```c
/*
 * fram_func.h
 *
 *  Created on: 6 juil. 2023
 *      Author: christop.grobety
 */

#ifndef INC_FRAM_FUNC_H_
#define INC_FRAM_FUNC_H_

#define MEMORY_SIZE 524288
#include "main.h"
#include <stdbool.h>
#include <stdio.h>

/**
 * @brief Write one or more value in a FRAM
 * @param State state_: FRAM where to send the value(s)
 * @param TX_FRAM: Pointer of the table to send
 * @param data_size: Number of values to send on the FRAM
 * @param hspi_1 : SPI of the STM32 to use to send the value
*/
void FRAM_write_one(enum State state_, uint8_t*TX_FRAM, uint8_t data_size, SPI_HandleTypeDef hspi_1);

/**
 * @brief Write one or more value in all the FRAMs
 * @param TX_FRAM: Pointer of the table to send
 * @param data_size: Number of values to send on the FRAMs
 * @param hspi_1 : SPI of the STM32 to use to send the value
*/
void FRAM_write_all(SPI_HandleTypeDef hspi_1, uint8_t*TX_FRAM, uint8_t data_size);

/**
 * @brief Write in all the FRAMs the instruction of receive datas from ADC DEVICE
 * @param hspi_1 : SPI of the STM32 to use to send the value
*/
void FRAM_write_ADC_to_FRAM(SPI_HandleTypeDef hspi_1);

/**
 * @brief Write in all the FRAMs the instruction to set the register for write instruction
 * @param hspi_1 : SPI of the STM32 to use to send the value
*/
void FRAM_write_reg(SPI_HandleTypeDef hspi_1);

/**
 * @brief Write in all the FRAMs the instruction to reset the register
 * @param hspi_1 : SPI of the STM32 to use to send the value
*/
void FRAM_reset_reg(SPI_HandleTypeDef hspi_1);

/**
 * @brief Write in all the FRAMs the instruction to reset the register
 * @param State state_: FRAM where to read the Register's values
 * @param hspi_1 : SPI of the STM32 to use to send the value
*/
void FRAM_read_reg(enum State state_, SPI_HandleTypeDef hspi_1);

/**
 * @brief Read in a FRAM his Device's value
 * @param hspi_1 : SPI of the STM32 to use to send the value
*/
void FRAM_device(SPI_HandleTypeDef hspi_1);

/**
 * @brief Read in a FRAM his Device's value
 * @param State state_: FRAM where to read the Register's values
 * @param add: Location of the first address to start read the value of a FRAM
 * @param hspi_1 : SPI of the STM32 to use to send the value
 * @param data_size : size of the address where to read value
 * @return a pointer of a table with the FRAM's values
*/
```

```c
uint8_t* FRAM_read(enum State state_, uint32_t add,SPI_HandleTypeDef hspi_1, uint8_t data_size);

/**
 * @brief Reset the values of the Selector's pins to send in the FPGA
*/
void PIN_reset();

/**
 * @brief Send the value of pretrigger to the FPGA
 * @param hspi_1 : SPI of the STM32 to use to send the value
 * @param pretrig : pretrig's value to send to the FPGA
*/
void setPreTrigg(SPI_HandleTypeDef hspi_1, uint8_t pretrig);

/**
 * @brief Selection of the channel to trigg
 * @param chan: value of the channel to use
*/
void setTriggChannel(enum trig_channel chan);

/**
 * @brief To turn on a Led
 * @param color: color of the Led to turn on
*/
void LED_on(enum color_Led color);

/**
 * @brief To turn off a Led
*/
void LED_off();


#endif /* INC_FRAM_FUNC_H_ */

```