

# Laboratorio #3

Laura Camila Blanco Gómez  
*Escuela de Ingeniería*  
Universidad Sergio Arboleda  
Bogotá, Colombia  
laura.blanco01@usa.edu.co

Santiago Cáceres Linares  
*Escuela de Ingeniería*  
Universidad Sergio Arboleda  
Bogotá, Colombia  
santiago.caceres01@usa.edu.co

## I. INTRODUCCIÓN

El objetivo principal de este laboratorio es diseñar e implementar un sistema capaz de calcular y mostrar la frecuencia de una señal generada por una onda cuadrada. Para lograr esto, utilizaremos contadores, temporizadores, interrupciones y cálculos de frecuencia. Además, presentaremos visualmente la frecuencia calculada en una pantalla LCD, y agregaremos una barra de progreso que ilustrará el valor relativo de la frecuencia actual en relación con un valor máximo posible por cada franja de valor.

## II. MARCO TEÓRICO

Como recapitulación de ciertos de los tópicos tratados durante el laboratorio se parte para el correcto entendimiento de algunos conceptos mencionados en la sección anterior.

**Frecuencia:** La frecuencia se refiere al número de ciclos completos de una señal que ocurren en un segundo. Se mide en hercios (Hz). En electrónica y telecomunicaciones, la frecuencia se utiliza para describir la velocidad de oscilación de una onda, como una señal eléctrica o una señal de radio.

La frecuencia tiene una relación inversa con el concepto de longitud de onda (distancia entre dos picos) de tal manera que la frecuencia es igual a la velocidad de desplazamiento de la onda dividida por la longitud de onda.

**Periodo:** El periodo de una señal periódica se refiere al tiempo que tarda en completar un ciclo completo. Se mide en segundos y es el inverso de la frecuencia. Matemáticamente, se puede calcular dividiendo 1 entre la frecuencia. Por ejemplo, si una señal tiene una frecuencia de 100 Hz, su periodo sería de 1/100 segundos, es decir, 0.01 segundos. El periodo es importante en la generación de señales periódicas y en el cálculo de tiempos en sistemas de temporización.

**Prescaler:** Un prescaler es un divisor de frecuencia utilizado en muchos sistemas electrónicos, especialmente en microcontroladores y microprocesadores. Su función principal es reducir la frecuencia de un reloj de entrada a una frecuencia más baja antes de que se aplique a un contador o a otro componente. Esto permite trabajar con frecuencias más manejables y precisas. La frecuencia del reloj a la salida del prescaler sigue la fórmula de la ecuación 1, por lo que el valor a escribir en el prescaler vendrá dado por la ecuación 2.

$$CK_{CNT} = \frac{CK_{PSC}}{TIMxPSC + 1}$$

Figure 1: Frecuencia de salida del prescaler.

$$TIMxPSC = \frac{CK_{CNT}}{CK_{PSC}} - 1$$

Figure 2: Cálculo del valor del prescaler.

Donde:

- CK\_PSC es la frecuencia de entrada al prescaler (Clock Prescaler)
- CK\_CNT es la frecuencia de salida del prescaler (Clock Counter)
- TIMx\_PSC es el valor contenido en el Prescaler Register

**Counter period:** El counter period, o periodo del contador, se refiere al tiempo necesario para que un contador complete un ciclo completo de conteo. En sistemas electrónicos, los contadores se utilizan para contar eventos o para generar señales de temporización. El periodo del contador determina la frecuencia a la que se cuentan los eventos o se generan las señales de temporización.

**Pantallas LCD:** Las pantallas LCD (Liquid Crystal Display, por sus siglas en inglés) son dispositivos de visualización utilizados en una amplia variedad de aplicaciones, como teléfonos móviles, televisores, relojes y paneles de instrumentos. Estas pantallas utilizan cristales líquidos y retroiluminación para mostrar información de manera visual. Proporcionan una interfaz gráfica o numérica para interactuar con dispositivos electrónicos.



CP (counter period) de 770, esto se realizó para poder ejecutar una IRS cada 500ms.

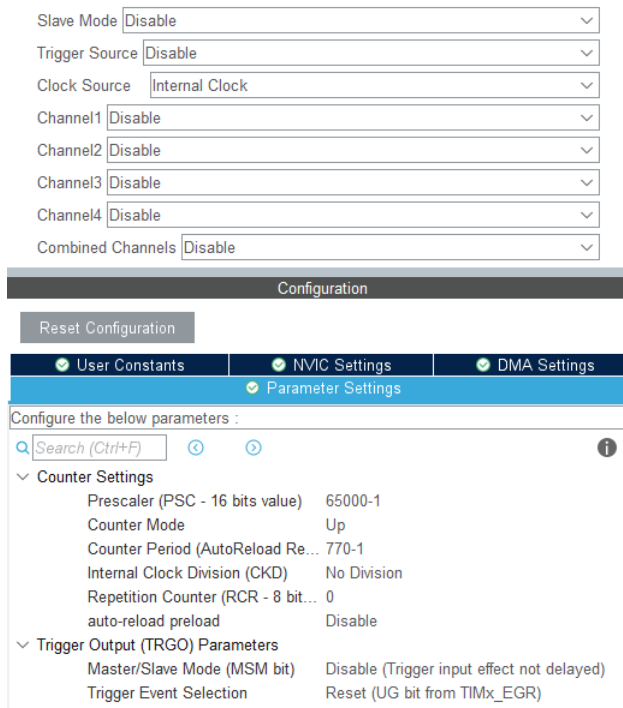


Figure 6: Configuración TIM1.

Finalmente, se configuró la interrupción tanto en el TIM como en el NVIC, la IRS seleccionada fue la de actualización [Update Interrupt] como se ve a continuación.

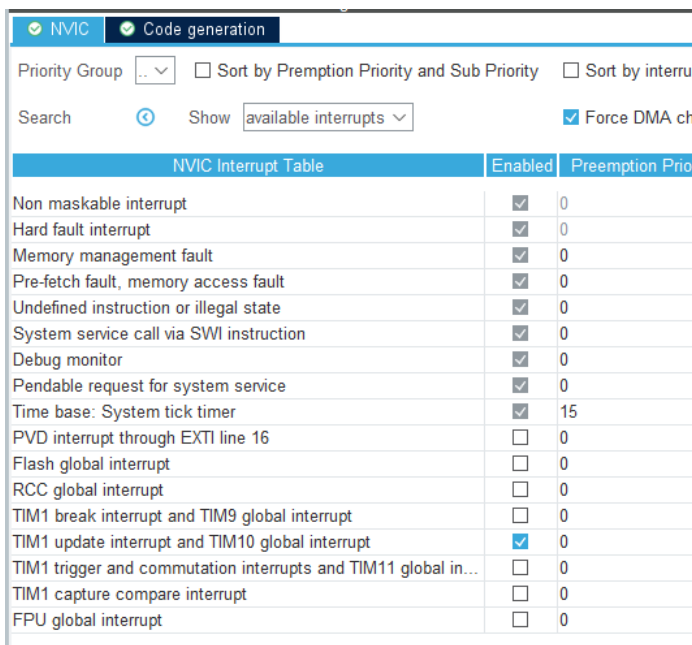


Figure 7: Configuración NVIC.

## • Código

Para el apartado del código, este se dividirá en 4 partes, las cuales serán.

### – Declaración de variables

En este apartado se observa que se tiene variables con el modificador **volatile** esto se realizó con el fin de evitar la optimización de estas variables por parte del IDE, además de las variables con ese modificador se crea un char de tamaño 150, esta variable se declaró para poder almacenar el texto de la frecuencia con su valor.

```
1 #include "main.h"
2 #include "lcd.h"
3 #include "math.h"
4 volatile uint32_t cont = 0;
5 volatile uint32_t cont2 = 0;
6 volatile uint32_t frecuencia = 0;
7 volatile uint32_t frecuencia2 = 0;
8 volatile uint32_t porcentaje = 0;
9 volatile uint8_t flanco = 0;
10 char x[130];
```

### – IRS

Para este apartado se programa la IRS la cual nos ayudara a tomar el número de pulsos que han pasado en 500ms como se configuró en el TIM

```
1 void HAL_TIM_PeriodElapsedCallback
2 (TIM_HandleTypeDef *htim){
3     if(htim->Instance == TIM1){
4         cont2 = cont;
5     }
6 }
```

### – Método, cálculo, frecuencia y visualización de la información en la LCD

El cálculo de la frecuencia se realiza con la siguiente fórmula,  $f = \frac{n}{t}$  donde **n** es el número de pulsos transcurridos en 500ms y **t** siendo el tiempo transcurrido el cual es 500ms debido a que la IRS se ejecuta cada vez que transcurre este tiempo. Después de realizar el cálculo se procede a verificar que valor tiene la frecuencia debido a que en este laboratorio se manejan diferentes rangos, los cuales son

$$\begin{bmatrix} f, & \geq 0 & \leq 999 \\ f, & \geq 1000 & \leq 999999 \\ f, & \geq 1000000 \end{bmatrix}$$

Después de mirar en que rango se encuentra la frecuencia se guarda este valor en otra variable, finalmente se calcula el porcentaje que ocupa la frecuencia en la barra de progreso, esta fórmula es la siguiente porcentaje =  $\frac{16 * f}{vr}$  donde **f** es la frecuencia y **vr** el valor del rango en su valor máximo [999, 999999 y 4000000], después de esto se concatena el valor de la frecuencia con un texto por medio de la función **sprintf** donde x es la variable en la cual se va a almacenar el resultado de la concatenación. Posteriormente, se pasa x a la función de la librería

**lcd.c** la cual nos permite manejar la pantalla LCD 16x2.

Finalmente, hacemos un ciclo en el que mientras el iterador **i** sea menor al porcentaje calculado se va a poner una línea en la barra de progreso, cuando esto ya no se cumpla ya no se pintarán más líneas.

```

1 void calcularFrecuencia(void){
2     frecuencia = (cont2 / ((float) 0.5));
3     if (frecuencia != 0) {
4         if (frecuencia < 1000) {
5             frecuencia2 = frecuencia;
6             porcentaje = (int) ((16 * ...
7                 frecuencia2) / 999);
8             sprintf(x, "f[Hz]: %d", ...
9                 frecuencia2);
10            lcd_setCursor(&lcd, 0, 0);
11            lcd_print(&lcd, x);
12        } else if (frecuencia >= 1000 ...
13            && frecuencia <= 999999) {
14            frecuencia2 = (int) ...
15                round(frecuencia / 1000);
16            porcentaje = (int) ((16 * ...
17                frecuencia2) / 999000);
18            sprintf(x, "f[Khz]: %d", ...
19                frecuencia2);
20            lcd_setCursor(&lcd, 0, 0);
21            lcd_print(&lcd, x);
22        } else if (frecuencia > ...
23            999999) {
24            frecuencia2 = (int) ...
25                round(frecuencia / ...
26                1000000);
27            porcentaje = (int) ((16 * ...
28                frecuencia2) / 4000000);
29            sprintf(x, "f[MHz]: %d", ...
30                frecuencia2);
31            lcd_setCursor(&lcd, 0, 0);
32            lcd_print(&lcd, x);
33        }
34    }
35    for (int i = 0; i < 16; i++) {
36        if (i < porcentaje) {
37            lcd_setCursor(&lcd, i, 1);
38            lcd_print(&lcd, "-");
39        } else {
40            lcd_setCursor(&lcd, i, 1);
41            lcd_print(&lcd, " ");
42        }
43    }
44    cont = 0;
45    cont2 = 0;
46 }

```

#### – Main

En este apartado se mira en que estado se encuentra el pin **PB5** mientras se encuentre en estado **GPIO\_PIN\_RESET** significa que la señal tiene un flanco de bajada, después de aumentar el contador de pulsos el flanco se pone en 0 para que mientras se llega al flanco de subida el contador no siga aumentando. Finalmente, cuando se activa la IRS el valor de **cont2** es el número de pulsos de los flancos de bajada, se llama a la función que calcula la frecuencia y la muestra.

```

1 while (1){
2     if (HAL_GPIO_ReadPin(GPIOB, ...
3         GPIO_PIN_5) == ...
4         GPIO_PIN_RESET) {
5         if (flanco == 0) {
6             cont++;
7             flanco = 1;
8         }
9     } else {
10        flanco = 0;
11    }
12    if (cont2 > 0) {
13        calcularFrecuencia();
14    }
15 }

```

Para finalizar en la figura 8 se muestra un modelo de montaje electrónico realizado para la práctica de laboratorio.

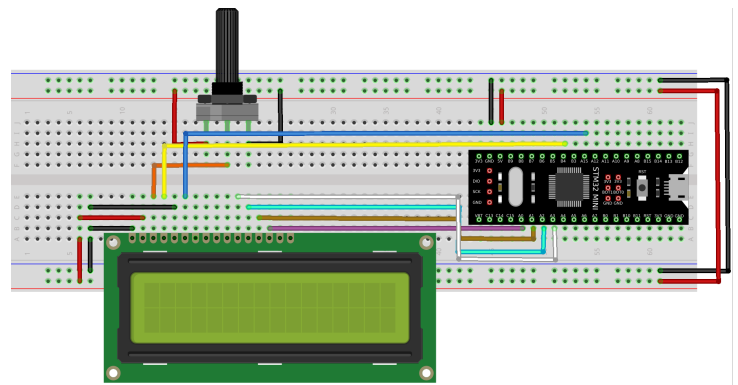


Figure 8: Montaje del circuito.

#### IV. RESULTADOS Y ANÁLISIS DE RESULTADOS

##### • Función calcularFrecuencia

La función **calcularFrecuencia()** calcula la frecuencia a partir del número de pulsos, realiza conversiones de unidades (Hz, kHz, MHz) según sea necesario y muestra la frecuencia en la pantalla LCD, además de hacer un gráfico de barra en el LCD. Finalmente, la frecuencia medida se muestra en la primera línea del LCD, y el gráfico de barras que representa el porcentaje de la frecuencia se muestra en la segunda línea del LCD.

##### • IRS y main

Estos dos apartados son los más importantes sin contar la función que calcula la frecuencia. Por el lado del main gracias a hacer uso del **flanco** se evita tener que calcular el número de pulsos porque solo se toma el tiempo cuando se llega al flanco de bajada y no todo el tiempo desde que llega al flanco de bajada hasta que llega el flanco de subida lo cual genera que la medición sea más precisa. Por la parte de la IRS nos permite tener una medida de tiempo constante con la cual se pueden llegar a realizar los cálculos.

## V. CONCLUSIONES

- Entre más cercano sé el valor de la frecuencia a los límites entre dos rangos, esto hará que el valor mostrado en la LCD fluctúa entre un rango u otro.
- Al hacer uso del operador logico ! se debe tener cuidado, ya que este operador comprueba si esta "false" no se hace igual que en una variable **booleana**, sino que esta validación se hace bit a bit.
- Al momento de mostrar información en la LCD se debe tener en cuenta que al sobrescribir un texto que sea más pequeño que el anterior los datos restantes se seguirán viendo, en este caso al tenerlo dentro del while no se podía eliminar el valor que se muestra en pantalla, ya que no permitía la visualización correcta de los datos, se soluciona este tema sobrescribiendo campos vacíos en los espacios no necesarios.
- El tiempo que se define para la interrupción debe ser un valor alto en tiempo, en este caso 0.5s, para que logre tomar uno o más pulsos de la onda de frecuencia sin importar el valor que se le enviara, entre menor sea el valor de la frecuencia el valor del tiempo que toma tener un pulso se alarga, por lo que al tener un tiempo de interrupción pequeño no lograría calcular el momento en el que el pulso se termina evitando así el adecuado cálculo de la frecuencia.

## REFERENCES

- [1] M. David. Lcd interfacing with microcontrollers tutorial - lcd commands and insturctionse. [Online]. Available: <http://www.matidavid.com/pic/LCD%20interfacing/commands.htm>
- [2] Prescaler. Universidad politica de valencia. [Online]. Available: <https://riunet.upv.es/bitstream/handle/10251/39538/ArticuloTimers.pdf?sequence=1>
- [3] Europa. Frecuencia. [Online]. Available: [https://ec.europa.eu/health/scientific\\_committees/opinions\\_layman/es/campos-electromagneticos/glosario/def/frecuencia.htm#:~:text=Frecuencia%20es%20la%20medida%20del,repite%20la%20onda%20por%20segundo](https://ec.europa.eu/health/scientific_committees/opinions_layman/es/campos-electromagneticos/glosario/def/frecuencia.htm#:~:text=Frecuencia%20es%20la%20medida%20del,repite%20la%20onda%20por%20segundo)
- [4] Wikipedia. Frecuencia. [Online]. Available: <https://es.wikipedia.org/wiki/Frecuencia>
- [5] Wiki. Pantalla de cristal líquido. [Online]. Available: [https://es.wikipedia.org/wiki/Pantalla\\_de\\_cristal\\_l%C3%ADquido](https://es.wikipedia.org/wiki/Pantalla_de_cristal_l%C3%ADquido)
- [6] CaliBeta. Lcd1602-stm32. [Online]. Available: <https://github.com/CaliBeta/LCD1602-STM32/blob/master/library/lcd.h>

## VI. ANEXOS