

PRACTICA 6: BUS SPI. LECTOR TARJETAS SD

Ejercicio 1: Lectura/Escritura de archivo

El siguiente código emplea la escritura y lectura de una tarjeta SD.

Debemos incluir las librerías SPI y SD para ello. También declaramos la variable `myFile` de tipo `File` para poder manipular archivos.

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

File myFile;
```

En el `setup()` debemos indicar los pines de conexión para el bus SPI.

Hecho comprobamos la conexión con la tarjeta. Para ello pasamos por parámetro el pin asignado a los datos (CS) y si esta conexión no se realizó enviamos al puerto serie un mensaje de error.

Si es exitosa se procede y da los mensajes de confirmación.

Tras esto, procedemos a abrir el fichero con nombre *archivo.txt*. En caso de que no exista el fichero, el parámetro `FILE_WRITE` permite que se cree el archivo y se abra inmediatamente.

Escribimos algo en el fichero y lo cerramos.

Si el fichero que quisimos crear sí se creó correctamente entonces leemos el contenido y se envía este por el puerto serie. Si no se pudo crear, enviamos un mensaje de error.

```
void setup(){

  Serial.begin(9600);
  Serial.print("Iniciando SD ...");

  SPI.begin(18,19,23,4);

  if(! SD.begin(4)){
    Serial.println("No se pudo inicializar");
    while(1);
  }

  Serial.println("SD OK");

  Serial.println("inicializacion exitosa");

  //Escribimos en el fichero
  myFile=SD.open("/archivo.txt",FILE_WRITE); // FILE_WRITE: si no existe el
```

```
archivo, lo crea.  
// SD.remove("/archivo.txt");  
// myFile=SD.open("/archivo.txt");  
myFile.println("Me estoy comunicando con SD");  
myFile.println("Doble entrada");  
myFile.close();  
  
myFile=SD.open("/archivo.txt"); //Abrimos, mostramos y leemos  
if (myFile) {  
    Serial.println("archivo.txt:");  
    while (myFile.available()) {  
        String line=myFile.readStringUntil('\n');  
        Serial.println(line);  
    }  
    myFile.close();  
}  
  
else {  
    Serial.println("Error al abrir el archivo");  
}  
}  
  
void loop(){}
```

Así pues por el puerto serie deberíamos ver lo siguiente:

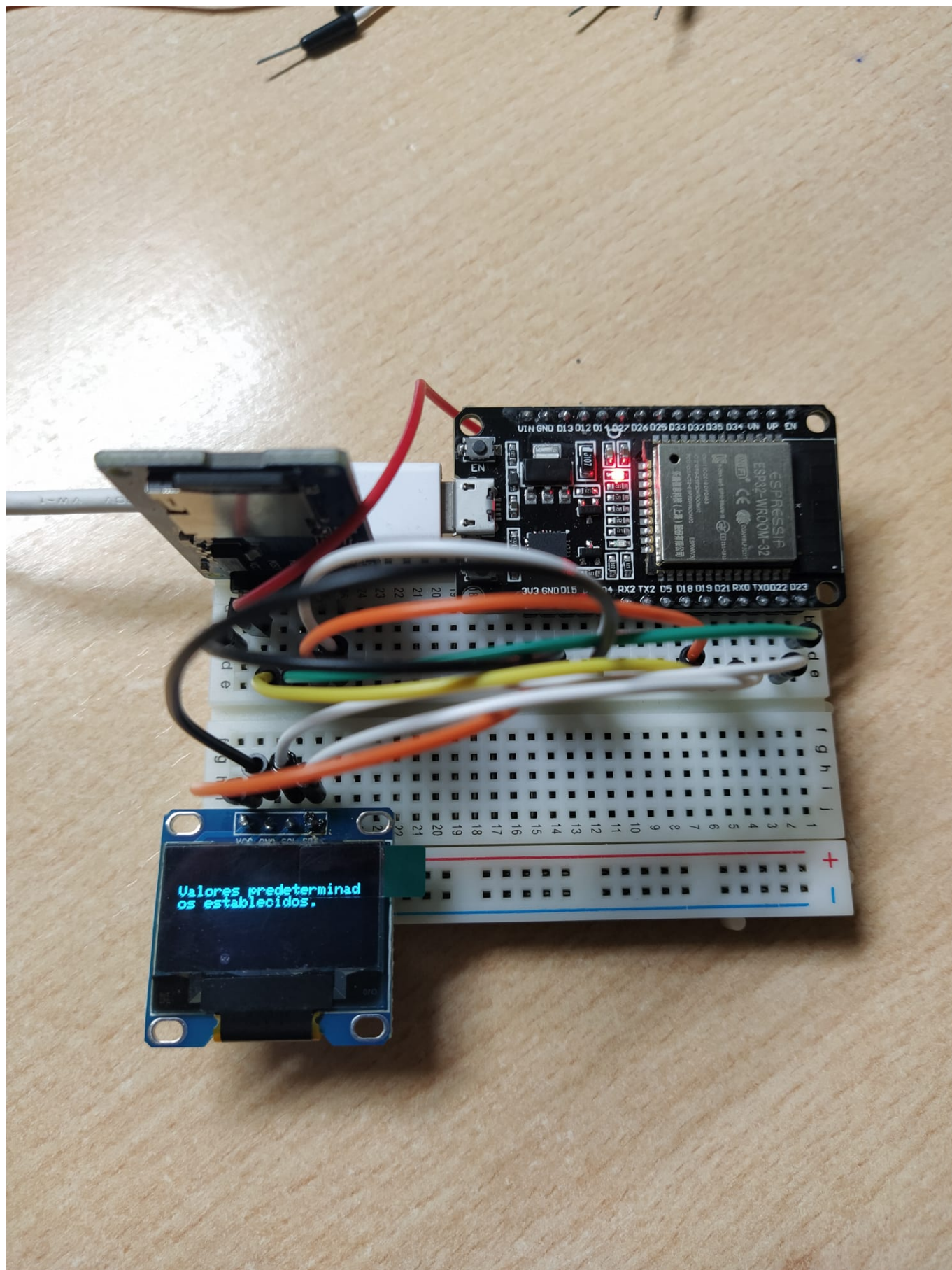
```
Iniciando SD ...sd init ok  
SD OK  
inicializacion exitosa  
Me estoy comunicando con SD  
Doble entrada
```

Este programa es un simple ejemplo de como usar el lector de tarjetas.

El lector de tarjetas puede ser útil sobretodo para almacenar datos si no tenemos una memoria RAM en nuestro microprocesador.

Como NO TENGO EL LECTOR RFID voy a utilizar el lector de tarjetas para otra aplicacion (tampoco puedo realizar EJERCICIO PRACTICO 2)

SALVADO DE CONFIGURACION PREDETERMINADA. Ejercicio practico subida de nota



En este ejercicio utilizaremos la tarjeta SD para guardar en un fichero los ajustes predeterminados, en este caso de la pantalla LCD I2C.

Requisitos

1. Para seguir adelante con esta practica, primero se debe crear el fichero *config.txt*. Ejecutar el siguiente codigo.

Para cambiar algun parametro de la configuración cambia los valores de los *myfile.println()*.

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

File myFile;

void setup(){

  Serial.begin(9600);
  Serial.print("Iniciando SD ...");

  SPI.begin(18,19,23,4);

  if(! SD.begin(4)){
    Serial.println("No se pudo inicializar");
    while(1);
  }

  Serial.println("SD OK");

  Serial.println("inicializacion exitosa");

  //Escribimos en el fichero
  myFile=SD.open("/config.txt",FILE_WRITE); // FILE_WRITE: si no existe el
archivo, lo crea.
  // SD.remove("/archivo.txt");
  // myFile=SD.open("/archivo.txt");
  myFile.println("ANCHO");
  myFile.println("128");
  myFile.println("ALTO");
  myFile.println("64");
  myFile.println("TAMAÑO");
  myFile.println("1");
  myFile.println("X");
  myFile.println("0");
  myFile.println("Y");
  myFile.println("16");
  myFile.close();

  myFile=SD.open("/config.txt"); //Abrimos, mostramos y leemos
  if (myFile) {
    Serial.println("config.txt:");
    while (myFile.available()) {
      String line=myFile.readStringUntil('\n');
      Serial.println(line);
    }
    myFile.close();
  }
```

```

    }

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

void loop(){}

```

2. Una vez hecho el paso 1, añadir la siguiente dependencia de librerías en el fichero *platformio.ini*

```

lib_deps =
    adafruit/Adafruit SSD1306@^2.4.4
    adafruit/Adafruit GFX Library@^1.10.7
    adafruit/Adafruit BusIO@^1.7.3

```

Desenvolupamento

- Cabezera Vamos a introducir las librerías necesarias para trabajar con el lector SD de bus SPI y la pantalla LCD.

```

#include <Arduino.h>

#include <SPI.h>
#include <SD.h>

#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

```

- Preambulo

Para hacer el código principal más reducido creamos dos funciones:

- *iniciarSD()* Tal y como su nombre indica inicializa la conexión con la tarjeta SD a través del bus SPI.

```

void iniciarSD(){

    Serial.print("Iniciando SD ...");
    SPI.begin(18,19,23,4);

    if(! SD.begin(4)){
        Serial.println("No se pudo inicializar");
    }
}

```

```

while(1);
}
Serial.println("SD OK");
Serial.println("inicializacion exitosa");

}

```

- *getConfig()* En esta funcion pide 5 parámetros donde grava los diferentes parametros predeterminados ya configurados. Estas variables son devueltas al programa.

Sabiendo el orden de la configuracion, usamos una linea auxiliar para leer la descripcion del parametro y el string line para los datos. Entonces convertimos el string con el dado deseado a un entero.

```

void getConfig(int & ancho, int & alto, int & size, int & x, int & y){
    File config;

    config=SD.open("/config.txt");

    if(config){

        String line, lineaux;

        lineaux=config.readStringUntil('\n');
        line=config.readStringUntil('\n');
        ancho=line.toInt();

        lineaux=config.readStringUntil('\n');
        line=config.readStringUntil('\n');
        alto=line.toInt();

        lineaux=config.readStringUntil('\n');
        line=config.readStringUntil('\n');
        size=line.toInt();

        lineaux=config.readStringUntil('\n');
        line=config.readStringUntil('\n');
        x=line.toInt();

        lineaux=config.readStringUntil('\n');
        line=config.readStringUntil('\n');
        y=line.toInt();

        config.close();}

    else {Serial.println("Archivo config.txt no encontrado");}
}

```

- Setup

Aquí utilizamos las dos funciones anteriores para conseguir los parametros de configuracion.

Tras esto iniciamos la declaracion del display. Usamos los parametros obtenidos para hacer tal declaracion.

Una vez hecha, limpiamos el display y imprimimos por pantalla (display) el mensaje de que la operacion se hizo correctamente y se resetea de nuevo preparado para ser usado.

```
void setup(){

  Serial.begin(9600);

  iniciarSD();

  int ancho, alto, size, x, y;
  String color;

  getConfig(ancho,alto,size,x,y);

  // Objeto de la clase Adafruit_SSD1306
  Adafruit_SSD1306 display(ancho, alto, &Wire, -1);

  // Iniciar pantalla OLED en la dirección 0x3C
  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
#ifdef __DEBUG__
    Serial.println("No se encuentra la pantalla OLED");
#endif
    while (true);
  }

  // Limpiar buffer
  display.clearDisplay();

  display.setTextSize(size);
  // Color del texto
  display.setTextColor(SSD1306_WHITE);
  // Posición del texto
  display.setCursor(x, y);
  display.println("Valores predeterminados establecidos.");
  // Enviar a pantalla
  display.display();
  delay(5);
  display.clearDisplay();

}

void loop(){}
```

Codigo final

```
#include <Arduino.h>

#include <SPI.h>
#include <SD.h>

#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

void iniciarSD(){

Serial.print("Iniciando SD ...");
SPI.begin(18,19,23,4);

if(! SD.begin(4)){
  Serial.println("No se pudo inicializar");
  while(1);
}
Serial.println("SD OK");
Serial.println("inicializacion exitosa");

}

void getconfig(int & ancho, int & alto, int & size, int & x, int & y){
  File config;

  config=SD.open("/config.txt");

  if(config){

    String line, lineaux;

    lineaux=config.readStringUntil('\n');
    line=config.readStringUntil('\n');
    ancho=line.toInt();

    lineaux=config.readStringUntil('\n');
    line=config.readStringUntil('\n');
    alto=line.toInt();

    lineaux=config.readStringUntil('\n');
    line=config.readStringUntil('\n');
    size=line.toInt();

    lineaux=config.readStringUntil('\n');
    line=config.readStringUntil('\n');
    x=line.toInt();

    lineaux=config.readStringUntil('\n');
    line=config.readStringUntil('\n');
    y=line.toInt();
```



```
        config.close();}

    else {Serial.println("Archivo config.txt no encontrado");}
}

void setup(){

Serial.begin(9600);

iniciarSD();

int ancho, alto, size, x, y;
String color;

getconfig(ancho,alto,size,x,y);

// Objeto de la clase Adafruit_SSD1306
Adafruit_SSD1306 display(ancho, alto, &Wire, -1);

// Iniciar pantalla OLED en la dirección 0x3C
if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
#ifdef __DEBUG__
    Serial.println("No se encuentra la pantalla OLED");
#endif
    while (true);
}

// Limpiar buffer
display.clearDisplay();


display.setTextSize(size);
// Color del texto
display.setTextColor(SSD1306_WHITE);
// Posición del texto
display.setCursor(x, y);
display.println("Valores predeterminados establecidos.");
// Enviar a pantalla
display.display();
delay(5000);
display.clearDisplay();

}

void loop(){}
}
```