

Objetivos actitudinales:

1. Promover el trabajo en equipo para obtener la solución a un problema.
2. Promover la habilidad de realizar una defensa de la solución propuesta para el problema planteado.
3. Promover la habilidad de elaborar un reporte escrito sobre el trabajo realizado.

Objetivos cognitivos:

1. Especificar el comportamiento del sistema utilizando el modelo de Máquina de Estado Finito / Statecharts UML.
2. Aplicar los conocimientos adquiridos sobre la arquitectura de la familia de microcontroladores KL46 para desarrollar una aplicación basada en la placa FRDM-KL46Z.
3. Aplicar los conocimientos adquiridos sobre protocolos off-board y los módulos UART del microcontrolador.
4. Diferenciar las tareas relacionadas a la detección de tramas del procesamiento de las mismas, en la solución propuesta
5. Utilizar las funciones de biblioteca suministrada por el fabricante para soportar el desarrollo de la aplicación software.
6. Aplicar el criterio de reutilización de código al definir la estructura del proyecto.
7. Aplicar los conocimientos adquiridos para el desarrollo de drivers de dispositivos y modificarlos para la utilización de DMA en circunstancias donde represente una ventaja.

Enunciado:

Este trabajo práctico consiste en la utilización de un microcontrolador para establecer comunicación serie con otro dispositivo.

Podrá optarse por:

Opción 1: RS485/UART0

En esta habrá comunicaciones en simultáneo:

- a) comunicarse por UART1/RS485 con con la PC (interactuar con el software provisto por la cátedra SD2_TP2_Master)
- b) comunicarse con la pc por UART0 enviando tramas periódicas de la aceleración en los 3 ejes a un software para animación en 3D
- c) se deberá mostrar en el display OLED los 3 valores de aceleración

Opción 2: solamente RS485

En este caso la comunicación con la pc será solo por UART1/RS485, tanto para interactuar con el software provisto por la cátedra como para el envío de las tramas periódicas de la aceleración en los 3 ejes a un software para animación en 3D

En este caso la selección, de una comunicación u otra, será **presionando SW1**. El display OLED indicará al usuario en qué modo se encuentra (SD2_TP3_Master o 3D).

Comunicación Maestro/Esclavo

La placa FRDM-KL46Z será el dispositivo esclavo en la comunicación, recibiendo las peticiones del maestro y dando la respuesta correspondiente a los diferentes mensajes, el dispositivo maestro será la computadora con un puerto RS485 conectado a uno de los puertos USB.

En la computadora se ejecutará un programa que enviará los mensajes hacia el esclavo y recibirá las respuestas para poder visualizarlas, este programa será suministrado por la cátedra.

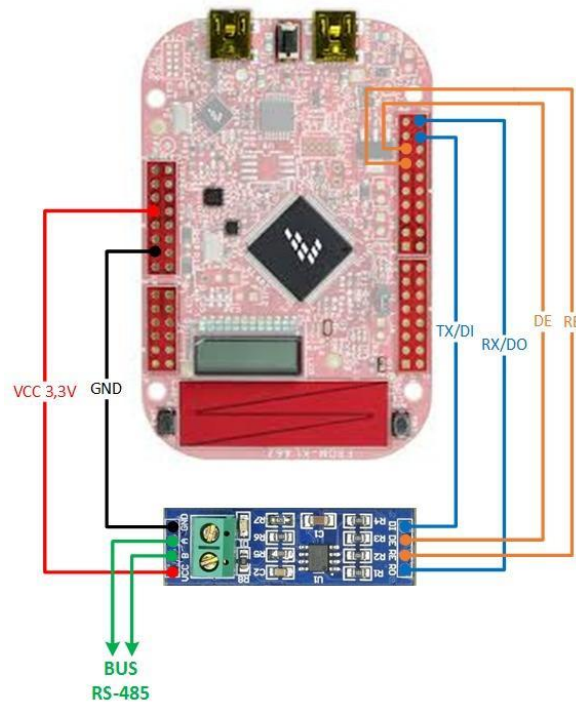
Todos los mensajes enviados y recibidos serán codificados en caracteres ASCII y comenzarán con el carácter ':' (0x3A), y terminaran con el carácter LF (0x0A).

Los posibles mensajes se detallan a continuación:

En todos los casos XX representa un número de identificación que va desde "01" a "99" cada grupo utilizará el número de grupo asignado en la planilla de inscripción al TP.

Petición	Respuesta	Descripción
:XX01Y'LF'	:XX01Y'LF'	Acciones sobre el Led Rojo. Valores posibles de Y:
<i>Ejemplo:</i> :0201E'LF'	<i>Respuesta:</i> :0201E'LF' (igual a la petición)	E: encender A: apagar T: toggle
:XX02Y'LF'	:XX02Y'LF'	Acciones sobre el Led Verde. Valores posibles de Y:
<i>Ejemplo:</i> :0202E'LF'	<i>Respuesta:</i> :0202E'LF' (igual a la petición)	E: encender A: apagar T: toggle
:XX11'LF'	:XX11Y'LF'	Leer estado del SW1. Valores posibles de Y:
<i>Ejemplo:</i> :0211'LF'	<i>Respuesta:</i> :0211P'LF'	N: pulsador no presionado P: Pulsador presionado
:XX13'LF'	:XX13Y'LF'	Leer estado del SW3. Valores posibles de Y:
<i>Ejemplo:</i> :0213'LF'	<i>Respuesta:</i> :0213P'LF'	N: pulsador no presionado P: Pulsador presionado
:XX21'LF'	:XX21SXXXSYYYSSZZZ'LF'	Transmitir el valor de los 3 ejes de aceleración en centésimas de G. "S" es el signo, puede ser '+' o '-'.
<i>Ejemplo:</i> :0121'LF'	<i>Respuesta:</i> :0121+017-002+101'LF'	

Conexionado FRDM-RS485



Requerimientos de implementación:

- El programa deberá ser estructurado en diferentes archivos de manera que resulte fácil de comprender el rol de cada uno en base a las diferentes máquinas de estado que se implementen y los diferentes drivers de periféricos.
- La **recepción** de datos se debe realizar mediante **buffer circulares**.
- La **transmisión** de datos se debe realizar mediante **DMA** sin intervención del procesador, el procesador solo interviene en preparar el mensaje y al finalizar toda la transmisión de los mismos.
- El programa debe responder perfectamente a todos los mensajes presentados y funcionar en modo “**Comunicación Continua**”

Recomendaciones para el desarrollo del programa:

- Reflejar el modelo de funcionamiento basado en el modelo de Máquina de Estado Finito (MEF) en la codificación de la solución propuesta.
- Reutilizar, de ser posible, las rutinas desarrolladas previamente.
- Documentar el código implementado.
- Se sugiere hacer la implementación de la solución para la UART0, que está conectada al OpenSDA, el cual crea un puerto COM virtual en la computadora, lo que facilita las

pruebas del programa, una vez que el protocolo funciona correctamente, modificar el código para que utilice la UART1 (conectada al conversor RS485).

- Para verificar el correcto funcionamiento con el puerto RS485 se puede utilizar una segunda placa FRDM-46Z y resolver el problema 7 de la plancha de problemas 3, el cual implementa un conversor USB-RS485 y conectar ésta a la PC. O utilizar un conversor USB-RS485 comercial.
- Todas las tramas comienzan con ':' (0x3A), y terminan con el caracter LF (0x0A), por lo que se sugiere abordar la solución dividiendo en subproblemas, uno para detección de tramas y otro para interpretación de las mismas.

Variante Comunicación por enlace inalámbrico

- En caso de disponer de módulos APC230, reemplazar los módulos RS485 por un transceptor conectado a la placa KL46 y otro conectado a la PC con el adaptador USB. **IMPORTANTE:** Los módulos se alimentan en 3,3V. Chequear la polaridad ya que de realizar una mala conexión se pueden dañar de manera irreversible.

Visualizador de Aceleraciones en Modelo 3D

Descargar el IDE Processing desde el siguiente enlace: <https://processing.org/download>

Ejecutar el código adjunto: "Codigo_Processing_IDE_3D_Animation"

Desde la placa de desarrollo se deberán enviar los datos de aceleraciones de la siguiente forma:

X Y Z (números enteros, separados por un espacio) y al finalizar, un LF, ejemplo:

-2 21 100'LF'

Para no sobrecargar el puerto se deberá esperar al menos 100 ms para volver a enviar otro conjunto de aceleraciones.

En la línea del código Processing:

```
sp = new Serial(this, "COM1", 9600);
```

Se podrá renombrar el puerto COM utilizado y modificar el Baudrate (por defecto 9600).

