



Lab 7: SPI e I²C

Nivel 5

1. Introducción

Este laboratorio busca introducirlo al uso de nuevos protocolos de comunicación. Por una parte, está el conocido como *I²C* (Inter-Integrated Circuit) ¹ que se implementará entre distintos dispositivos. Para esto, se empleará el uso de dos componentes periféricos: un sensor de temperatura TMP102 y un conversor de señal digital a analógica MCP4725.

Por otra parte, se introducirá al uso de un nuevo protocolo de comunicación: *Serial Peripheral Interface*, más conocido como SPI. Para esto se hará uso de un *display* TFT, que posee un controlador integrado y *framebuffer* ST7735, el cual hace que el módulo TFT sea fácil de manejar por medio del microcontrolador.

El objetivo de este laboratorio es lograr la implementación de un pequeño sistema de comunicaciones, en que un dispositivo es el encargado de tomar datos y procesarlos, para distribuirlos entre los distintos componentes. Para lograr esto, la actividad se dividirá en 5 partes:

- Task 0: Habilitación de gráficos de prueba.
- Task 1: Habilitación de pantallas
 1. Task 1.1: Graficar datos de TMP102.
 2. Task 1.2: Graficar u enviar datos a un DAC.
 3. Task 1.3: Implementación de una pantalla de inicio.

Task 2: Implementación de menú de usuario

- Task 3: Mejorar el sistema
- Task 4: Salva pantallas

¹Tambien conocido por Two Wire Interface en algunos dispositivos



2. Evaluación

Al completar los requerimientos, este laboratorio será evaluado con nota 1.0 a 7.0. Existirá un apartado de bonus que serán décimas extra que se sumarán a la nota promedio de todos los laboratorios. La experiencia se separa en 5 actividades complementarias, que ayudarán a guiar el trabajo a desarrollar. Para la evaluación online se revisará hasta el Task que logró realizar.

3. Actividad a realizar

La actividad consiste en la implementación de un menú de usuario para controlar y desplegar la información de un TMP102 y un DAC en la pantalla TFT. Harán uso de 3 protocolos de comunicación, UART, I2C y SPI. Es una actividad que engloba gran parte de los contenidos aprendidos y busca prepararlos para el proyecto.

La actividad está pensada para realizarse en su totalidad con el ATmega328P, aunque está la libertad de implementarse en MSP.

3.1. Task 0: Habilidad de gráficos de prueba (0 %)

En esta primera actividad usted deberá rellenar las funciones de SPI entregadas en el repositorio de este laboratorio. Con estas, debe ser capaz de visualizar las distintas animaciones del código entregado. Esto le permitirá observar de forma gráfica que funciones tiene y cómo poder implementarlas en las siguientes actividades.

Importante: Esta actividad no es evaluada, es introductoria.

3.2. Task 1, Total (50 %)

En esta actividad deberá realizar la implementación de distintas interfaces que permitirán controlar cada uno de los componentes. Se sugiere realizar cada una en códigos separados, para así tener respaldo de lo realizado y después dedicarse a unir las partes importantes.

3.2.1. Task 1.1 Comunicación TMP102 (10 %)

Para esta actividad deberá utilizar el sensor TMP102. Se le han proporcionado distintos archivos que deberá modificar para poder llevar a cabo la comunicación con este dispositivo.

Como podrá observar, se tiene el archivo `i2c_temperature.c`, que contiene la estructura lista para poder realizar la comunicación con el sensor TMP y poder enviar los datos mediante UART.



- Investigue acerca del protocolo I^2C ,
 - ¿Cómo se realiza la comunicación? ¿Qué son los *masters* y *slaves*?
 - ¿Qué valores se necesitan para poder entablar la conexión con un dispositivo?
 - ¿Qué son las condiciones de **Start**, **Repeated Start** y **Stop**?
 - ¿Cómo se puede leer o escribir desde un dispositivo? ¿Cómo se le dice que tipo de información quieres leer?
 - Investigue acerca de las distintas velocidades que los dispositivos soportan. En general es de 100 kHz a 400kHz, por lo que utilizar el más rápido es lo deseable. Actualmente hay pocos componentes que soportan modos más lentos de comunicación.
 - Lea el datasheet del TMP102 para poder comprender su funcionamiento.
- Complete los archivos dentro de la carpeta USART con las funciones ya utilizadas en laboratorios anteriores. Esto le ayudará a corroborar una correcta conexión con el TMP102.
- Posterior a lo anterior, deberá completar las funciones de los archivos que se encuentran dentro de la carpeta I2C.
- De realizar los pasos anteriores de forma correcta, deberá ser capaz de poder medir la temperatura.
- Modifique la lectura de la temperatura para medir con mayor precisión. Actualmente el código trunca todos los decimales, por lo que usted deberá incorporar al menos 1 decimal en la lectura.
- Corroborado lo anterior, deberá incorporar el envío de datos a la pantalla TFT. Para esto debe realizar un pequeño gráfico de Temperatura vs Tiempo. El rango de temperatura a mostrar y el tiempo de muestreo lo puede indicar usted, la idea es que sea posible visualizar un gráfico simple con esta información. Puede guiarse de la siguiente imagen:

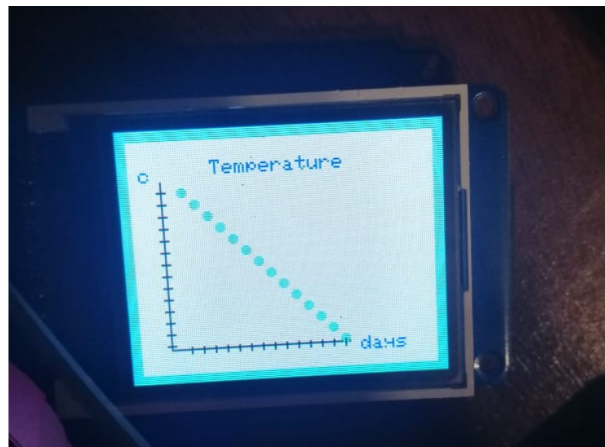


Figura 1: Ejemplo de interfaz

Advertencias

El sensor de temperatura TMP102 funciona con un voltaje máximo de 3.3V. NO intente alimentarlo con valores más altos. Es necesario conectar el pin VCC de este a una alimentación de 3.3V. Las líneas SDA y SCL, están llevadas al riel de VCC con las resistencias de pull-up de los dispositivos en la línea, por lo que su valor no debería alcanzar voltajes más altos. Sin embargo, en caso de configurar estos pines como outputs en su microcontrolador podría accidentalmente suministrar 5.0V a esta línea. Es por esto que **SDA y SCL deben ser configurados como inputs mientras se este conectado a TMP102.**

3.2.2. Task 1.2: Comunicación DAC (MCP4725) (20 %)

En esta actividad deberá establecer una comunicación con el DAC MCP4725 para poder generar una senoide en un LED.

- Para esto necesitará toda la preparación ya realizada en el *Task 1*.
- Revise el datasheet del DAC entregado y observe el código que se encuentra en el archivo `i2c_DAC.c`.
- Este es un DAC de 12 bits, piense en cómo enviar los datos necesarios para poder entregar la información requerida y lograr que funcione.



- Encuentre una forma básica de establecer comunicación con este dispositivo. Es útil realizar un diagrama de flujo del programa para poder programarlo de forma adecuada.
- Conecte un LED o un osciloscopio para ver la salida que se produce con este DAC.
- Incorpore la opción de setear un valor fijo del DAC con una instrucción simple de UART. A modo de ejemplo, si recibe por UART la frase:

V 2.54

Debe ser capaz de generar una salida de 2.54 V en el DAC. Para esta ocasión solo se solicitará un decimal de precisión y la salida se corroborará al mostrar el código. Es su deber decidir cómo hará la conversión y de qué forma se recibirá el valor.

- Una vez logrado lo anterior, incorpore un pequeño gráfico en la pantalla TFT que permita mostrar un dibujo de una senoide o de una línea recta que indica el valor fijo enviado por UART. Idee una forma de alternar en cada modo, puede ser con botón o enviando un mensaje por UART. **La decisión de diseño queda en sus manos.**

El MCP4725 si soporta voltajes de 5.0V, sin embargo, como debe conectar ambos dispositivos a las líneas de comunicación, **MCP4725 debe ser alimentado con 3.3V también.**

3.2.3. Task 1.3 Pantalla Inicio (20 %)

En esta actividad deberá incorporar una pantalla de inicio en la TFT, que contenga 2 textos:

1. TMP102
2. DAC MCP4725

Y una tercera línea que indique la hora actual. Dado que su ATmega no tiene un RTC (Real Time Clock), al momento de iniciar el sistema, debe enviarle la hora actual (en formato HHMM) y desplegarla en pantalla.

Para que el funcionamiento sea lo más parecido a un reloj, este se deberá ir actualizando a medida que avanza la hora. Para la revisión se considerará lo siguiente:

- Lo mínimo a mostrar es horas y minutos.
- Para ver el avance al momento de la presentación, los minutos cambiarán cada 1 segundo (es un reloj de alta velocidad).
- Considere que los minutos cambian cada 60 minutos y use un formato de 24 hrs.



3.3. Task 2. Implementación de menú de usuario (20 %)

En este task deberá implementar todo el Task 1 en un solo código. Mediante el menú de usuario creado en el Task 1.3 se deberá poder acceder a las pantallas 1.1 y 1.2. La forma de elegir cada pantalla queda a su criterio, puede implementar el botón de usuario del AVR o un mensaje por UART.

Cabe señalar que también debe ser posible volver al menú principal desde las pantallas 1.1 y 1.2. **Esto debe realizarse sin tener que reiniciar el microcontrolador ni reiniciarse el reloj implementado.**

3.4. Task 3. Animar del sistema (20 %)

Al sistema realizado en el Task 2 se le deben incorporar nuevas funcionalidades. Para esto, debe incorporar el uso de animaciones que le den movimiento a las pantallas.

Se requiere de al menos 3 animaciones simples que se incorporen al sistema (En rojo se indica las que deben ir sí o sí). Algunos ejemplos pueden ser:

- Un cuadrado que se va moviendo al ir escogiendo alguna de las opciones.
- **Una animación que se encuentre en la interfaz.**
- Una animación que indique “OK” al momento de seleccionar una opción y que muestre una animación de “Cargando...” antes de pasar a la siguiente pantalla
- **Una animación asociada al reloj. Puede ser un cuadrado que se completa cada 10 “minutos”, un círculo que tarda 60 “minutos” en completarse. (Entendiendo que 60 minutos son 60 segundos en nuestro sistema).**
- Una línea que se completa de izq a derecha (o viceversa) al moverse entre las opciones.
- Sugerir y preguntar alguna animación en las Issues de GitHub.

3.5. Task 4. SalvaPantallas (10 %)

Debido a que su sistema está pensado para un usuario, debe reaccionar cuando este no se encuentra en el lugar. Para esto, si el sistema no recibe un input de usuario durante al menos 20 seg (ya sea input por UART o por algún botón) debe cargar un salvapantallas que debe cumplir las siguientes características.

- El salva pantallas solo se activa en la pantalla principal.



- Mostrar el reloj en un tamaño más grande, con el tiempo actual desde el momento que se configuró y que siga avanzando. Si se realiza un reloj digital, este debe incorporar alguna animación extra que permita mostrar que el tiempo está avanzando (más que solo mostrar los cambios de números.)
- El reloj puede ser analógico, en ese caso debe ser en formato 12 hrs. Una idea de como implementarlo, puede ser encontrado aquí <https://www.youtube.com/watch?v=3gFMphirZaE>.
- Incorporar alguna animación que esté en la pantalla, ya sea una línea que avanza por la pantalla o algún bloque que se rellene al ir avanzando el tiempo.

4. Recomendaciones y requerimientos

Dado que su sistema estará realizando muchas acciones al mismo tiempo, se recomienda considerar lo siguiente:

- El control de usuario es importante, ya que se estarán realizando muchas acciones al mismo tiempo. Por lo que el uso de interrupciones de Rx como interrupciones de pin es necesario.
- El uso de librerías es esencial y requerido. Por lo que no se aceptarán códigos que no tengan librerías para I2C, UART, SPI, TFT y si aplica, SD.
- Puede crear sus propias librerías si estima que le ayudarán a programar mejor.
- Utilicen funciones y máquinas de estados, ayudarán a separar mejor el problema a realizar.
- 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.
- 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.
- No intenten saltarse tasks. El mejor camino es ir avanzando de apoco e incorporar paso a paso lo que se solicita.



5. Lectura recomendada

- **ATmega328/P Complete Datasheet.**
- **MSP430x5xx and MSP430x6xx Family User's Guide**
- **MSP430F552x, MSP430F551x Mixed-Signal Microcontrollers datasheet.**
- Ayudantías y códigos subidos
- **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.



6. Rúbrica de Evaluación

6.1. Consideraciones generales

- Deben cumplir con el horario asignado. Tendrá un periodo de 40 min de revisión.
- 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.
- No se reciben trabajos después del módulo de presentación. Trabajos no entregados son calificados con nota mínima.
- Solo dispone del tiempo asignado para ser revisado, atrasarse y no cumplir hará que no sea posible revisarlo de forma correcta.
- Debe presentar hasta el Task que alcanzo a revisar, todo en un mismo código. No se considera completo si es que realizó códigos separados.
- **IMPORTANTE:** se prohibirá el uso de funciones de Arduino IDE y Energia.nu para la programación de las tarjetas de desarrollo, esto por la simplicidad que involucra, por lo que deberán mostrar que entienden qué están realizando al momento de programar.



6.2. Criterios de aprobación

6.2.1. Task 1-4

1. Funcionamiento de los requerimientos. El alumno realiza una presentación de su trabajo y se responsabiliza de exponer que su trabajo satisfaga todos los requerimientos mínimos solicitados en la *descripción de la actividad*, los cuales incluyen en este laboratorio:
 - Programa compilado y ejecutándose
 - Control de usuario. Debe justificar la elección de usar botón o mensajes por UART. Estos pueden ser implementados con o sin interrupciones.
 - Conexión de TMP102 y MCP4725 al mismo tiempo.
 - Cumplimiento de los requisitos indicados en cada Task.
 - Todo el código debe estar en solo 1 proyecto, no varios códigos.
2. Preguntas. Se responde satisfactoriamente preguntas aleatorias al momento de la presentación final, las cuales abarcan los siguientes temas:
 - Preguntas de código.
 - Funcionamiento del sistema del protocolo UART.
 - Funcionamiento del sistema del protocolo I2C.
 - Funcionamiento del sistema del protocolo SPI.
 - Funcionamiento de la pantalla TFT.
 - Uso de interrupciones en microcontroladores.
 - Uso de máquinas de estado
 - Decisiones de diseño de su sistema.



7. Bonus

Al cumplir con todos los puntos indicados en la **Rúbrica de Evaluación** obtendrá la nota máxima equivalente a 7.0. Contará con los siguientes bonus complementarios, que se sumarán al promedio de laboratorios del curso (Tope 4 décimas). Estos se pueden implementar en códigos aparte, ya que lo anterior podría ocupar gran parte de la memoria.

7.1. Mostrar imagen BMP alojada en SD (2 décimas)

https://youtu.be/MOQHcLSpl_Y

En este bonus, deberá reunir los conocimientos adquiridos en el *Task 0* y lo mostrado en ayudantía, 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 la imagen de su animal favorito con nombre `animal.bmp`, siguiendo el estándar que aprendió en el punto anterior.
- 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.

7.2. Cronómetro cuenta vueltas (1 décima)

En esta actividad, deberá implementar un cronómetro de la forma `00:00.00`, que sea capaz de mostrar el tiempo en minutos, segundos, décimas y centésimas en la pantalla. El tiempo

inicia cuando se presiona el botón de su microcontrolador y cada vez que es presionado nuevamente, va guardando el tiempo de vuelta (al menos 3). Puede utilizar el diseño que estime conveniente. La siguiente imagen puede servir de inspiración:

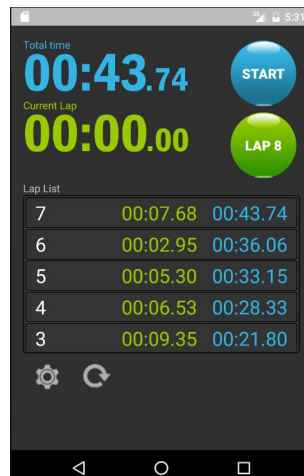


Figura 2: Fuente: https://play.google.com/store/apps/details?id=com.bagum.timeslot&hl=es_CL

7.3. Tic Tac Toe (2 décimas)

Implementar una versión sencilla del juego Tic Tac Toe, Tres en línea o Gato dependiendo del nombre que lo conozca <https://en.wikipedia.org/wiki/Tic-tac-toe>. La interfaz puede ser controlada por UART. La siguiente imagen es una idea de lo que se quiere implementar.

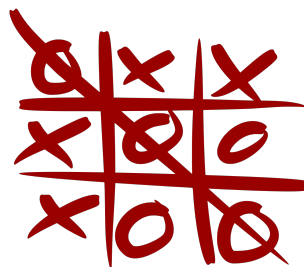


Figura 3: Fuente: Wikipedia



7.4. SEP Simon (4 décimas)

¿Recuerdan el Laboratorio 04 ? En este bonus la idea será recrear el juego implementado. Para simplificar la implementación, considere lo siguiente:

- Solo se usará un buffer lineal que sea de al menos largo 20.
- Consideren una implementación sencilla, puede ser con cuadros o círculos sin tener que hacer una animación compleja. La siguiente imagen puede ser de inspiración

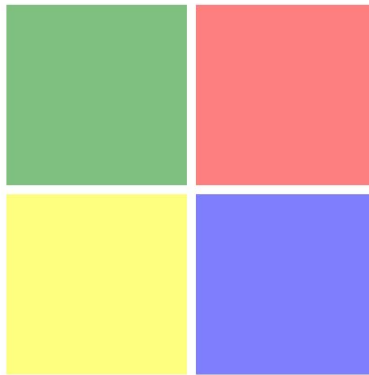


Figura 4: Sep Simon Game, fuente: <https://www.lawebdelprogramador.com/codigo/JQuery/3044-Juego-del-Simon.html>