



Lab 8: Serial Peripheral Interface (SPI)

Nivel 5, Obligatorio y reprobatorio

1. Objetivos

El presente laboratorio tiene por finalidad introducirlo al uso de un nuevo protocolo de comunicación: *Serial Peripheral Interface*, más conocido como SPI. Para concretar este objetivo se llevarán a cabo dos tareas de implementación.

Su primera tarea consistirá en la lectura y escritura de archivos en una tarjeta microSD. Todas las tarjetas SD son compatibles con el estándar MMC (*MultiMediaCard*), el cual usa un *bus* SPI para establecer la comunicación entre la tarjeta y el procesador en cuestión. Su segunda tarea consistirá en configurar y controlar un *display* TFT, el cual posee un controlador integrado y *framebuffer* ST7735, el cual hace que el módulo TFT sea fácil de manejar por medio del microcontrolador.

2. Descripción actividad

2.1. *Task 1: Gestión de data en una SDcard*

Para esta actividad se le han proporcionado distintos archivos que deberá modificar para poder llevar a cabo la lectura y escritura de datos en la tarjeta SD.

- En primer lugar, revise el archivo `SPI/SPI_implement_me.c` y complete las funciones con las rutinas faltantes para inicialización, transmisión y recepción SPI.
- Como podrá apreciar en el archivo `SDcardtest.c`, necesita hacer uso de la librería `USART_implement_me.h` que trabajó en las experiencias anteriores. Asegúrese de que el formato de dicha librería sea coherente con las funciones utilizadas en `SDcardtest.c`. En caso de ser necesario, realice las modificaciones pertinentes.
- Copie los archivos `SEPttest.txt`, `writeMe.txt` y `writeMe2.txt` en la tarjeta microSD e inserte la tarjeta dentro del *socket* del *display* TFT.
- Si su inicialización, transmisión/recepción SPI y librería USART son las correctas, el programa de prueba para la tarjeta SD debería funcionar sin problemas.



- Conecte el módulo TFT *display* según corresponda e inicie la comunicación serial. Primero, el programa abrirá el archivo de lectura `SEPtest.txt`. Luego, escribirá un mensaje en el archivo de prueba de escritura `writeMe.txt`. Finalmente, le solicitará enviar un mensaje por USART, para escribirlo en el archivo de prueba `writeMe2.txt`.
- Si logra visualizar la lectura de `SEPtest.txt` en el monitor serial y al introducir la tarjeta SD en su computador puede corroborar que los contenidos de los archivos `writeMe.txt` y `writeMe2.txt` son los correctos, ¡Felicitaciones, ha completado exitosamente esta *Task*!

NOTA: Para el caso de la librería USART, puede utilizar el *buffer* que usted prefiera (lineal, circular, *linked list*). La longitud del *buffer* también queda a su elección, pero se recomienda escoger un tamaño prudente para evitar problemas de *overflow*.

2.2. *Task 2: Rutina de prueba en pantalla TFT*

Teniendo parte de lo realizado en el *Task 1*, en esta actividad deberá modificar y completar los archivos entregados, para de esta forma correr una rutina de prueba en la pantalla TFT.

- En primer lugar, extender el archivo `SPI/SPI_implement_me.c` desarrollado en la sección anterior, agregando lo necesario para también cumplir con los requerimientos de esta *Task*, completando las rutinas de transmisión especificadas en cada archivo como corresponda.
- Complete las funciones de UART al igual que en el *Task 1*. No debería ser necesario realizar ninguna modificación extra.
- De haber realizado de forma correcta la extensión de la librería SPI, en su pantalla verá una serie de ejemplos gráficos. Con esto tienen una gran cantidad del trabajo realizado.
- **Importante** Antes de pasar al *Task* siguiente, tiene que entender cómo operan estas distintas rutinas de testeo. Gran cantidad de estas utilizan varias funciones definidas, tales como `wd(...)` y `wc(...)`. Este paso será muy necesario para lo siguiente, ya que de no comprender cómo funcionan, podría volverse extremadamente difícil finalizar el laboratorio.

2.3. *Task 3: Despliegue de su animal espiritual*

En esta actividad, deberá reunir los conocimientos adquiridos en el *Task 1* y *Task 2*, para poder desplegar una imagen en formato `.BMP` de su animal favorito :). Para esto, se le recomienda realizar los siguientes pasos.



- Estudie el formato `.BMP`, entienda su funcionamiento, headers y cómo se almacena la información de los colores en esta. A modo de ejemplo, puede ver la información de *bytes* de las imágenes de ejemplo subidas al repositorio, puede abrirlas con su editor de texto favorito y leerlas de esta forma.
- Guarde las imágenes de prueba en la tarjeta SD, junto con esto, cree y guarde su imagen de su animal favorito de nombre `animal.bmp`, siguiendo el estándar que aprendió en el punto anterior, es **importante** que siga esta guía de nombre, ya que será necesario para el *Task* final.
- Modifique y cree su propia librería de SPI, que sea capaz de acceder a un archivo `.BMP` de nombre específico en la tarjeta SD. Dado que una imagen contiene una gran cantidad de información, no podrá leerla completamente para desplegarla de forma inmediata. Por esto debe guardar la información en un buffer de algún tamaño determinado por usted e implementar un proceso iterativo, para así poder enviar los datos a la pantalla TFT y comenzar el despliegue de la imagen.
- Dadas las limitaciones de la pantalla TFT y la comunicación, el despliegue de la imagen tardará unos segundos. Es importante que este proceso sea lo **más rápido posible**. **NO** es aceptarán soluciones que tarden más de 60 segundos en mostrar la imagen completa en pantalla.

2.4. *Task* final a revisar

El task final consistirá en la unión del *Task* 1 y el Task 3 en un mismo código. Si bien el *Task* 2 no se revisará, es fuertemente recomendable que lo hagan antes de hacer el 3, ya que es la mejor forma de comprender.

Cosas importantes que deben considerar:

- Ambas tasks deben funcionar sobre un mismo código, es su decisión si se despliega primero una y luego la otra, o realiza un menú simple con UART para seleccionar ambas, pero **no pueden estar en códigos separados**.
- Debe tener precaución con el tamaño del programa a cargar en su microcontrolador ya que este no tiene un espacio infinito, por lo que si supera cierto límite, es probable que dañe lo dañe y tenga problemas. Debe revisar el tamaño de su programa antes de cargarlo.
- Tiene que indicar mediante alguna forma visual (un LED) que está utilizando la tarjeta SD, para que esta no sea retirada en ese momento y así evitar errores.



- Su despliegue de imágenes debe ser capaz de realizarlo con los archivos de otros compañeros, por lo que es importante que todos tengan una imagen de nombre **animal.bmp**, para que puedan intercambiar tarjetas SD sin problemas.

3. Lecturas recomendadas

- **ATmega328/P Complete Datasheet.**
- **MSP430x5xx and MSP430x6xx Family User's Guide.**
- **MSP430F552x, MSP430F551x Mixed-Signal Microcontrollers datasheet.**
- **Board** Página web de la pantalla TFT a utilizar en el laboratorio.
- **PFF library** Es la página del desarrollador de la librería PFF. Fíjese que hay información extra para cada función de la librería. En este laboratorio se utiliza una versión muy similar a la mostrada en esta página.
- **Display Controller** Página del fabricante del controlador del display ST7735.
- **Datasheet del display utilizado**
- **TFT Display** Contiene un tutorial de cómo usar el display en un ambiente de Arduino. Si bien no se está utilizando Arduino y sus librerías no funcionarán fuera de ese ambiente, contiene información útil y gran parte del código entregado está basado en esas librerías. Además el tutorial les entrega una buena impresión de las capacidades de este hardware.
- **BMP Format** Contiene todo lo necesario que se debe saber acerca del formato BMP. Provee un ejemplo de un archivo con configuración RGB24, en donde cada uno de esos colores es guardado en un valor de 8 bit. Comprenda el array de pixeles dentro de un archivo bitmap. Considere el ancho y alto del display TFT que se utilizará, para que la imagen tenga este mismo tamaño. No es necesario extenderse tanto con el DIB header, hay que mantener las cosas **básicas y simples**. Lo que se busca es que sean capaces de mostrar una imagen en el display.



4. Pauta de Evaluación

4.1. Consideraciones generales

- El laboratorio será evaluado exclusivamente con nota 1.0 (**R**eprobado), 5.5 (**A**probado) y 7.0 (**D**istinguido). En ningún caso habrán notas intermedias.
- Este laboratorio es **inapelablemente** reprobatorio. La no presentación de esta experiencia funcionando se traduce en la reprobación del ramo, indistintamente del registro de notas en experiencias anteriores.
- No se reciben trabajos después del módulo de presentación. Trabajos no entregados son calificados con nota 1.0 y son considerados dentro del criterio de aprobación del curso. La hora límite para inscribir a revisión es a las 10:20 hrs, posterior a esto se asignará una posición aleatoria.
- Cualquier consulta sobre los criterios de evaluación de cada laboratorio debe ser realizada en las **issues**, donde estará disponible para que sea revisada por todos los alumnos.



4.2. Criterios de Aprobación

Se requiere cumplir con todos los puntos mencionados a continuación para poder aprobar. No existen casos excepcionales. La elección del μC es libre.

1. Funcionamiento de los requerimientos. El alumno realiza una presentación de su trabajo y se responsabiliza de exponer que este satisfaga todos los requerimientos mínimos solicitados en la *Descripción de la actividad*, los cuales incluyen en este laboratorio:
 - Debe comprender cada parte de su código, referenciar si corresponde y entender su solución.
 - La actividad del *Task 1* funciona correctamente y puede ser leído en el terminal el archivo .txt especificado. Siguiendo las indicaciones, esta tarea debiese funcionar automáticamente.
 - Debe lograrse desplegar la imagen desde la tarjeta SD. Debe mostrar una imagen suya y otra leída desde la tarjeta de otro compañero al azar. Se debe utilizar el nombre estándar `animal.bmp` en cada tarjeta para agilizar esta labor.
 - *Concurrencia*: Ambas tareas deben poder ejecutarse concurrentemente. Resultaría ideal disponer de un menú para acceder a las distintas características, pero esta característica no representa distinción ni tampoco es totalmente necesaria.
 - *Hot swap*: La tarjeta debe poder ser removida e insertada mientras el código está siendo ejecutado. Indique mediante un LED que la tarjeta está siendo utilizada para indicarle al usuario que no debe retirarla en dicho instante¹. Al volver a insertarla, el programa debe seguir funcionando .
 - *Velocidad*: un despliegue de una imagen BMP en pantalla no debiese tardar más de 60 segundos. Un tiempo superior a este indica una mala implementación del algoritmo de lectura y/o escritura, por lo que su código también debe satisfacer este requisito.
2. Preguntas. Se le realizarán diversas preguntas que abarcan (no exclusivamente) los siguientes temas. Solo dispone de una oportunidad para la evaluación oral y será su ayudante corrector el que determinará si su nivel de conocimientos es el adecuado para esta experiencia.
 - ¿Qué hacen las macros `wd(...)` y `wc(...)`? ¿Qué ventaja entregan desde el punto de vista de la portabilidad de código?

¹De hacerlo, existe riesgo de dañar la tarjeta, y este requisito no apunta a manejar dichas situaciones, si no poder mantener el flujo cuando no está siendo usada la tarjeta.



- ¿Qué representa cada línea de código dentro de las secciones que deben ser implementadas?
- Detalles del protocolo SPI y las funciones más relevantes de la librería TFT que deben ser usadas en esta experiencia.
- Estándar BMP de imágenes *bitmap*. ¿Cuál es el flujo de un programa que lee BMP?

Importante: Dada la complejidad de este laboratorio y su valor como certificación de conocimiento en sistemas embebidos, en caso de detectarse escaso o nulo conocimiento de los conceptos aplicados a la experiencia –a discreción exclusiva del corrector– se aplicará la reprobación de la experiencia, y por consiguiente, del curso.

3. Formalidades: Para la presentación debe cumplirse con las siguientes formalidades:
 - a) Uso de buenas prácticas.
 - b) Código debidamente comentado y ordenado.
 - c) Existen diversas implementaciones para las estructuras de código requeridas en este laboratorio. Por esta razón, cualquier uso de ellas sin citar a los autores e indicar explícitamente la fuente –indistintamente de sus intenciones– constituye **plagio** y una violación al código de honor, razón por la cual se aplicarán las sanciones académicas establecidas.
4. Recomendaciones: Para agilizar el desarrollo de esta experiencia, se le aconseja seguir las siguientes recomendaciones:
 - Cuide el tamaño de sus archivos a *flashear*. Tamaños grandes pueden generar comportamientos inesperados en la tarjeta, a veces carentes de todo sentido. Tamaños superiores a la capacidad máxima de memoria pueden provocar que dañe permanentemente la tarjeta, en cuyo caso la reposición de esta corre por su cuenta.
 - Documentese sobre el algoritmo de lectura BMP y piense cómo conjugará este algoritmo con el de escritura en pantalla, de modo de poder cumplir con el requisito de tiempo. *Pensar* antes de programar siempre resulta ser una buena práctica.
 - Separe por módulos o librerías cada funcionalidad importante del código. Ya dispone de librerías para lectura de sistema de archivos FAT y escritura en pantalla. Su comunicación USART sepárela en otra librería, y la comunicación SPI en otra. De la misma forma, sus algoritmos BMP no deberían estar insertos en el flujo del programa, si no que en una librería aparte, para luego llamar a las funciones respectivas en el flujo principal del programa.



4.3. Criterios de Distinción

La distinción representa un trabajo adicional que sobresale a los requerimientos mínimos para la aprobación. Agregados adicionales no constituyen por sí mismo una distinción si no representan un verdadero trabajo adicional de comprensión y/o análisis.

Asimismo, el trabajo primero debe cumplir los criterios de aprobación para ser un candidato a distinción. Es decir, trabajos que hayan dejado de funcionar por culpa de la extensión agregada al código con fin de distinguir son reprobados, por lo que puede presentarlo en un código aparte.

Los trabajos distinguidos pueden caer (no exclusivamente) en algunas de las siguientes líneas generales:

- Lectura de otro formato de imágenes más complejo, como PNG ó JPG, GIF.
- Despliegue de animaciones en pantalla.
- Despliegue de gráficos del estilo de un osciloscopio, utilizando un ADC para algún rango establecido (sensor de temperatura, luz, etc).
- Realizar la actividad en ambos microcontroladores. No se asusten y consideren esto realizable, pueden ser códigos apartes y no necesariamente portable.
- Presentación creativa.
- Chat entre microcontroladores mostrando el texto escrito en pantalla TFT, si lo presenta con un compañero, ambos tendrán su distinguido.
- Utilización del protocolo I2C para medir temperatura con el sensor TMP102. Esa información debe ser desplegada en la pantalla TFT, mostrando un log de los últimos datos adquiridos (con algún cierto intervalo de tiempo) o mostrar un gráfico del estilo de un osciloscopio.

Si usted utilizó una característica en un laboratorio anterior para lograr distinción, esta no vuelve a considerarse para una distinción en este laboratorio.