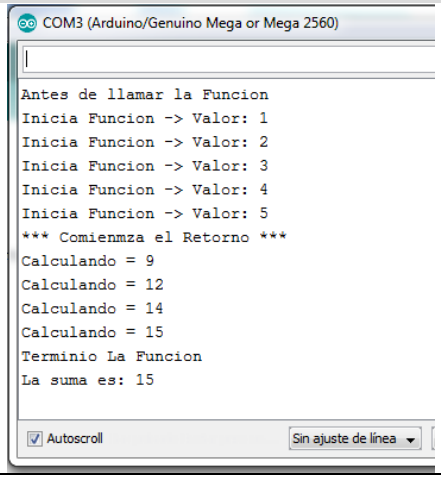
```
int Valor_Inicial = 1,
    Resultado;
 void setup( ){
  Serial.begin(9600);
  while (!Serial) {
    ; // Esperamos que el puerto serie este abierto.
  }
  Serial.println("Antes de llamar la Función");
  Resultado = Suma(Valor_Inicial);
  Serial.println("Terminó La Función");
  Serial.print("La suma es: ");
  Serial.println(Resultado);
 }

 int Suma(int Valor){
  int S = 0;
  Serial.print("Inicia Función -> Valor: ");
  Serial.println(Valor);
  delay(700);
  if(Valor < 5){
     S += Valor + Suma( Valor + 1 );
     Serial.print("Calculando = ");
     Serial.println(S);
  }else{
     S = Valor;  // Este es el ultimo valor
     Serial.println("*** Comienza el Retorno *** ");
  }
  delay(700);
  return(S);
 }

 void loop( ){
    // Ninguna acción por ahora.
 }
```

Programo Acciones y comandos en la Función **Setup( )** para que se ejecuten solo una vez.





C- Realizar un programa en el que una función recursiva Realice el Calculo del Factorial de 5 (cinco).
(Programa "900_Recursividad_03_Factorial")

```
1   int Valor_Inicial = 5,
2       Resultado;
-
3   void setup( ){
4    Serial.begin(9600);
5    while (!Serial) {
6      ; // Esperamos que el puerto serie este abierto.
7    }
8    Serial.println("Antes de llamar la Función");
9    Resultado = Factorial(Valor_Inicial);
10   Serial.println("Termino La Función");
11   Serial.print("El Factorial de ");
12   Serial.print(Valor_Inicial);
13   Serial.print(" es: ");
14   Serial.println( Resultado );
15  }
-
16  int Factorial(int N){
17   if (N == 0 ){
18     return 1;
19    }else{
20     return (N * Factorial(N-1));
21     }
22  }
-
23  void loop( ){
24     // Ninguna acción por ahora.
25  }
```

Programo Acciones y comandos en la Función **Setup( )** para que se ejecuten solo una vez.

Quieres Más?

Si este tema realmente te interesa, te sugiero, veas la partes de Acceso a Ethernet / Internet (Está junto con el uso del **Ethernet Shield W5100).**

Ahí encontrarás los ejemplos donde se graba lo que lees en Internet.

Mira También donde se explica como Reiniciar Arduino, con Archivos de configuración.

Imagina ..!

CONTROL REMOTO INFRARROJO

01001100 01101111 01110011 00100000 01110000 01110010 01101001 01101101 01100101
01110010 01101111 01110011 00100000 01110000 01100001 01110011 01101111 00100000
01100100 01100101 00100000 01110101 01101110 00100000 01100110 01110101 01110100
01101111 01110010 01101111 00100000 01110000 01110010 01101111 01101101 01100101
01110100 01100101 01100100 01101111 01110010

Si tienes algunas Correcciones y/o Sugerencias, por favor contáctame.