



Facultad Regional Tucumán
Departamento Electrónica

Técnicas Digitales II

Actividad de Formación Práctica 4 -bis-

**Tema: Uso de la UART con STM32CubeIDE, programación de
microcontroladores.**

Profesor:
Ing. Rubén Darío Mansilla

ATTP:
Ing. Lucas Abdala

vencimiento:
29 de noviembre de 2024

Año: 2024

Índice

1. Objetivos	4
2. Generalidades	4
3. Consignas para desarrollar	5
4. Bibliografía y documentación	6

Registros de cambios

Revisión	Detalles de los cambios realizados	Fecha
0	Creación del documento	31 de octubre de 2024
1	Se agrega bibliografía sobre la HAL de STM32	1 de noviembre de 2024

1. Objetivos

- Que el alumno logre implementar un módulo de software sencillo para trabajar con la *UART* utilizando el *IDE* STM32CubeIDE.
- Que el alumno desarrolle un *driver* de funciones para uso de la *UART*.
- Que el alumno aplique este *driver* en la implementación de aplicaciones desarrolladas con el IDE que estamos utilizando.

2. Generalidades

Esta actividad de formación práctica es del tipo **práctica de laboratorio**.

Cómo desarrollar esta actividad:

- Esta actividad debe desarrollarse siguiendo las consignas del punto 3.
- Se debe tomar como base la copia del último proyecto desarrollado en la práctica 4, si este incluye el módulo *USART3*. Si no es el caso se debe crear un nuevo proyecto desde cero que incluya el módulo *USART3*.
- Se debe implementar un módulo de software que maneje la *UART* que se utilizará en esta práctica, implementando las funciones necesarias para el manejo de la *UART*. Para esto se deben crear los archivos **API_uart.c** y **API_uart.h**, ubicando las declaraciones de las funciones prototipos y sus implementaciones donde corresponda.
- Una vez probadas las aplicaciones siguiendo las indicaciones del punto 3, se las ubicará en el repositorio grupal para su presentación. Todas las Apps deben presentar una coherencia en cuanto al formato del nombre del mismo y de las funciones desarrolladas para este.
- Se debe utilizar el repositorio grupal creado en GitHub para la presentación de las actividades de formación práctica:
 - Se debe crear una carpeta para cada actividad de formación práctica.
 - Se deben ubicar en la carpeta correspondiente, en este repositorio, las aplicaciones desarrolladas para esta actividad de formación práctica respetando el siguiente formato de nombre: **App_1_Y_Grupo_X_2024**.

Nota sobre el formato de nombre de la carpeta: **X** es el número de su grupo y **Y** es el número de aplicación como figura en el enunciado.

- Se debe insertar el link al repositorio de GitHub al final del informe que se presenta con esta actividad de formación práctica.
- El archivo que contiene el desarrollo del informe de la actividad de formación práctica debe ser de tipo pdf y su nombre debe respetar el siguiente formato:

AFP_4_Grupo_X_TDII_2024.pdf

- Se debe realizar la presentación de las aplicaciones, en el laboratorio de Sistemas Digitales, funcionando según lo que pide la consigna, con su correspondiente defensa de lo desarrollado. Fecha a acordar. Cada alumno debe presentar y defender la aplicación que ha desarrollado para esta actividad.

Nota 1: Para el desarrollo de las aplicaciones de esta actividad de formación práctica se utilizará el entorno de desarrollo integrado STM32CubeIDE para aplicar sobre una placa de desarrollo STM32-NUCLEO-F4XX-YY.

Nota 2: Las placas de desarrollo sugeridas para las practicas son: STM32-NUCLEO-F401-RE, STM32-NUCLEO-F412-ZG, STM32-NUCLEO-F413-ZH ó STM32-NUCLEO-F429-ZI.

Nota 3: El desarrollo de las actividades de formación práctica se realizará y presentará de manera grupal.

3. Consignas para desarrollar

1. Tome como base un proyecto nuevo, para lo cual parta de una copia del ultimo proyecto desarrollado para la practica 4, que ya tiene la estructura de los *drivers* **API_Delay** y **API_debounce** para implementar las funciones necesarias, para el uso de la (*UART*), en el archivo fuente **main.c** con su correspondiente archivo de cabecera **main.h**. Asegúrese que en **main.c** este incluido el módulo *USART3*, para esto, vea la parte de declaración de prototipos de funciones y asegúrese que esté la línea:

```
static void MX_USART3_UART_Init(void);
```

como se ve en la figura 1.

```
50
51 /* Private function prototypes -----
52 void SystemClock_Config(void);
53 static void MX_GPIO_Init(void);
54 static void MX_USART3_UART_Init(void);|
55 static void MX_USB_OTG_FS_PCD_Init(void);
56 /* USER CODE BEGIN PFP */
57
```

Figura 1. Sección de declaración de prototipos de funciones en main.c.

Si no está incluido el módulo *USART3*, que es el que utiliza la placa para comunicarse con el ordenador vía USB y transferir datos entre este y la placa, es recomendable crear un proyecto nuevo e incluir este módulo para poder utilizarlo.

Una vez creado el nuevo proyecto con las consideraciones del punto anterior proceda a implementar un módulo de software en un archivo fuente **API_uart.c** con su correspondiente archivo de cabecera **API_uart.h** y ubicarlos en el proyecto dentro de las carpetas */drivers/API/src* y */drivers/API/inc*, respectivamente.

- 1.1. En **API_uart.h** se deben ubicar las declaraciones de los prototipos de las siguientes funciones públicas:

- *bool_t* **uartInit()**;
- *void* **uartSendString**(*uint8_t* * *pstring*);
- *void* **uartSendStringSize**(*uint8_t* * *pstring*, *uint16_t* *size*);
- *void* **uartReceiveStringSize**(*uint8_t* * *pstring*, *uint16_t* *size*);

- 1.2. En **API_uart.c** se debe ubicar los prototipos de las funciones privadas y la implementación de todas las funciones del módulo, privadas y públicas.

Consideraciones para la implementación:

- 1) **uartInit()** debe realizar toda la inicialización de la *UART*. Adicionalmente, debe imprimir por la terminal serie un mensajes con sus parámetros de configuración. La función devuelve:
 - *true*: si la inicialización es exitosa.
 - *false*: si la inicialización no es exitosa.
- 2) **void uartSendString(uint8_t * pstring)** recibe un puntero a un *string* que se desea enviar completo por la *UART* (hasta el caracter '\0') y debe utilizar la función de la *HAL* **HAL_UART_Transmit(...)** para transmitir el *string*.
- 3) **void uartSendStringSize(uint8_t * pstring, uint16_t size)** recibe un puntero a un *string* que se desea enviar por la *UART* y un entero con la cantidad de caracteres que debe enviar. La función debe utilizar **HAL_UART_Transmit(...)** para transmitir el *string*.

Una vez creadas estas funciones, haga las pruebas básicas necesarias en **main.c** para verificar que estas se comportan como se pide en el punto anterior. Compile el proyecto y verifique que funciona como corresponde.

2. Cree un nuevo proyecto como copia del proyecto con la implementación del punto 1 de la práctica anterior (Práctica 4), si este incluye la utilización del módulo *USART3* como se aclaró mas arriba. Si ese proyecto no incluye el módulo *USART3* use como base la copia del proyecto del punto anterior de esta práctica y copie la implementación del punto 2 del proyecto de la practica 4 para tomar como base de desarrollo del nuevo punto.
3. Sobre el ejercicio de la práctica anterior (práctica 4), implementar una aplicación que utilice el módulo de *UART* para informar al usuario cuando se producen los flancos ascendente y descendente al presionar y soltar el pulsador.
4. Compile, y realice el *debug* y la programación de la placa de desarrollo para verificar que la aplicación se comporta como debería hacerlo, según indica la consigna anterior.
5. Desarrolle un breve informe que contenga los siguientes ítems:
 - 5.1. **Introducción al tema:** Todo lo que vio acerca de la *UART* y la implementación de funciones para su utilización en el *IDE* STM32CubeIDE.
 - 5.2. **Aplicaciones desarrolladas:**
 - **Aplicación:** Nombre de la aplicación desarrollada.
 - **Autor o autores de la aplicación:** Apellido y nombres de alumnos que la desarrollaron.
 - **Observaciones:** sobre lo realizado en esa aplicación. En este campo puede compartir su experiencia en el desarrollo, los problemas que pudo haber enfrentado y como los solucionó. Recomendaciones si las hubiera.
 - 5.3. **Link al repositorio grupal:** en el que se encuentran las aplicaciones desarrolladas para esta actividad de formación práctica.

4. Bibliografía y documentación

- Documentación accesible desde el IDE en:

Help → Information Center → STM32CubeIDE Manuals

- Documentación de STM sobre las placas de desarrollo Núcleo F4XX:
 - [MB1137-EsquemáticoPlacaSTM-F4XX.pdf](#)
 - [UM1974 - STM32 Nucleo-144 board User Manual.pdf](#)
- Documentación de STM sobre las funciones y organización de la HAL:
 - [UM1725 - Descripción de la HAL STM32](#)
- Bibliografía sobre programación en STM32
 - [Donald Norris - Programming With STM32_ Getting Started With the Nucleo Board and C_C++-McGraw-Hill Education \(2018\)](#)
 - [Mastering STM](#)
- Documentación desarrollada por terceras partes:
 - [UART en STM.pdf](#)