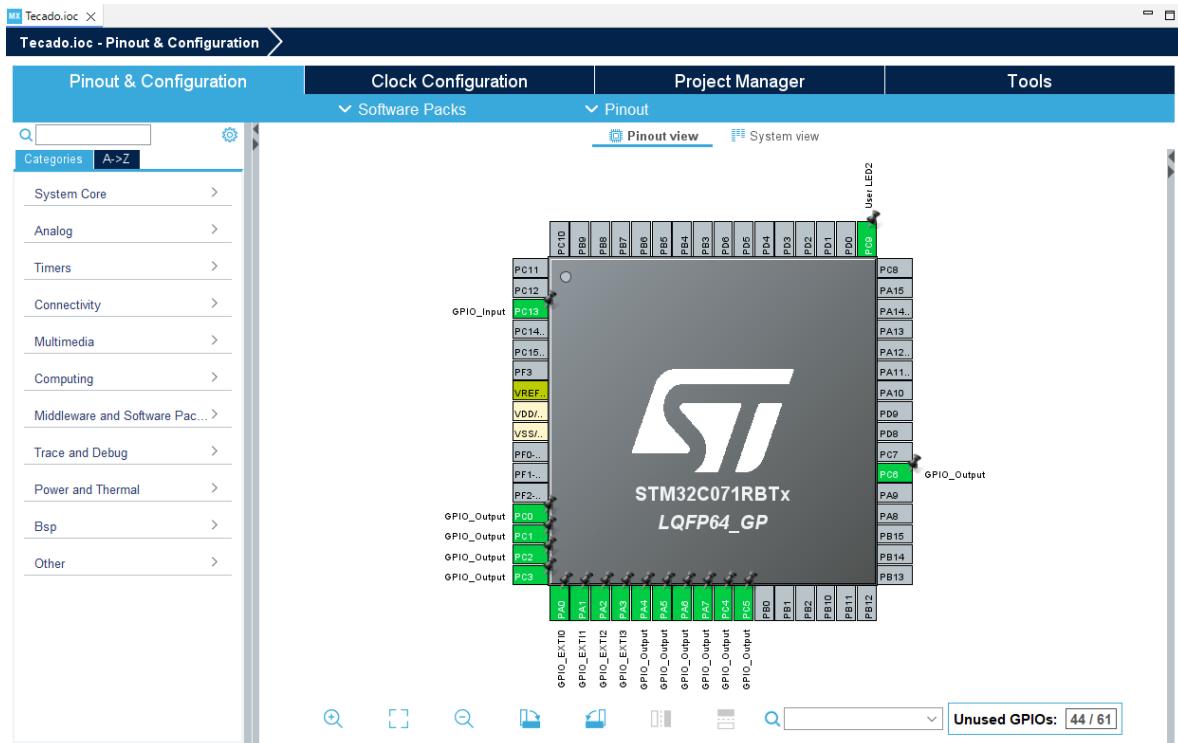


TECLADO

CONFIGURACION INICIAL



MX Tecado.ioc X

Tecado.ioc - Pinout & Configuration >

- Pinout & Configuration**
- Clock Configuration**
- Project Manager**

▼ Software Packs ▼ Pinout

Categories A-Z

System Core

- CORTEX_M0+
- DMA
- FLASH
- GPIO**
- IWDG
- NVIC
- RCC
- SYS
- WWDG

Analog

Timers

Connectivity

Multimedia

Computing

Middleware and Software P...

Trace and Debug

Power and Thermal

Bsp

GPIO Mode and Configuration

Mode

Configuration

Group By Peripherals

GPIO NVIC

Search Signals

Search (Ctrl+F)

Show only Modified Pins

| Pi... | Signal... | GPIO ... | GPIO ... | GPIO ... | Maxi... | Fast ... | User ... | Modified |
|-------|-----------|----------|----------|-----------|---------|----------|----------|-------------------------------------|
| PA0 | n/a | n/a | Exter... | Pull-d... | n/a | n/a | | <input checked="" type="checkbox"/> |
| PA1 | n/a | n/a | Exter... | Pull-d... | n/a | n/a | | <input checked="" type="checkbox"/> |
| PA2 | n/a | n/a | Exter... | Pull-d... | n/a | n/a | | <input checked="" type="checkbox"/> |
| PA3 | n/a | n/a | Exter... | Pull-d... | n/a | n/a | | <input checked="" type="checkbox"/> |
| PA4 | n/a | Low | Outpu... | No pu... | Low | n/a | | <input type="checkbox"/> |
| PA5 | n/a | Low | Outpu... | No pu... | Low | n/a | | <input type="checkbox"/> |
| PA6 | n/a | Low | Outpu... | No pu... | Low | n/a | | <input type="checkbox"/> |

?] Select Pins from table to configure them. Multiple selection is Allowed.

MX Tecado.ioc X

Tecado.ioc - Pinout & Configuration >

Pinout & Configuration Clock Configuration Project Manager

Categories A-Z

System Core

- CORTEX_M0+
- DMA
- FLASH
- GPIO**
- IWDG
- NVIC
- RCC
- SYS
- WWDG

Analog

Timers

Connectivity

Multimedia

Computing

Middleware and Software P...

Trace and Debug

Power and Thermal

Bsp

GPIO Mode and Configuration

Mode

Configuration

Group By Peripherals

GPIO NVIC

| NVIC Interrupt Table | Enabled | Preemption Priority |
|-----------------------------------|-------------------------------------|---------------------|
| EXTI line 0 and line 1 interrupts | <input checked="" type="checkbox"/> | 0 |
| EXTI line 2 and line 3 interrupts | <input checked="" type="checkbox"/> | 0 |

+ 

1.- Descripción del Problema

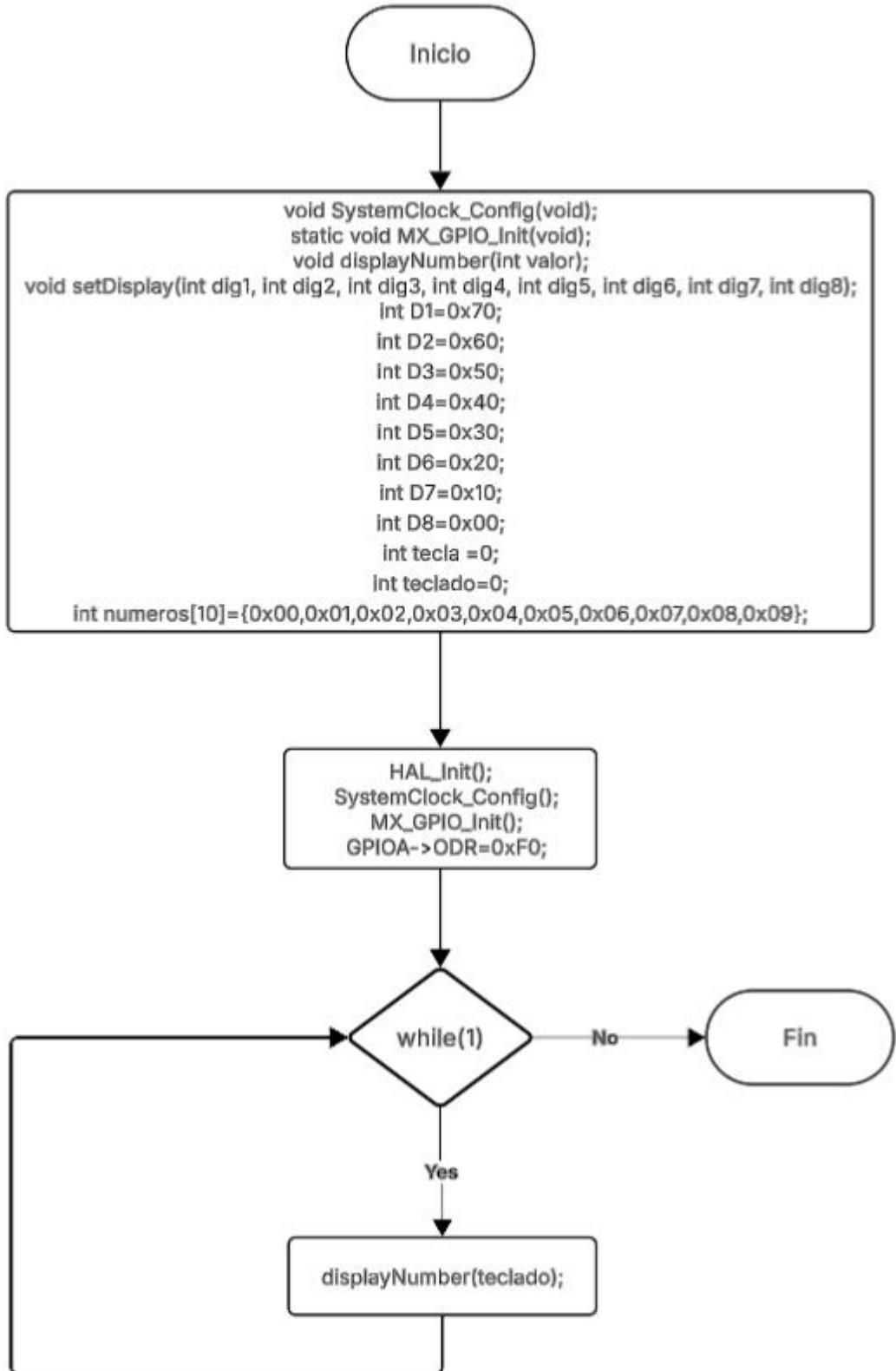
Se requiere un código que decodifique un teclado hexadecimal matricial de membrana y escriba solamente los números en un juego de displays con 8 segmentos.

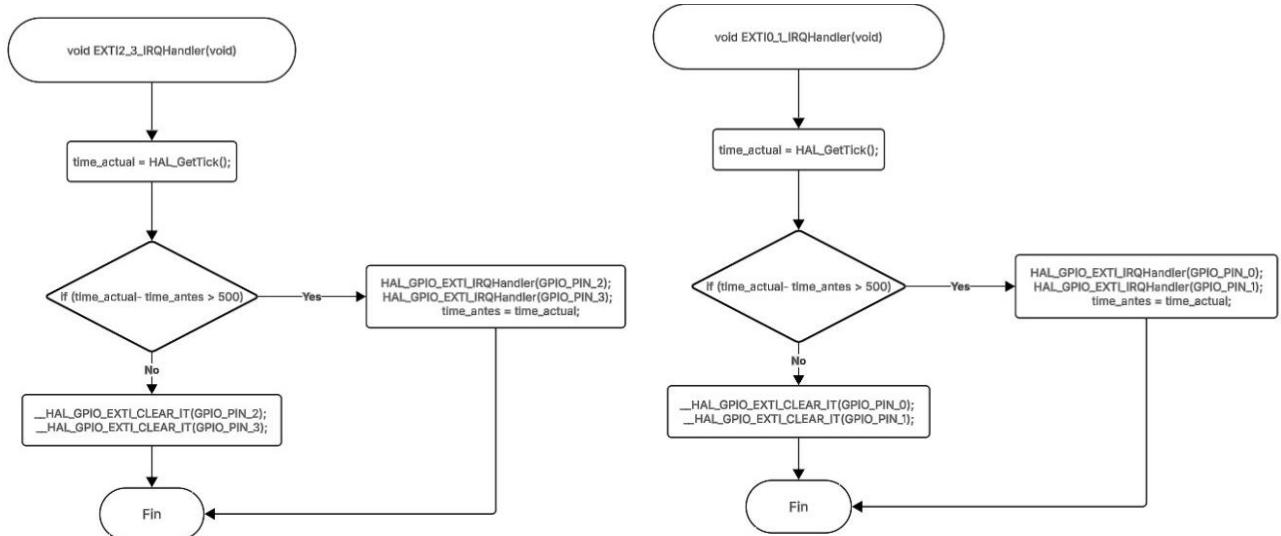
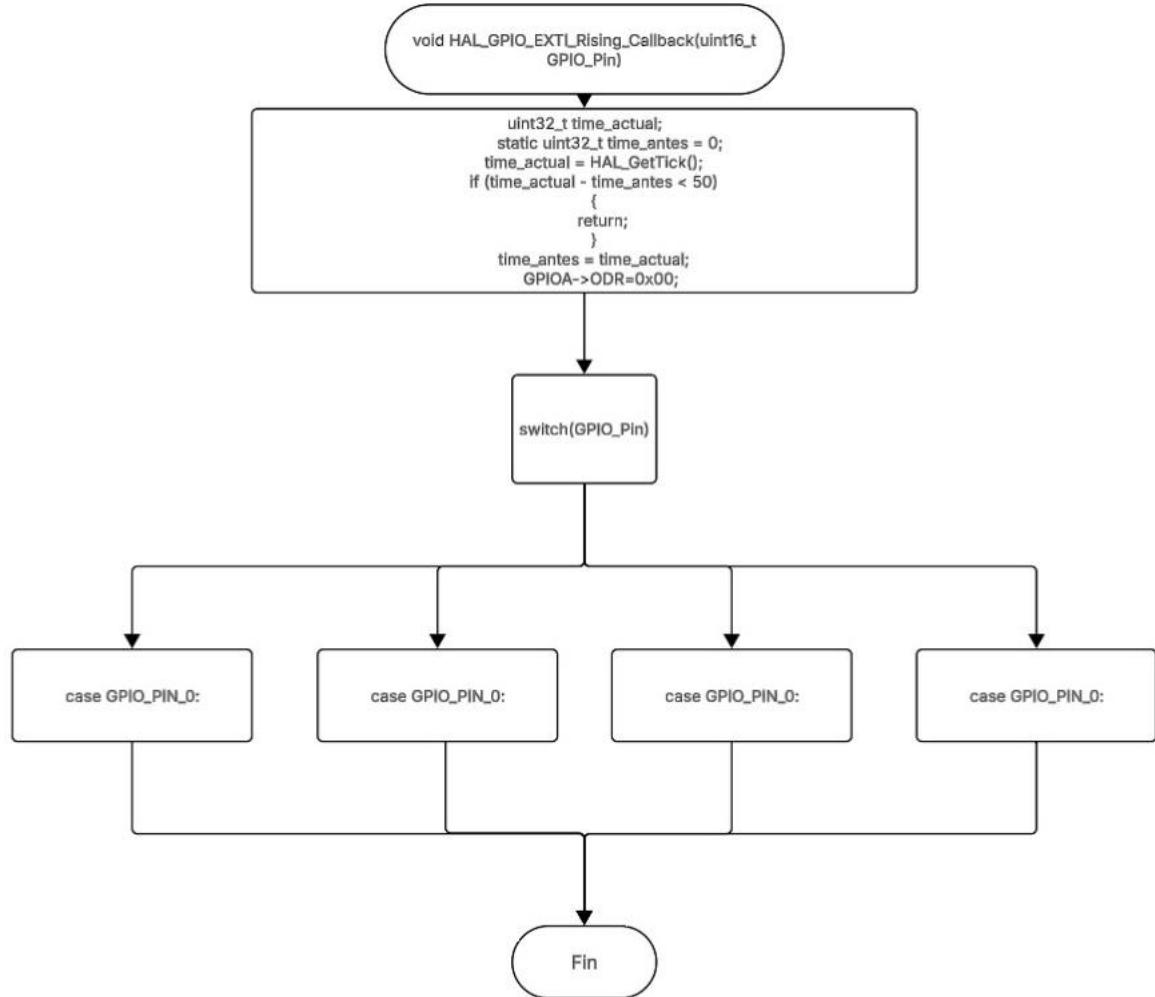
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 |
| 0 | 0 | 0 | 0 | 0 | 1 | 2 | 3 |
| 0 | 0 | 0 | 0 | 1 | 2 | 3 | 4 |
| 0 | 0 | 0 | 1 | 2 | 3 | 4 | 5 |
| 0 | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 |

2.- Objetivo

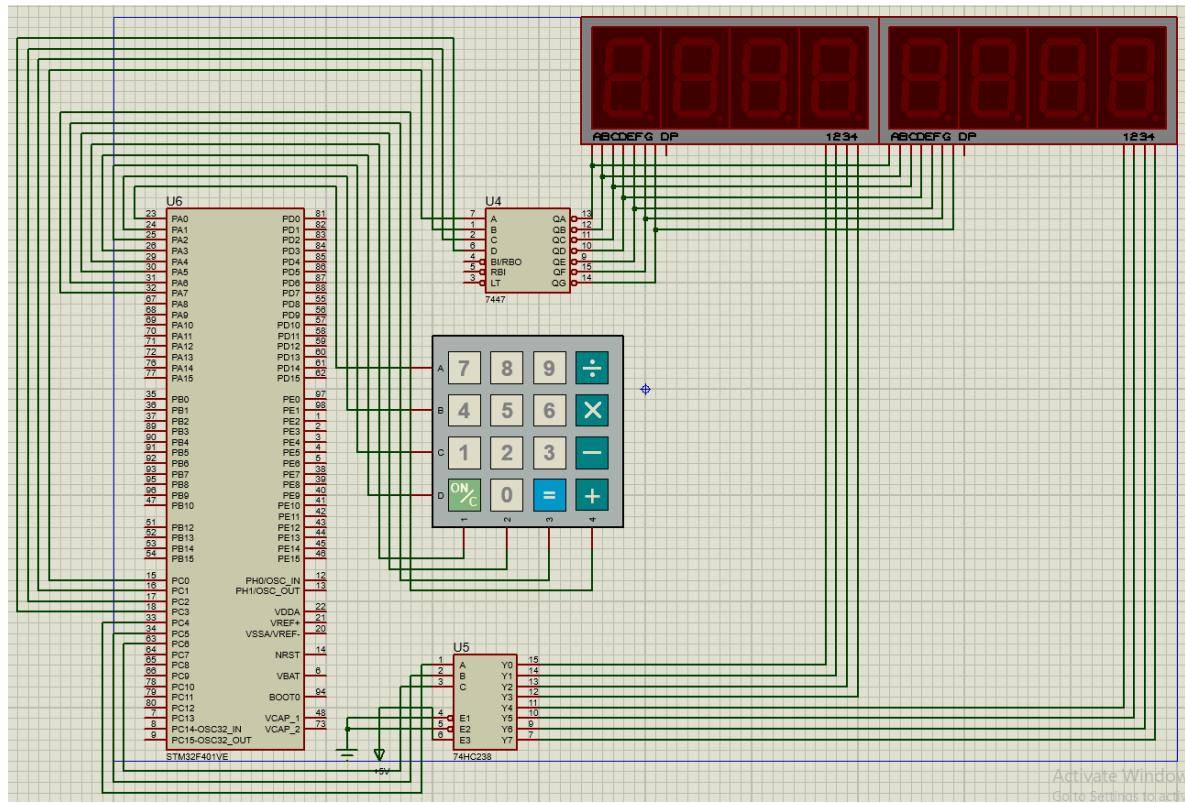
1. Utilizar interrupciones en el teclado para generar mediante funciones los valores a mostrar.
2. Programar el barrido para mostrar los valores deseados.
3. Que el teclado genere y muestre los valores de 0 al 9 en los displays.

3.- Diagrama de Flujo Propuesto



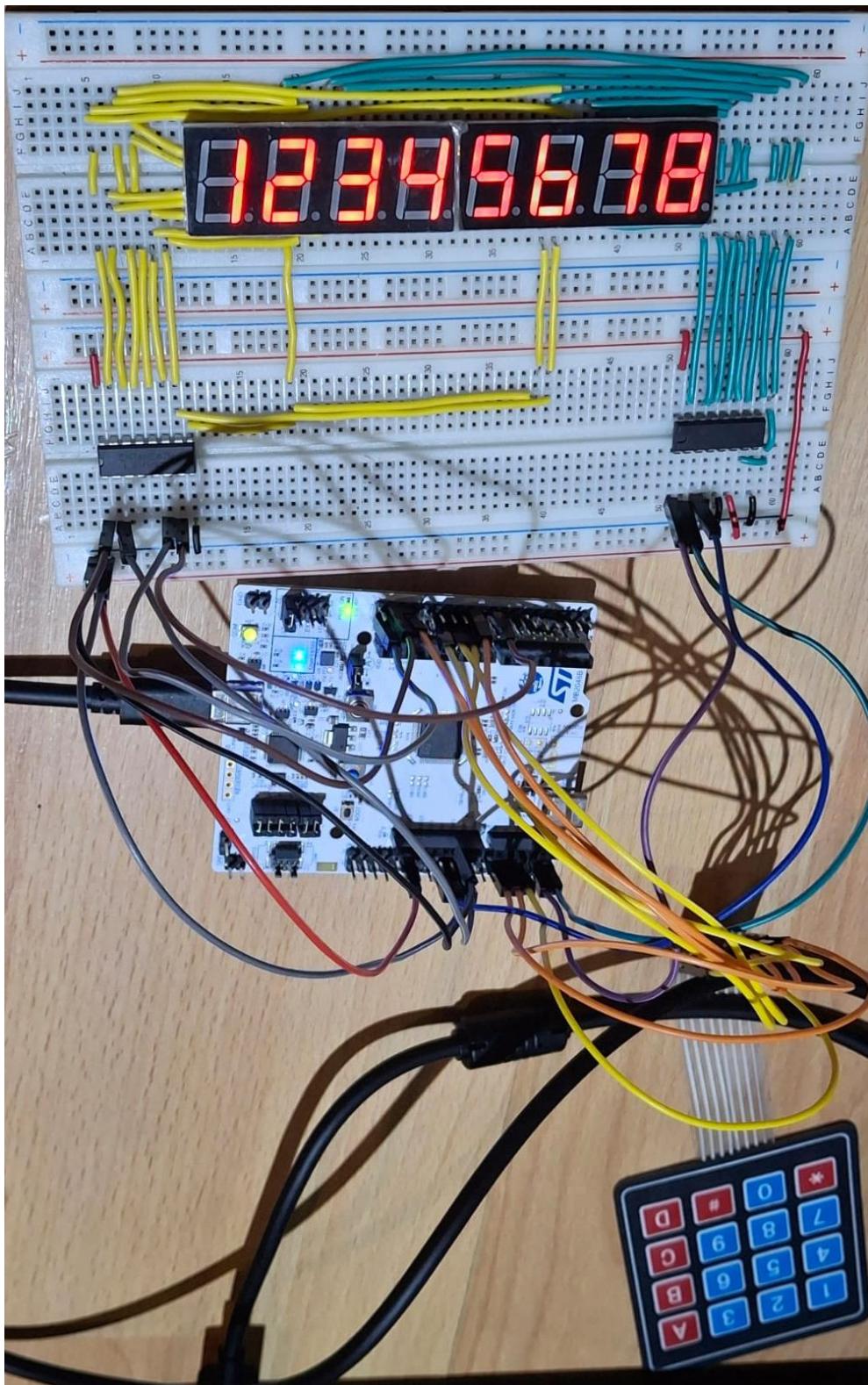


4.- Diagrama de la Implementación del Circuito .



5.- Evidencia Grafica

Agregue algunas imágenes representativas del funcionamiento del circuito.



6.- Prototipos de funciones principales

- Escriba los prototipos de funciones utilizadas para el funcionamiento del display y la subrutinas de interrupción.

Línea 1: void displayNumber(int valor);

Función que convierte un número entero en sus 8 dígitos decimales individuales para mostrarlos en un display multiplexado. Toma el valor numérico completo y realiza divisiones sucesivas para separar cada dígito desde las unidades hasta las decenas de millones.

Línea 2: void setDisplay(int dig1, int dig2, int dig3, int dig4, int dig5, int dig6, int dig7, int dig8);

Función que controla físicamente el display de 8 dígitos multiplexado. Activa secuencialmente cada dígito enviando la combinación adecuada al puerto GPIOC, utilizando delays entre cada activación para crear la persistencia visual.

Línea 3: void HAL_GPIO_EXTI_Rising_Callback(uint16_t GPIO_Pin)

Subrutina de interrupción que se ejecuta automáticamente cuando se detecta un flanco ascendente en los pines GPIO configurados. Implementa un sistema de detección de teclas matricial con debouncing, escaneando filas y columnas para identificar qué tecla fue presionada y actualizando la variable teclado con el nuevo valor ingresado.

| | |
|---|--|
| 1 | void displayNumber(int valor); |
| 2 | void setDisplay(int dig1, int dig2, int dig3, int dig4, int dig5, int dig6, int dig7, int dig8); |
| 3 | void HAL_GPIO_EXTI_Rising_Callback(uint16_t GPIO_Pin) |

- Describa la decodificación del teclado desde la subrutina de interrupción.

Línea 1: void HAL_GPIO_EXTI_Rising_Callback(uint16_t GPIO_Pin)

Subrutina de interrupción que maneja teclas presionadas en una matriz 4x4. Implementa debouncing de 50ms para evitar rebotes, identifica la columna mediante GPIO_Pin, escanea secuencialmente las 4 filas activándolas una por una, detecta la tecla presionada en la intersección fila-columna, y actualiza la variable teclado desplazando los dígitos existentes y agregando la nueva tecla al final.

Línea 5: case GPIO_PIN_0: - Teclas: 1,2,3,0

Escanea la primera columna del teclado matricial. Al activar la fila 1 (PA4) detecta tecla "1", con fila 2 (PA5) detecta tecla "2", con fila 3 (PA6) detecta tecla "3", y con fila 4 (PA7) detecta tecla "0". Cada detección actualiza el display con la fórmula de desplazamiento de dígitos.

Línea 8: case GPIO_PIN_1:- Teclas: 4,5,6,0

Escanea la segunda columna del teclado. Con fila 1 detecta tecla "4", con fila 2 detecta

tecla "5", con fila 3 detecta tecla "6", y con fila 4 detecta tecla "0". Mantiene el mismo patrón de escaneo secuencial por filas.

Línea 11: case GPIO_PIN_2: - Teclas: 7,8,9,0

Escanea la tercera columna del teclado. Con fila 1 detecta tecla "7", con fila 2 detecta tecla "8", con fila 3 detecta tecla "9", y con fila 4 detecta tecla "0". Completa el conjunto numérico del 1 al 9.

Línea 14: case GPIO_PIN_3: - Teclas: 0,0,0,0

Escanea la cuarta columna configurada exclusivamente para tecla "0". Todas las combinaciones de filas en esta columna devuelven el valor cero, proporcionando múltiples puntos de acceso para el dígito cero.

Línea 16: GPIOA->ODR = 0xF0

Reinicializa el puerto GPIOA al estado por defecto después de cada escaneo, estableciendo las filas (PA4-PA7) en alto y las columnas (PA0-PA3) en bajo, preparando el sistema para detectar la siguiente interrupción de teclado.

| | |
|----|--|
| 1 | void HAL_GPIO_EXTI_Rising_Callback(uint16_t GPIO_Pin) |
| 2 | { |
| 3 | switch(GPIO_Pin) |
| 4 | { |
| 5 | case GPIO_PIN_0: |
| 6 | } |
| 7 | { |
| 8 | case GPIO_PIN_1: |
| 9 | } |
| 10 | { |
| 11 | case GPIO_PIN_2: |
| 12 | } |
| 13 | { |
| 14 | case GPIO_PIN_3: |
| 15 | } |
| 16 | GPIOA->ODR=0xF0; |
| 17 | } |

- Describa el funcionamiento del loop de control principal.

Línea 1: int main(void)

Función principal del programa que sirve como punto de entrada de la aplicación. Inicializa todos los periféricos necesarios y entra en el bucle infinito de control.

Línea 3: HAL_Init();

Inicializa la capa de abstracción de hardware (HAL) de STM32, configurando los controladores básicos, el systick para delays e interrupciones temporizadas, y preparando el entorno para el uso de periféricos.

Línea 4: SystemClock_Config();

Configura el reloj del sistema del microcontrolador, estableciendo el HSI (oscilador interno de 16 MHz) como fuente de reloj principal con los divisores apropiados para obtener la frecuencia de operación deseada del CPU y buses periféricos.

Línea 5: MX_GPIO_Init();

Inicializa los pines GPIO, configurando PA0-PA3 como entradas con interrupción por flanco ascendente para el teclado, PA4-PA7 como salidas para las filas del teclado, y GPIOC como salidas para controlar el display de 7 segmentos.

Línea 6: GPIOA->ODR = 0xF0;

Configura el registro de datos de salida del puerto GPIOA con el valor 0xF0, estableciendo los pines PA4-PA7 en estado alto (filas activas) y PA0-PA3 en estado bajo, inicializando el teclado en estado de espera para detección.

Línea 7: while (1)

Bucle infinito principal que se ejecuta continuamente después de la inicialización, manteniendo la aplicación en funcionamiento constante y ejecutando la lógica de refresco del display.

Línea 9: displayNumber(teclado);

Función que se ejecuta continuamente en el bucle principal, tomando el valor actual de la variable teclado y actualizando el display de 8 dígitos con el número ingresado por el usuario mediante el teclado matricial.

| | |
|----|---------------------------|
| 1 | int main(void) |
| 2 | { |
| 3 | HAL_Init(); |
| 4 | SystemClock_Config(); |
| 5 | MX_GPIO_Init(); |
| 6 | GPIOA-> ODR =0xF0; |
| 7 | while (1) |
| 8 | { |
| 9 | displayNumber(teclado); |
| 10 | } |
| 11 | } |