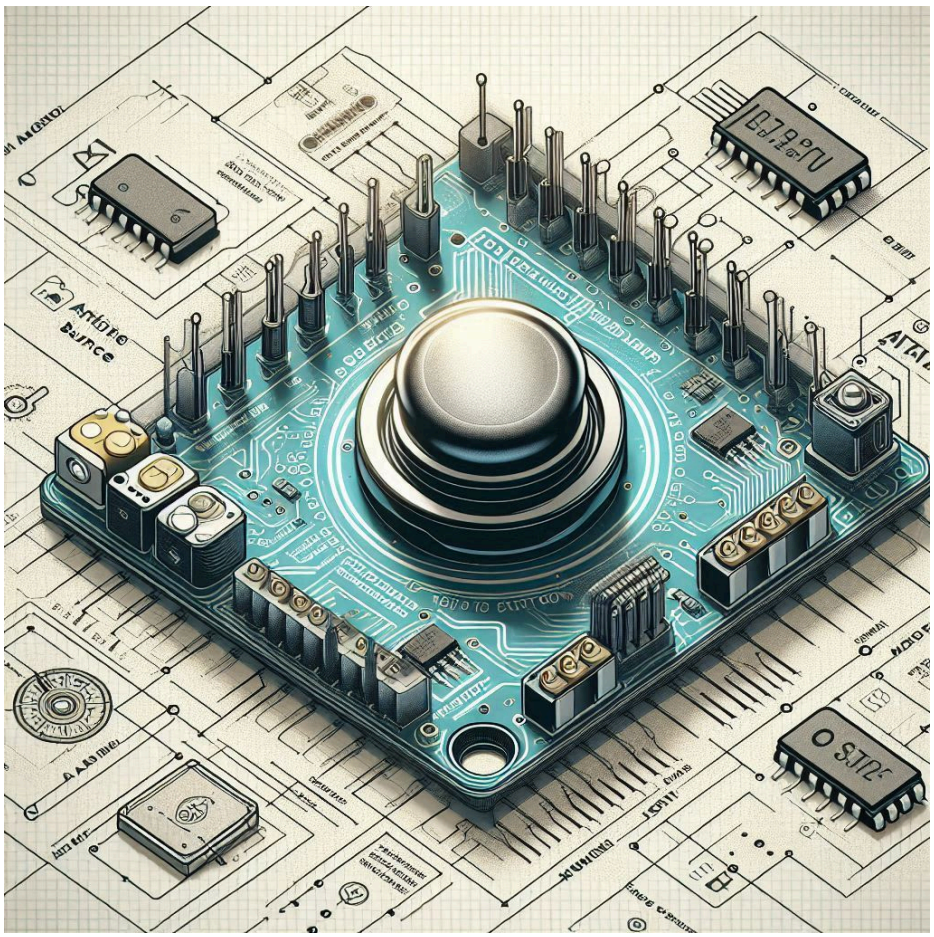
ANTI-REBOTE DESARROLLADO POR MEF PARA DETECCIÓN POSITIVA DE PULSADORES MECÁNICOS



Curso: Técnicas Digitales II

Profesores: Ing. Rubén Darío Mansilla, Ing. Lucas Abdala

Integrantes:

- Barrientos Lucas
- Cuellar Agustin
- Vera Monasterio Candela

Fecha de entrega: Noviembre 2024

11. Informe

11.1 Introducción

Los pulsadores mecánicos presentan un fenómeno conocido como "rebote" cuando son accionados. Este fenómeno produce múltiples transiciones en la señal eléctrica debido a los contactos mecánicos, lo que puede resultar en lecturas erróneas por parte del microcontrolador. La implementación de un sistema anti-rebote mediante una Máquina de Estados Finitos (MEF) representa una solución robusta y eficiente para este problema, permitiendo una detección precisa y confiable del estado real del pulsador.

Objetivo general

Implementar un sistema anti-rebote por software utilizando una Máquina de Estados Finitos, desarrollando un driver modular y reutilizable que garantice una detección confiable de las pulsaciones, integrándose con el sistema de retardos no bloqueantes previamente desarrollado.

Importancia

La implementación de sistemas anti-rebote es crucial en aplicaciones comerciales por múltiples razones:

- Mejora la confiabilidad del sistema al evitar falsas detecciones
- Reduce la probabilidad de errores en la interfaz de usuario
- Optimiza el uso de recursos del microcontrolador al utilizar un enfoque no bloqueante
- Facilita el mantenimiento y la reutilización del código al implementarse como un driver modular
- Permite una integración más robusta con otros sistemas de la aplicación

Descripción del Proceso

El sistema anti-rebote se implementó siguiendo una arquitectura modular basada en una MEF con cuatro estados principales:

1. **BUTTON_UP:** Estado estable cuando el botón no está presionado
2. **BUTTON_FALLING:** Estado transitorio durante la detección inicial de presión
3. **BUTTON_DOWN:** Estado estable cuando el botón está presionado
4. **BUTTON_RISING:** Estado transitorio durante la detección de liberación

Desarrollo del API_Debounce.h:

- Se definieron los tipos de datos necesarios, incluyendo la enumeración **debounceState_t** con los cuatro estados posibles de la MEF (UP, FALLING, DOWN, RISING)
- Se declararon las funciones públicas del driver:
    - **debounceFSM_init:** Para la inicialización.
    - **debounceFSM_update:** Para la actualización periódica.
    - **readKey:** Para la lectura del estado del botón.
    - Funciones de callback para eventos de presión y liberación.

Implementación en API_Debounce.c:

- Se implementó la MEF utilizando una estructura switch-case para manejar las transiciones entre estados.
- Se integró el sistema de retardos no bloqueantes con un tiempo de 40ms para el filtrado de rebotes.
- Se implementó un sistema de banderas (flags) para detectar eventos de presión del botón.
- Se incluyeron las funciones de callback que manipulan los LEDs como respuesta a los eventos.

11.2 Aplicaciones desarrolladas

Modificación de las aplicaciones

Aplicacion 1

La única acción inicial fue copiar los archivos de Drivers/API y pegarlos en la carpeta Drivers del nuevo proyecto (App_1_1_grupo_6_2024). Para la App 1.1 no fue necesaria la implementación del driver al ser una secuencia simple sin interacción.

Aplicacion 2

Las modificaciones principales fueron:

1. **Integración del Driver Anti-rebote**:
    - Se eliminó el sistema anterior de detección del botón (**button_state y last_button_state**)
    - Se agregó la inicialización del driver anti-rebote (**debounceFSM_init()**)
    - Se implementó la actualización periódica de la MEF (**debounceFSM_update()**)
    - Se reemplazó la lectura directa del botón por la función `readKey()`
2. **Mantenimiento de la Funcionalidad**:
    - Se conservó la lógica de cambio de dirección de los LEDs
    - Se mantuvo el sistema de retardos no bloqueantes para la secuencia
    - La funcionalidad de cambio de dirección se implementó usando **readKey()**

Aplicacion 3

Las modificaciones principales fueron:

1. **Integración del Driver Anti-rebote**:
    - Se eliminó el sistema anterior de detección del botón (**button_state y last_button_state**)
    - Se agregó la inicialización del driver anti-rebote (**debounceFSM_init()**)
    - Se implementó la actualización periódica de la MEF (**debounceFSM_update()**)
    - Se reemplazó la lectura directa del botón por la función **readKey()**
2. **Mantenimiento de la Funcionalidad**:
    - Se conservaron intactas las cuatro secuencias de LEDs originales
    - Se mantuvo la lógica de retardos no bloqueantes para las secuencias
    - Se implementaron las funciones de callback **buttonPressed()** y **buttonReleased()**

Aplicacion 4

Las modificaciones principales fueron:

1. **Integración del Driver Anti-rebote**:
    - Se eliminó el sistema anterior de detección del botón (**button_state y last_button_state**)
    - Se implementó una máquina de estados de anti-rebote (**debounceFSM_update()**)
    - Se agregó la función `readKey()` para la detección de pulsaciones
2. **Cambios en la Estructura del Programa**:
    - Se abandonó el bucle único con control directo de LEDs

- Se implementó una estructura con sentencia `switch` para selección de secuencias
- Se añadió inicialización de retardos para cada secuencia
- Se mejoró la modularidad del código mediante funciones específicas para cada secuencia

3. **Gestión de Estados**:
   - Introducción de variables estáticas para mantener el estado entre llamadas
   - Uso de banderas para inicialización de retardos
   - Implementación de lógica de conmutación más compleja

Autor de cada aplicación:

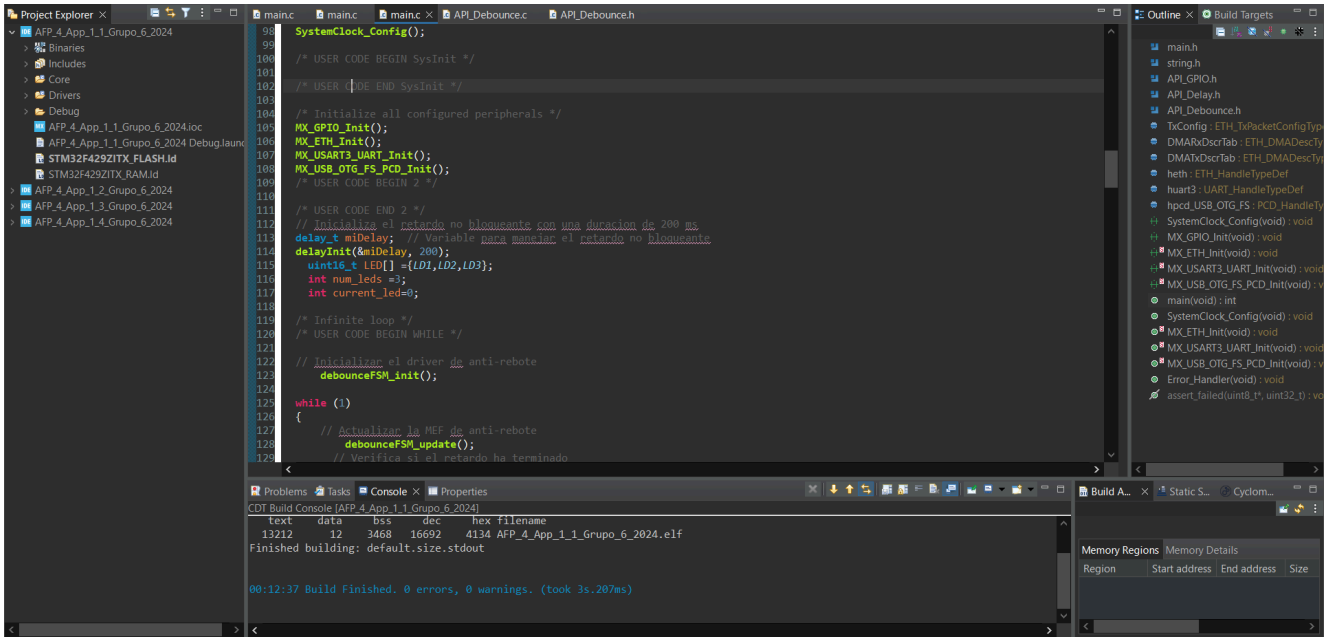
- **Aplicación 1.1:** Vera Monasterio Candela
- **Aplicación 1.2:** Vera Monasterio Candela
- **Aplicación 1.3:** Cuellar Agustin
- **Aplicación 1.4:** Barrientos Lucas

11.3 Link al repositorio grupal

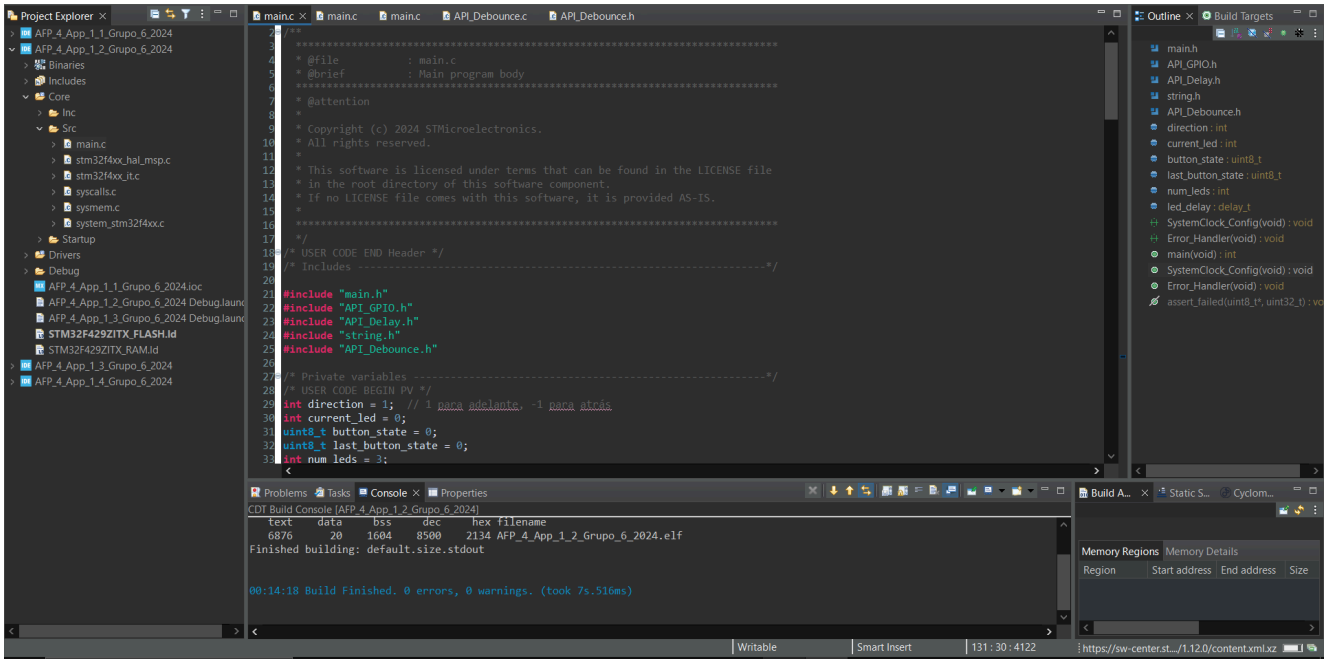
https://github.com/codecuellar/Grupo_6_TDII_2024

En las siguientes capturas de pantalla podemos observar como queda cada aplicación una vez finalizada. Los problemas que se presentaron fueron solucionados después de agregar las direcciones que nos recomendó la IA haciendo que se pueda compilar correctamente, sin ningún error.

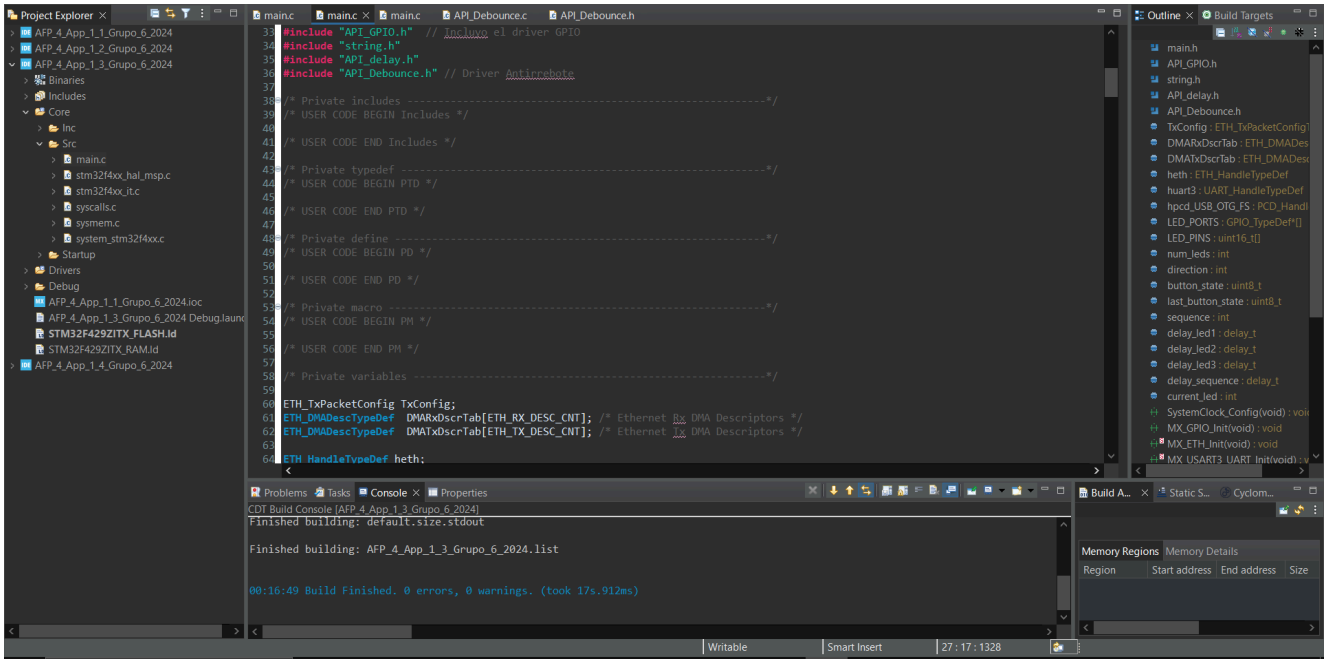
App 1.1


Project Explorer
AFP_4_App_1_1_Grupo_6_2024
Binaries
Includes
Core
Drivers
Debug
AFP_4_App_1_1_Grupo_6_2024.ioc
AFP_4_App_1_1_Grupo_6_2024 Debug.launc
STM32F429ZITX_FLASH.ld
STM32F429ZITX_RAM.ld
AFP_4_App_1_2_Grupo_6_2024
AFP_4_App_1_3_Grupo_6_2024
AFP_4_App_1_4_Grupo_6_2024
main.c
main.c
main.c
API_Debounce.c
API_Debounce.h
SystemClock_Config();
/* USER CODE BEGIN SysInit */
/* USER CODE END SysInit */
/* Initialize all configured peripherals */
MX_GPIO_Init();
MX_ETH_Init();
MX_USART3_UART_Init();
MX_USB_OTG_FS_PCD_Init();
/* USER CODE BEGIN 2 */
/* USER CODE END 2 */
// Inicializa el retardo no bloqueante con una duracion de 200 ms
delay_t miDelay; // Variable para manejar el retardo no bloqueante
delayInit(&miDelay, 200);
uint16_t LED[] ={LD1,LD2,LD3};
int num_leds =3;
int current_led=0;
/* Infinite loop */
/* USER CODE BEGIN WHILE */
// Inicializar el driver de anti-rebote
debounceFSM_init();
while (1)
{
// Actualizar la MEF de anti-rebote
debounceFSM_update();
// Verifica si el retardo ha terminado
Outline
Build Targets
main.h
string.h
API_GPIO.h
API_Delay.h
API_Debounce.h
main(void) : int
SystemClock_Config(void) : void
MX_GPIO_Init(void) : void
MX_ETH_Init(void) : void
Error_Handler(void) : void
Problems
Tasks
Console
Properties
CDT Build Console [AFP_4_App_1_1_Grupo_6_2024]
text data bss dec hex filename
13212 12 3468 16692 4134 AFP_4_App_1_1_Grupo_6_2024.elf
Finished building: default.size.stdout
00:12:37 Build Finished. 0 errors, 0 warnings. (took 3s.207ms)
Build A...
Static S...
Cyclom...
Memory Regions
Memory Details
Region
Start address
End address
Size

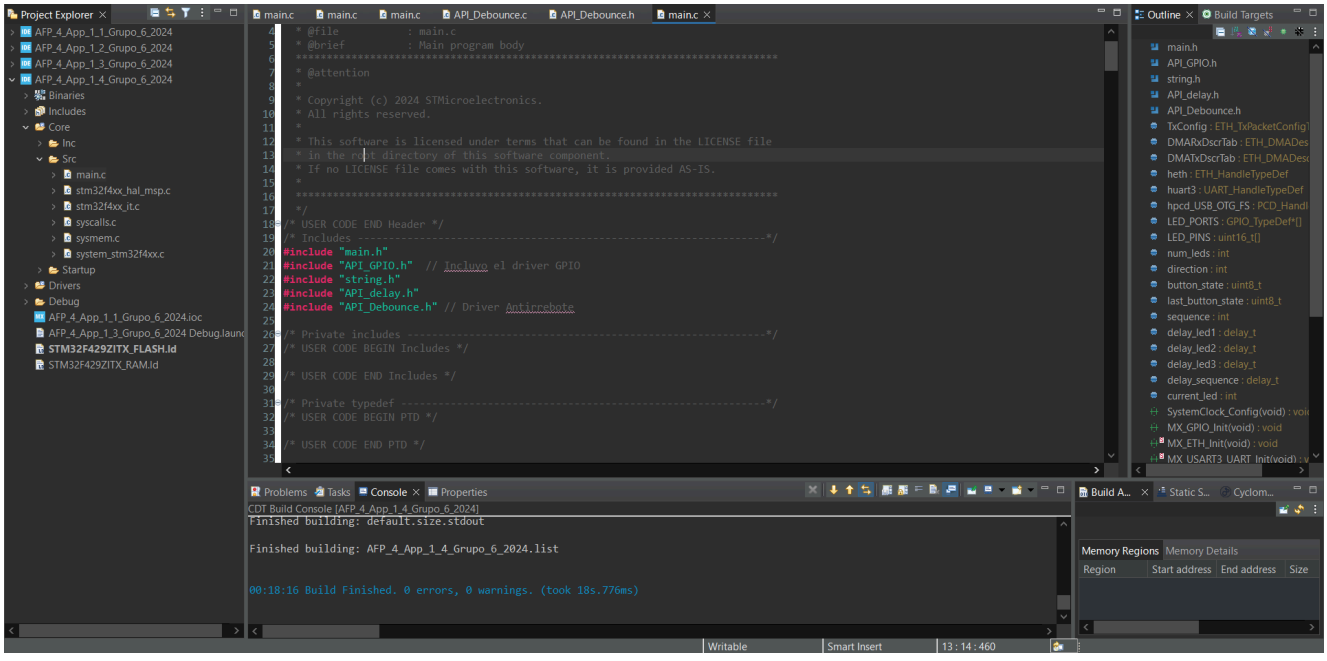
App 1.2


Project Explorer
AFP_4_App_1_1_Grupo_6_2024
AFP_4_App_1_2_Grupo_6_2024
Binaries
Includes
Core
Inc
Src
main.c
stm32f4xx_hal_msp.c
stm32f4xx_it.c
syscalls.c
sysmem.c
system_stm32f4xx.c
Startup
Drivers
Debug
AFP_4_App_1_1_Grupo_6_2024.ioc
AFP_4_App_1_2_Grupo_6_2024 Debug.launc
AFP_4_App_1_3_Grupo_6_2024 Debug.launc
STM32F429ZITX_FLASH.ld
STM32F429ZITX_RAM.ld
AFP_4_App_1_3_Grupo_6_2024
AFP_4_App_1_4_Grupo_6_2024
main.c
main.c
main.c
API_Debounce.c
API_Debounce.h
/**
* @file : main.c
* @brief : Main program body
* @attention
* Copyright (c) 2024 STMicroelectronics.
* All rights reserved.
* This software is licensed under terms that can be found in the LICENSE file
* in the root directory of this software component.
* If no LICENSE file comes with this software, it is provided AS-IS.
*/
/* USER CODE END Header */
/* Includes */
#include "main.h"
#include "API_GPIO.h"
#include "API_Delay.h"
#include "string.h"
#include "API_Debounce.h"
/* Private variables */
/* USER CODE BEGIN PV */
int direction = 1; // 1 para adelante, -1 para atrás
int current_led = 0;
uint8_t button_state = 0;
uint8_t last_button_state = 0;
int num_leds = 3;
Outline
Build Targets
main.h
API_GPIO.h
API_Delay.h
string.h
API_Debounce.h
direction : int
current_led : int
button_state : uint8_t
last_button_state : uint8_t
num_leds : int
led_delay : delay_t
SystemClock_Config(void) : void
Error_Handler(void) : void
main(void) : int
Problems
Tasks
Console
Properties
CDT Build Console [AFP_4_App_1_2_Grupo_6_2024]
text data bss dec hex filename
6876 20 1604 8500 2134 AFP_4_App_1_2_Grupo_6_2024.elf
Finished building: default.size.stdout
00:14:18 Build Finished. 0 errors, 0 warnings. (took 7s.516ms)
Build A...
Static S...
Cyclom...
Memory Regions
Memory Details
Region
Start address
End address
Size
Writable
Smart Insert
131 : 30 : 4122

App 1.3


Project Explorer
AFP_4_App_1_1_Grupo_6_2024
AFP_4_App_1_2_Grupo_6_2024
AFP_4_App_1_3_Grupo_6_2024
Binaries
Includes
Core
Inc
Src
main.c
stm32f4xx_hal_msp.c
stm32f4xx_it.c
syscalls.c
sysmem.c
system_stm32f4xx.c
Startup
Drivers
Debug
AFP_4_App_1_1_Grupo_6_2024.ioc
AFP_4_App_1_3_Grupo_6_2024 Debug.launc
STM32F429ZITX_FLASH.ld
STM32F429ZITX_RAM.ld
AFP_4_App_1_4_Grupo_6_2024
main.c
main.c
main.c
API_Debounce.c
API_Debounce.h
#include "API_GPIO.h" // Incluyo el driver GPIO
#include "string.h"
#include "API_delay.h"
#include "API_Debounce.h" // Driver Antirrebote
/* Private includes */
/* USER CODE BEGIN Includes */
/* USER CODE END Includes */
/* Private typedef */
/* USER CODE BEGIN PTD */
/* USER CODE END PTD */
/* Private define */
/* USER CODE BEGIN PD */
/* USER CODE END PD */
/* Private macro */
/* USER CODE BEGIN PM */
/* USER CODE END PM */
/* Private variables */
ETH_TxPacketConfig TxConfig;
ETH_DMADescTypeDef DMARxDscrTab[ETH_RX_DESC_CNT]; /* Ethernet Rx DMA Descriptors */
ETH_DMADescTypeDef DMATxDscrTab[ETH_TX_DESC_CNT]; /* Ethernet Tx DMA Descriptors */
ETH_HandleTypeDef heth;
Outline
Build Targets
main.h
API_GPIO.h
string.h
API_delay.h
API_Debounce.h
TxConfig : ETH_TxPacketConfig
DMARxDscrTab : ETH_DMADes
DMATxDscrTab : ETH_DMADes
heth : ETH_HandleTypeDef
huart3 : UART_HandleTypeDef
hpcd_USB_OTG_FS : PCD_Handl
LED_PORTS : GPIO_TypeDef*[]
LED_PINS : uint16_t[]
num_leds : int
direction : int
button_state : uint8_t
last_button_state : uint8_t
sequence : int
delay_led1 : delay_t
delay_led2 : delay_t
delay_led3 : delay_t
delay_sequence : delay_t
current_led : int
SystemClock_Config(void) : voi
MX_GPIO_Init(void) : void
MX_ETH_Init(void) : void
MX_USART3_UART_Init(void) : v
Problems
Tasks
Console
Properties
CDT Build Console [AFP_4_App_1_3_Grupo_6_2024]
Finished building: default.size.stdout
Finished building: AFP_4_App_1_3_Grupo_6_2024.list
00:16:49 Build Finished. 0 errors, 0 warnings. (took 17s.912ms)
Build A...
Static S...
Cyclom...
Memory Regions
Memory Details
Region
Start address
End address
Size
Writable
Smart Insert
27 : 17 : 1328

App 1.4


Project Explorer
AFP_4_App_1_1_Grupo_6_2024
AFP_4_App_1_2_Grupo_6_2024
AFP_4_App_1_3_Grupo_6_2024
AFP_4_App_1_4_Grupo_6_2024
Binaries
Includes
Core
Inc
Src
main.c
stm32f4xx_hal_msp.c
stm32f4xx_it.c
syscalls.c
sysmem.c
system_stm32f4xx.c
Startup
Drivers
Debug
AFP_4_App_1_1_Grupo_6_2024.ioc
AFP_4_App_1_3_Grupo_6_2024 Debug.launc
STM32F429ZITX_FLASH.ld
STM32F429ZITX_RAM.ld
main.c
main.c
main.c
API_Debounce.c
API_Debounce.h
main.c
* @file : main.c
* @brief : Main program body
* @attention
* Copyright (c) 2024 STMicroelectronics.
* All rights reserved.
* This software is licensed under terms that can be found in the LICENSE file
* in the root directory of this software component.
* If no LICENSE file comes with this software, it is provided AS-IS.
/* USER CODE END Header */
/* Includes */
#include "main.h"
#include "API_GPIO.h" // Incluyo el driver GPIO
#include "string.h"
#include "API_delay.h"
#include "API_Debounce.h" // Driver Antirrebote
/* Private includes */
/* USER CODE BEGIN Includes */
/* USER CODE END Includes */
/* Private typedef */
/* USER CODE BEGIN PTD */
/* USER CODE END PTD */
Outline
Build Targets
main.h
API_GPIO.h
string.h
API_delay.h
API_Debounce.h
TxConfig : ETH_TxPacketConfig
DMARxDscrTab : ETH_DMADes
DMATxDscrTab : ETH_DMADes
heth : ETH_HandleTypeDef
huart3 : UART_HandleTypeDef
hpcd_USB_OTG_FS : PCD_Handl
LED_PORTS : GPIO_TypeDef*[]
LED_PINS : uint16_t[]
num_leds : int
direction : int
button_state : uint8_t
last_button_state : uint8_t
sequence : int
delay_led1 : delay_t
delay_led2 : delay_t
delay_led3 : delay_t
delay_sequence : delay_t
current_led : int
SystemClock_Config(void) : voi
MX_GPIO_Init(void) : void
MX_ETH_Init(void) : void
MX_USART3_UART_Init(void) : v
Problems
Tasks
Console
Properties
CDT Build Console [AFP_4_App_1_4_Grupo_6_2024]
Finished building: default.size.stdout
Finished building: AFP_4_App_1_4_Grupo_6_2024.list
00:18:16 Build Finished. 0 errors, 0 warnings. (took 18s.776ms)
Build A...
Static S...
Cyclom...
Memory Regions
Memory Details
Region
Start address
End address
Size
Writable
Smart Insert
13 : 14 : 460