

```

1  /*
2  * fram_func.h
3  *
4  * Created on: 6 juil. 2023
5  * Author: christop.grobety
6  */
7
8  #ifndef INC_FRAM_FUNC_H_
9  #define INC_FRAM_FUNC_H_
10
11  #define MEMORY_SIZE 524288
12  #include "main.h"
13  #include <stdbool.h>
14  #include <stdio.h>
15
16  /**
17   * @brief Write one or more value in a FRAM
18   * @param State state_: FRAM where to send the value(s)
19   * @param TX_FRAM: Pointer of the table to send
20   * @param data_size: Number of values to send on the FRAM
21   * @param hspi_1 : SPI of the STM32 to use to send the value
22   */
23  void FRAM_write_one(enum State state_, uint8_t*TX_FRAM, uint8_t data_size,
24  SPI_HandleTypeDef hspi_1);
25
26  /**
27   * @brief Write one or more value in all the FRAMs
28   * @param TX_FRAM: Pointer of the table to send
29   * @param data_size: Number of values to send on the FRAMs
30   * @param hspi_1 : SPI of the STM32 to use to send the value
31   */
32  void FRAM_write_all(SPI_HandleTypeDef hspi_1, uint8_t*TX_FRAM, uint8_t data_size);
33
34  /**
35   * @brief Write in all the FRAMs the instruction of receive datas from ADC DEVICE
36   * @param hspi_1 : SPI of the STM32 to use to send the value
37   */
38  void FRAM_write_ADC_to_FRAM(SPI_HandleTypeDef hspi_1);
39
40  /**
41   * @brief Write in all the FRAMs the instruction to set the register for write
42   * instruction
43   * @param hspi_1 : SPI of the STM32 to use to send the value
44   */
45  void FRAM_write_reg(SPI_HandleTypeDef hspi_1);
46
47  /**
48   * @brief Write in all the FRAMs the instruction to reset the register
49   * @param hspi_1 : SPI of the STM32 to use to send the value
50   */
51  void FRAM_reset_reg(SPI_HandleTypeDef hspi_1);
52
53  /**
54   * @brief Write in all the FRAMs the instruction to reset the register
55   * @param State state_: FRAM where to read the Register's values
56   * @param hspi_1 : SPI of the STM32 to use to send the value
57   */
58  void FRAM_read_reg(enum State state_, SPI_HandleTypeDef hspi_1);
59
60  /**
61   * @brief Read in a FRAM his Device's value
62   * @param hspi_1 : SPI of the STM32 to use to send the value
63   */
64  void FRAM_device(SPI_HandleTypeDef hspi_1);
65
66  /**
67   * @brief Read in a FRAM his Device's value
68   * @param State state_: FRAM where to read the Register's values
69   * @param add: Location of the first address to start read the value of a FRAM
70   * @param hspi_1 : SPI of the STM32 to use to send the value
71   * @param data_size : size of the address where to read value
72   * @return a pointer of a table with the FRAM's values
73   */

```

```

72  uint8_t* FRAM_read(enum State state_, uint32_t add, SPI_HandleTypeDef hspi_1, uint8_t
data_size);
73
74  /**
75   * @brief Reset the values of the Selector's pins to send in the FPGA
76   */
77  void PIN_reset();
78
79  /**
80   * @brief Send the value of pretrigger to the FPGA
81   * @param hspi_1 : SPI of the STM32 to use to send the value
82   * @param pretrig : pretrig's value to send to the FPGA
83   */
84  void setPreTrigg(SPI_HandleTypeDef hspi_1, uint8_t pretrig);
85
86  /**
87   * @brief Selection of the channel to trigg
88   * @param chan: value of the channel to use
89   */
90  void setTriggChannel(enum trig_channel chan);
91
92  /**
93   * @brief To turn on a Led
94   * @param color: color of the Led to turn on
95   */
96  void LED_on(enum color_Led color);
97
98  /**
99   * @brief To turn off a Led
100  */
101  void LED_off();
102
103
104  #endif /* INC_FRAM_FUNC_H_ */
105

```