Laboratorio 2

Samuel Enrique Bejarano Soto [est.samuel.bejarano@unimilitar.edu.co](mailto:est.samuel.bejarano@unimilitar.edu.co) Juan David Medina [est.juan.medina@unimilitar.edu.co](mailto:est.juan.medina@unimilitar.edu.co) Andrés Felipe Bernal - 7003748 [est.andres.bernal1@unimilitar.edu.co](mailto:est.andres.bernal1@unimilitar.edu.co)

Profesor: Diego.

***Resumen:*** Este informe de laboratorio detalla una práctica donde se utilizó el microcontrolador STM32F767. Con acceso a todos los pines disponibles, se llevaron a cabo cuatro ejercicios distintos para abordar diversos aspectos de la programación y el uso de los pines. Los ejercicios cubrieron desde la configuración de puertos GPIO hasta la comunicación serial. A lo largo del informe, se describirán en detalle los procedimientos y los resultados obtenidos en cada ejercicio, destacando la importancia de la exploración de pines en el contexto de la STM32F767.

**Palabras clave:** Stm32F767 – Keil Uvision – leds – motor paso a paso – corriente – display.

**Abstract:** This laboratory report details an experiment where the STM32F767 microcontroller was utilized. With access to all available pins, four distinct exercises were conducted to address various aspects of programming and pin usage. The exercises ranged from configuring GPIO ports to serial communication. Throughout the report, the procedures and outcomes of each exercise will be described in detail, emphasizing the significance of pin exploration within the context of the STM32F767.

**Keywords**: Stm32F767 – Keil Uvision – leds – motor step by step – current – display.



1. INTRODUCCION.

En el mundo de la electrónica y la programación de sistemas, la STM32F767 ha surgido como una herramienta esencial que combina potencia y versatilidad en un solo microcontrolador. Este informe de laboratorio se adentra en el emocionante universo de la STM32, centrándose en su programación y aplicación práctica a través de una serie de ejercicios diseñados para expandir nuestra comprensión y habilidades en este campo.

Cada ejercicio aborda un aspecto específico de la programación y uso de la STM32F767, desde la configuración inicial hasta la interacción con periféricos como puertos GPIO, convertidores analógico-digitales (ADC) y comunicación serie. A medida que progresamos en el informe, examinaremos cómo los conceptos teóricos se traducen en la práctica, permitiéndonos comprender mejor cómo la programación eficiente de la STM32F767 puede desbloquear un mundo de posibilidades.

La relevancia de este informe radica en la creciente demanda de profesionales que pueden trabajar con microcontroladores avanzados. Además, el informe busca fomentar la comprensión de cómo la programación y la experimentación práctica se combinan para fortalecer nuestra competencia en la resolución de problemas de electrónica.

A medida que nos sumergimos en los ejercicios planteados en la guía, descubriremos la utilidad y la versatilidad de la STM32F767, y cómo su programación puede abrir puertas a la innovación en una amplia gama de aplicaciones.

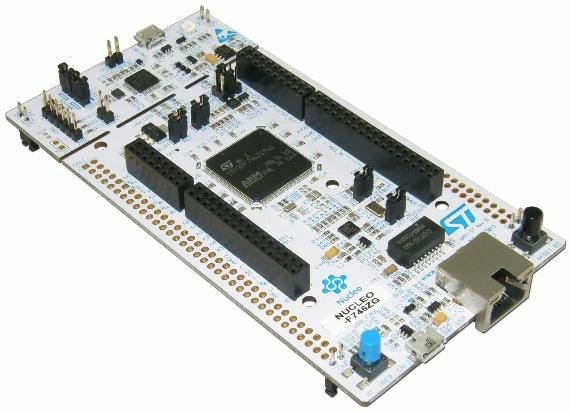
1. MARCO TEORICO.

## STM32:

El STM32 es una familia de microcontroladores de 32 bits desarrollados por la compañía STMicroelectronics. Estos microcontroladores ofrecen una amplia variedad de características y opciones de conectividad para una amplia gama de aplicaciones. Son muy populares en el

campo de la electrónica, la robótica y la industria, gracias a su potencia y versatilidad.





**Imagen 1**. STM32

## KEIL:

Keil uVision es un entorno integrado de desarrollo (IDE) utilizado principalmente para programar microcontroladores de la familia ARM, aunque también admite otros microcontroladores. Fue desarrollado por la compañía Keil, que fue adquirida por ARM en 2005 y ahora es parte de la empresa de software de desarrollo ARM. Keil uVision ofrece una plataforma completa para el desarrollo de proyectos de microcontroladores, que incluye herramientas de compilación, depuración, simulación y programación. Es una herramienta popular entre los desarrolladores de sistemas embebidos debido a su facilidad de uso y su amplio soporte para microcontroladores de diferentes fabricantes**.**

## RESISTENCIAS:

Una resistencia es un componente eléctrico pasivo diseñado para limitar el flujo de corriente eléctrica en un circuito. Su función principal es ofrecer resistencia al paso de la corriente, lo que regula la cantidad de corriente que fluye a través de ella y también afecta la caída de voltaje en el circuito.



**Imagen 2**. Resistencias.

## LEDS:

Un LED, o Diodo Emisor de Luz, es un dispositivo semiconductor que emite luz cuando pasa corriente eléctrica a través de él. Su capacidad para generar luz en diversos colores, su eficiencia energética y durabilidad lo convierten en una opción ampliamente utilizada en indicadores visuales, pantallas, iluminación y una variedad de aplicaciones electrónicas. Su tamaño compacto, encendido instantáneo y la posibilidad de controlar su intensidad hacen que los LEDs sean esenciales en la tecnología moderna.



**Imagen 3**. Leds

## MOTOR PASO A PASO:

Un motor paso a paso es un dispositivo electromecánico que convierte pulsos eléctricos en movimientos discretos o "pasos". Estos movimientos angulares se logran mediante la activación secuencial de bobinas electromagnéticas ubicadas en el motor. Cada pulso eléctrico hace que el motor avance un paso, lo que lo hace ideal para aplicaciones que requieren precisión en el control del movimiento, como impresoras 3D, máquinas CNC, sistemas de posicionamiento y robots, entre otros. La característica distintiva del motor paso a paso es su capacidad para moverse en incrementos predefinidos sin necesidad de retroalimentación, lo que lo hace muy útil en situaciones donde la posición precisa es esencial.



1. PROCEDIMIENTO.

Para esta practica se incio con la selección de los pines necesarios para cumplir con los requisitos de la practica, los cuales son 10 pines de salida para controlar la LCD, y tres pines de entrada para los botones de contador, descontador y seleccionador de frecuencia:

|  |  |  |
| --- | --- | --- |
| Salidas | Pin | Ubicacion |
| E | PD9 | CN11 - 69 |
| RS | PD8 | CN12 - 10 |
| d7 | PD7 | CN9 - 2 |
| d6 | PD6 | CN9 - 4 |
| d5 | PD5 | CN9 - 6 |
| d4 | PD4 | CN9 - 8 |
| d3 | PD3 | CN9 - 10 |
| d2 | PD2 | CN8 - 12 |
| d1 | PD1 | CN9 - 27 |
| d0 | PD0 | CN9 - 25 |

Tabla 1. Pines de control LCD

|  |  |  |
| --- | --- | --- |
| Entradas | Pin | Ubicacion |
| count- | PA1 | CN11 - 30 |
| count+ | PA0 | CN11 - 28 |
| frecuency | PB6 | CN12 - 17 |

Tabla 2. Pines de entrada para pulsadores

Una vez realizada esta asignación se hizo la configuración respectiva para cada pin, así como el moder, y el pupdr necesario para cada pin, además de la interrupción requerida para el cambio de frecuencia:

int main(){

SysTick\_Init();

//Configurar salidas

//Habilitar CLK

RCC->AHB1ENR |= (0x1<<3);

//MODER

GPIOD->MODER |= 0x55555;

//PUSH-PULL mode

GPIOD->OTYPER |= 0x0000;

//OSPEEDR

GPIOD->OSPEEDR |= 0xAAAAAAAA;

//No pull-up, no pull-down

GPIOD->PUPDR |= 0x00000000;

//Salidas configuradas

//Configurar entradas

//Habilitar CLK

RCC->AHB1ENR |= (0x1<<0);

//MODER

GPIOA->MODER |= (0x05);

//OSPEEDR

GPIOA->OSPEEDR |= 0xAAAAAAAA;

//PUPDR

GPIOA->PUPDR |= 0x05;

//Entradas configuradas

//EXTI PULSADOR PB6

RCC->AHB1ENR |= (0x1<<1);

RCC->APB2ENR |= (0x1<<14);

SYSCFG->EXTICR[1] |= 0x100;

EXTI->IMR |= (0x1<<6);

EXTI->RTSR |= (0x1<<6);

NVIC\_EnableIRQ(EXTI9\_5\_IRQn); //6

Teniendo esto pines se procedio a generar una librería .h que nos permitia controlar la LCD teniendo en cuenta únicamente dos tipos de comando, escritura y configuración, esta librería se encuentra adjunta en los códigos suministrados en la entrega.

Una vez podíamos mostrar y controlar la LCD, se procedio a conectar los pulsadores para asignar la lógica que nos permitia cumplir con los objetivos de la práctica, como el generar un numero aleatorio o que el numero mostrado solo varíe mientras este pulsado el botón de contar o de descontar.

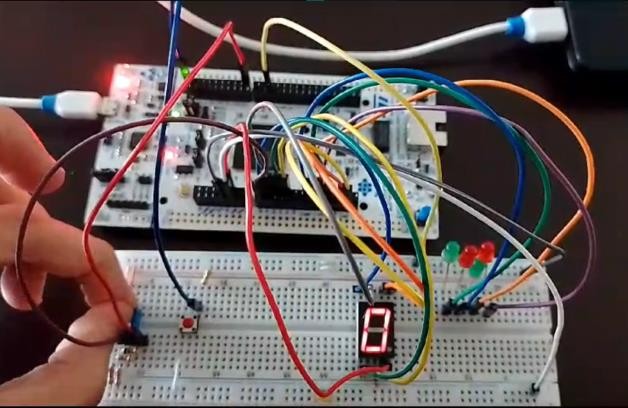
Para aclarar, los comandos de la LCD requieren de un tiempo entre un comando y otro, este tiempo se asegura mediante el uso de SysTick.h, esta librería se creó para agilizar el desarrollo de las prácticas y se aplicara a futuros códigos. Todas las librerías están adjuntas así como el código principal del micro controlador.

1. Análisis

Cabe destacar que para mostrar un numero de más de un digito en la LCD, es necesario separar este número en sus cifras para poder enviarlas individualmente para que sean graficadas en la LCD, además, se requiere convertir este número a su equivalente en ASCII para que pueda ser procesado por la LCD.

También es necesario entender el funcionamiento y asignación de las interrupciones, ya que a partir de la interrupción de los pines 5, se debe tener en cuenta que las interrupciones pueden no funcionar con pines individuales sino con conjuntos de pines, por lo que si se requiere usar más de 1 pin del mismo grupo, se tendría que implementar una lógica que permita diferenciar que interrupción está siendo activada.

1. RESULTADOS.



**Imagen 14.** Montaje motor, nombre y contador.

1. CONCLUCIONES.
2. El control de la LCD mediante STM32 requiere conocer el datasheet del dispositivo LCD, ya que muchos comandos están predefinidos y no se limita solo a enviar información, sino que requiere indicarle a la LCD que debe hacer con la información suministrada.
3. El uso de interrupciones, permite tener un mayor control de las actividades que debe realizar el microcontrolador, así como determinar hasta cuando las debe realizar, teniendo en cuenta señales externas para su detección.