**FSM Assignment**

* Code:

/\* USER CODE BEGIN Header \*/

/\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* @file : main.c

\* @brief : Main program body

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* @attention

\*

\* <h2><center>&copy; Copyright (c) 2022 STMicroelectronics.

\* All rights reserved.</center></h2>

\*

\* This software component is licensed by ST under BSD 3-Clause license,

\* the "License"; You may not use this file except in compliance with the

\* License. You may obtain a copy of the License at:

\* opensource.org/licenses/BSD-3-Clause

\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

/\* USER CODE END Header \*/

/\* Includes ------------------------------------------------------------------\*/

**#include** "main.h"

/\* 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 ---------------------------------------------------------\*/

/\* USER CODE BEGIN PV \*/

/\* USER CODE END PV \*/

/\* Private function prototypes -----------------------------------------------\*/

**void** **SystemClock\_Config**(**void**);

**static** **void** **MX\_GPIO\_Init**(**void**);

/\* USER CODE BEGIN PFP \*/

/\* USER CODE END PFP \*/

/\* Private user code ---------------------------------------------------------\*/

/\* USER CODE BEGIN 0 \*/

/\* USER CODE END 0 \*/

/\*\*

\* @brief The application entry point.

\* @retval int

\*/

**int** **main**(**void**)

{

/\* USER CODE BEGIN 1 \*/

/\* USER CODE END 1 \*/

/\* MCU Configuration--------------------------------------------------------\*/

/\* Reset of all peripherals, Initializes the Flash interface and the Systick. \*/

HAL\_Init();

/\* USER CODE BEGIN Init \*/

/\* USER CODE END Init \*/

/\* Configure the system clock \*/

SystemClock\_Config();

/\* USER CODE BEGIN SysInit \*/

/\* USER CODE END SysInit \*/

/\* Initialize all configured peripherals \*/

MX\_GPIO\_Init();

/\* USER CODE BEGIN 2 \*/

/\* USER CODE END 2 \*/

/\* Infinite loop \*/

/\* USER CODE BEGIN WHILE \*/

**int** ctr=0;

**int** pos;

**int** loop=1;

**int** pre;

**while** (1)

{

/\* USER CODE END WHILE \*/

/\* USER CODE BEGIN 3 \*/

pre=pos;

pos=HAL\_GPIO\_ReadPin(button\_GPIO\_Port, button\_Pin);

**if**(pos==0 && pre==1){

ctr=ctr+1;

**if**(ctr==1){

**for**(**int** i=0;i<5;i++){

HAL\_GPIO\_WritePin(led1\_GPIO\_Port, led1\_Pin, *GPIO\_PIN\_SET*);

HAL\_Delay(1000);

HAL\_GPIO\_WritePin(led1\_GPIO\_Port, led1\_Pin, *GPIO\_PIN\_RESET*);

HAL\_Delay(1000);

}

}

**if**(ctr==2){

**for**(**int** i=0;i<5;i++){

HAL\_GPIO\_WritePin(led2\_GPIO\_Port, led2\_Pin, *GPIO\_PIN\_SET*);

HAL\_Delay(1000);

HAL\_GPIO\_WritePin(led2\_GPIO\_Port, led2\_Pin, *GPIO\_PIN\_RESET*);

HAL\_Delay(1000);

}

}

**if**(ctr==3){

HAL\_GPIO\_WritePin(led1\_GPIO\_Port, led1\_Pin, *GPIO\_PIN\_SET*);

HAL\_GPIO\_WritePin(led2\_GPIO\_Port, led2\_Pin, *GPIO\_PIN\_SET*);

}

**if**(ctr==4){

HAL\_GPIO\_WritePin(led1\_GPIO\_Port, led1\_Pin, *GPIO\_PIN\_RESET*);

HAL\_GPIO\_WritePin(led2\_GPIO\_Port, led2\_Pin, *GPIO\_PIN\_RESET*);

}

**if**(ctr==5){

ctr=0;

}

}

/\* USER CODE BEGIN 3 \*/

}

/\* USER CODE END 3 \*/

/\*\*

\* @brief System Clock Configuration

\* @retval None

\*/

**void** **SystemClock\_Config**(**void**)

{

RCC\_OscInitTypeDef RCC\_OscInitStruct = {0};

RCC\_ClkInitTypeDef RCC\_ClkInitStruct = {0};

/\*\* Configure the main internal regulator output voltage

\*/

**if** (HAL\_PWREx\_ControlVoltageScaling(PWR\_REGULATOR\_VOLTAGE\_SCALE1) != *HAL\_OK*)

{

Error\_Handler();

}

/\*\* Initializes the RCC Oscillators according to the specified parameters

\* in the RCC\_OscInitTypeDef structure.

\*/

RCC\_OscInitStruct.OscillatorType = RCC\_OSCILLATORTYPE\_MSI;

RCC\_OscInitStruct.MSIState = RCC\_MSI\_ON;

RCC\_OscInitStruct.MSICalibrationValue = 0;

RCC\_OscInitStruct.MSIClockRange = RCC\_MSIRANGE\_6;

RCC\_OscInitStruct.PLL.PLLState = RCC\_PLL\_ON;

RCC\_OscInitStruct.PLL.PLLSource = RCC\_PLLSOURCE\_MSI;

RCC\_OscInitStruct.PLL.PLLM = 1;

RCC\_OscInitStruct.PLL.PLLN = 40;

RCC\_OscInitStruct.PLL.PLLP = RCC\_PLLP\_DIV7;

RCC\_OscInitStruct.PLL.PLLQ = RCC\_PLLQ\_DIV2;

RCC\_OscInitStruct.PLL.PLLR = RCC\_PLLR\_DIV2;

**if** (HAL\_RCC\_OscConfig(&RCC\_OscInitStruct) != *HAL\_OK*)

{

Error\_Handler();

}

/\*\* Initializes the CPU, AHB and APB buses clocks

\*/

RCC\_ClkInitStruct.ClockType = RCC\_CLOCKTYPE\_HCLK|RCC\_CLOCKTYPE\_SYSCLK

|RCC\_CLOCKTYPE\_PCLK1|RCC\_CLOCKTYPE\_PCLK2;

RCC\_ClkInitStruct.SYSCLKSource = RCC\_SYSCLKSOURCE\_PLLCLK;

RCC\_ClkInitStruct.AHBCLKDivider = RCC\_SYSCLK\_DIV1;

RCC\_ClkInitStruct.APB1CLKDivider = RCC\_HCLK\_DIV1;

RCC\_ClkInitStruct.APB2CLKDivider = RCC\_HCLK\_DIV1;

**if** (HAL\_RCC\_ClockConfig(&RCC\_ClkInitStruct, FLASH\_LATENCY\_4) != *HAL\_OK*)

{

Error\_Handler();

}

}

/\*\*

\* @brief GPIO Initialization Function

\* @param None

\* @retval None

\*/

**static** **void** **MX\_GPIO\_Init**(**void**)

{

GPIO\_InitTypeDef GPIO\_InitStruct = {0};

/\* GPIO Ports Clock Enable \*/

\_\_HAL\_RCC\_GPIOE\_CLK\_ENABLE();

\_\_HAL\_RCC\_GPIOC\_CLK\_ENABLE();

\_\_HAL\_RCC\_GPIOA\_CLK\_ENABLE();

\_\_HAL\_RCC\_GPIOB\_CLK\_ENABLE();

\_\_HAL\_RCC\_GPIOD\_CLK\_ENABLE();

/\*Configure GPIO pin Output Level \*/

HAL\_GPIO\_WritePin(GPIOE, M24SR64\_Y\_RF\_DISABLE\_Pin|M24SR64\_Y\_GPO\_Pin|ISM43362\_RST\_Pin, *GPIO\_PIN\_RESET*);

/\*Configure GPIO pin Output Level \*/

HAL\_GPIO\_WritePin(GPIOA, ARD\_D10\_Pin|led1\_Pin|SPBTLE\_RF\_RST\_Pin|ARD\_D9\_Pin, *GPIO\_PIN\_RESET*);

/\*Configure GPIO pin Output Level \*/

HAL\_GPIO\_WritePin(GPIOB, ARD\_D8\_Pin|ISM43362\_BOOT0\_Pin|ISM43362\_WAKEUP\_Pin|led2\_Pin

|SPSGRF\_915\_SDN\_Pin|ARD\_D5\_Pin, *GPIO\_PIN\_RESET*);

/\*Configure GPIO pin Output Level \*/

HAL\_GPIO\_WritePin(GPIOD, USB\_OTG\_FS\_PWR\_EN\_Pin|PMOD\_RESET\_Pin|STSAFE\_A100\_RESET\_Pin, *GPIO\_PIN\_RESET*);

/\*Configure GPIO pin Output Level \*/

HAL\_GPIO\_WritePin(SPBTLE\_RF\_SPI3\_CSN\_GPIO\_Port, SPBTLE\_RF\_SPI3\_CSN\_Pin, *GPIO\_PIN\_SET*);

/\*Configure GPIO pin Output Level \*/

HAL\_GPIO\_WritePin(GPIOC, VL53L0X\_XSHUT\_Pin|LED3\_WIFI\_\_LED4\_BLE\_Pin, *GPIO\_PIN\_RESET*);

/\*Configure GPIO pin Output Level \*/

HAL\_GPIO\_WritePin(SPSGRF\_915\_SPI3\_CSN\_GPIO\_Port, SPSGRF\_915\_SPI3\_CSN\_Pin, *GPIO\_PIN\_SET*);

/\*Configure GPIO pin Output Level \*/

HAL\_GPIO\_WritePin(ISM43362\_SPI3\_CSN\_GPIO\_Port, ISM43362\_SPI3\_CSN\_Pin, *GPIO\_PIN\_SET*);

/\*Configure GPIO pins : M24SR64\_Y\_RF\_DISABLE\_Pin M24SR64\_Y\_GPO\_Pin ISM43362\_RST\_Pin ISM43362\_SPI3\_CSN\_Pin \*/

GPIO\_InitStruct.Pin = M24SR64\_Y\_RF\_DISABLE\_Pin|M24SR64\_Y\_GPO\_Pin|ISM43362\_RST\_Pin|ISM43362\_SPI3\_CSN\_Pin;

GPIO\_InitStruct.Mode = GPIO\_MODE\_OUTPUT\_PP;

GPIO\_InitStruct.Pull = GPIO\_NOPULL;

GPIO\_InitStruct.Speed = GPIO\_SPEED\_FREQ\_LOW;

HAL\_GPIO\_Init(GPIOE, &GPIO\_InitStruct);

/\*Configure GPIO pins : USB\_OTG\_FS\_OVRCR\_EXTI3\_Pin SPSGRF\_915\_GPIO3\_EXTI5\_Pin SPBTLE\_RF\_IRQ\_EXTI6\_Pin ISM43362\_DRDY\_EXTI1\_Pin \*/

GPIO\_InitStruct.Pin = USB\_OTG\_FS\_OVRCR\_EXTI3\_Pin|SPSGRF\_915\_GPIO3\_EXTI5\_Pin|SPBTLE\_RF\_IRQ\_EXTI6\_Pin|ISM43362\_DRDY\_EXTI1\_Pin;

GPIO\_InitStruct.Mode = GPIO\_MODE\_IT\_RISING;

GPIO\_InitStruct.Pull = GPIO\_NOPULL;

HAL\_GPIO\_Init(GPIOE, &GPIO\_InitStruct);

/\*Configure GPIO pin : button\_Pin \*/

GPIO\_InitStruct.Pin = button\_Pin;

GPIO\_InitStruct.Mode = GPIO\_MODE\_INPUT;

GPIO\_InitStruct.Pull = GPIO\_NOPULL;

HAL\_GPIO\_Init(button\_GPIO\_Port, &GPIO\_InitStruct);

/\*Configure GPIO pins : ARD\_A5\_Pin ARD\_A4\_Pin ARD\_A3\_Pin ARD\_A2\_Pin

ARD\_A1\_Pin ARD\_A0\_Pin \*/

GPIO\_InitStruct.Pin = ARD\_A5\_Pin|ARD\_A4\_Pin|ARD\_A3\_Pin|ARD\_A2\_Pin

|ARD\_A1\_Pin|ARD\_A0\_Pin;

GPIO\_InitStruct.Mode = GPIO\_MODE\_ANALOG\_ADC\_CONTROL;

GPIO\_InitStruct.Pull = GPIO\_NOPULL;

HAL\_GPIO\_Init(GPIOC, &GPIO\_InitStruct);

/\*Configure GPIO pins : ARD\_D1\_Pin ARD\_D0\_Pin \*/

GPIO\_InitStruct.Pin = ARD\_D1\_Pin|ARD\_D0\_Pin;

GPIO\_InitStruct.Mode = GPIO\_MODE\_AF\_PP;

GPIO\_InitStruct.Pull = GPIO\_NOPULL;

GPIO\_InitStruct.Speed = GPIO\_SPEED\_FREQ\_VERY\_HIGH;

GPIO\_InitStruct.Alternate = GPIO\_AF8\_UART4;

HAL\_GPIO\_Init(GPIOA, &GPIO\_InitStruct);

/\*Configure GPIO pins : ARD\_D10\_Pin led1\_Pin SPBTLE\_RF\_RST\_Pin ARD\_D9\_Pin \*/

GPIO\_InitStruct.Pin = ARD\_D10\_Pin|led1\_Pin|SPBTLE\_RF\_RST\_Pin|ARD\_D9\_Pin;

GPIO\_InitStruct.Mode = GPIO\_MODE\_OUTPUT\_PP;

GPIO\_InitStruct.Pull = GPIO\_NOPULL;

GPIO\_InitStruct.Speed = GPIO\_SPEED\_FREQ\_LOW;

HAL\_GPIO\_Init(GPIOA, &GPIO\_InitStruct);

/\*Configure GPIO pin : ARD\_D4\_Pin \*/

GPIO\_InitStruct.Pin = ARD\_D4\_Pin;

GPIO\_InitStruct.Mode = GPIO\_MODE\_AF\_PP;

GPIO\_InitStruct.Pull = GPIO\_NOPULL;

GPIO\_InitStruct.Speed = GPIO\_SPEED\_FREQ\_LOW;

GPIO\_InitStruct.Alternate = GPIO\_AF1\_TIM2;

HAL\_GPIO\_Init(ARD\_D4\_GPIO\_Port, &GPIO\_InitStruct);

/\*Configure GPIO pin : ARD\_D7\_Pin \*/

GPIO\_InitStruct.Pin = ARD\_D7\_Pin;

GPIO\_InitStruct.Mode = GPIO\_MODE\_ANALOG\_ADC\_CONTROL;

GPIO\_InitStruct.Pull = GPIO\_NOPULL;

HAL\_GPIO\_Init(ARD\_D7\_GPIO\_Port, &GPIO\_InitStruct);

/\*Configure GPIO pins : ARD\_D12\_Pin ARD\_D11\_Pin \*/

GPIO\_InitStruct.Pin = ARD\_D12\_Pin|ARD\_D11\_Pin;

GPIO\_InitStruct.Mode = GPIO\_MODE\_AF\_PP;

GPIO\_InitStruct.Pull = GPIO\_NOPULL;

GPIO\_InitStruct.Speed = GPIO\_SPEED\_FREQ\_VERY\_HIGH;

GPIO\_InitStruct.Alternate = GPIO\_AF5\_SPI1;

HAL\_GPIO\_Init(GPIOA, &GPIO\_InitStruct);

/\*Configure GPIO pin : ARD\_D3\_Pin \*/

GPIO\_InitStruct.Pin = ARD\_D3\_Pin;

GPIO\_InitStruct.Mode = GPIO\_MODE\_IT\_RISING;

GPIO\_InitStruct.Pull = GPIO\_NOPULL;

HAL\_GPIO\_Init(ARD\_D3\_GPIO\_Port, &GPIO\_InitStruct);

/\*Configure GPIO pin : ARD\_D6\_Pin \*/

GPIO\_InitStruct.Pin = ARD\_D6\_Pin;

GPIO\_InitStruct.Mode = GPIO\_MODE\_ANALOG\_ADC\_CONTROL;

GPIO\_InitStruct.Pull = GPIO\_NOPULL;

HAL\_GPIO\_Init(ARD\_D6\_GPIO\_Port, &GPIO\_InitStruct);

/\*Configure GPIO pins : ARD\_D8\_Pin ISM43362\_BOOT0\_Pin ISM43362\_WAKEUP\_Pin led2\_Pin

SPSGRF\_915\_SDN\_Pin ARD\_D5\_Pin SPSGRF\_915\_SPI3\_CSN\_Pin \*/

GPIO\_InitStruct.Pin = ARD\_D8\_Pin|ISM43362\_BOOT0\_Pin|ISM43362\_WAKEUP\_Pin|led2\_Pin

|SPSGRF\_915\_SDN\_Pin|ARD\_D5\_Pin|SPSGRF\_915\_SPI3\_CSN\_Pin;

GPIO\_InitStruct.Mode = GPIO\_MODE\_OUTPUT\_PP;

GPIO\_InitStruct.Pull = GPIO\_NOPULL;

GPIO\_InitStruct.Speed = GPIO\_SPEED\_FREQ\_LOW;

HAL\_GPIO\_Init(GPIOB, &GPIO\_InitStruct);

/\*Configure GPIO pins : DFSDM1\_DATIN2\_Pin DFSDM1\_CKOUT\_Pin \*/

GPIO\_InitStruct.Pin = DFSDM1\_DATIN2\_Pin|DFSDM1\_CKOUT\_Pin;

GPIO\_InitStruct.Mode = GPIO\_MODE\_AF\_PP;

GPIO\_InitStruct.Pull = GPIO\_NOPULL;

GPIO\_InitStruct.Speed = GPIO\_SPEED\_FREQ\_LOW;

GPIO\_InitStruct.Alternate = GPIO\_AF6\_DFSDM1;

HAL\_GPIO\_Init(GPIOE, &GPIO\_InitStruct);

/\*Configure GPIO pins : QUADSPI\_CLK\_Pin QUADSPI\_NCS\_Pin OQUADSPI\_BK1\_IO0\_Pin QUADSPI\_BK1\_IO1\_Pin

QUAD\_SPI\_BK1\_IO2\_Pin QUAD\_SPI\_BK1\_IO3\_Pin \*/

GPIO\_InitStruct.Pin = QUADSPI\_CLK\_Pin|QUADSPI\_NCS\_Pin|OQUADSPI\_BK1\_IO0\_Pin|QUADSPI\_BK1\_IO1\_Pin

|QUAD\_SPI\_BK1\_IO2\_Pin|QUAD\_SPI\_BK1\_IO3\_Pin;

GPIO\_InitStruct.Mode = GPIO\_MODE\_AF\_PP;

GPIO\_InitStruct.Pull = GPIO\_NOPULL;

GPIO\_InitStruct.Speed = GPIO\_SPEED\_FREQ\_VERY\_HIGH;

GPIO\_InitStruct.Alternate = GPIO\_AF10\_QUADSPI;

HAL\_GPIO\_Init(GPIOE, &GPIO\_InitStruct);

/\*Configure GPIO pins : INTERNAL\_I2C2\_SCL\_Pin INTERNAL\_I2C2\_SDA\_Pin \*/

GPIO\_InitStruct.Pin = INTERNAL\_I2C2\_SCL\_Pin|INTERNAL\_I2C2\_SDA\_Pin;

GPIO\_InitStruct.Mode = GPIO\_MODE\_AF\_OD;

GPIO\_InitStruct.Pull = GPIO\_NOPULL;

GPIO\_InitStruct.Speed = GPIO\_SPEED\_FREQ\_VERY\_HIGH;

GPIO\_InitStruct.Alternate = GPIO\_AF4\_I2C2;

HAL\_GPIO\_Init(GPIOB, &GPIO\_InitStruct);

/\*Configure GPIO pins : INTERNAL\_UART3\_TX\_Pin INTERNAL\_UART3\_RX\_Pin \*/

GPIO\_InitStruct.Pin = INTERNAL\_UART3\_TX\_Pin|INTERNAL\_UART3\_RX\_Pin;

GPIO\_InitStruct.Mode = GPIO\_MODE\_AF\_PP;

GPIO\_InitStruct.Pull = GPIO\_NOPULL;

GPIO\_InitStruct.Speed = GPIO\_SPEED\_FREQ\_VERY\_HIGH;

GPIO\_InitStruct.Alternate = GPIO\_AF7\_USART3;

HAL\_GPIO\_Init(GPIOD, &GPIO\_InitStruct);

/\*Configure GPIO pins : LPS22HB\_INT\_DRDY\_EXTI0\_Pin LSM6DSL\_INT1\_EXTI11\_Pin ARD\_D2\_Pin HTS221\_DRDY\_EXTI15\_Pin

PMOD\_IRQ\_EXTI12\_Pin \*/

GPIO\_InitStruct.Pin = LPS22HB\_INT\_DRDY\_EXTI0\_Pin|LSM6DSL\_INT1\_EXTI11\_Pin|ARD\_D2\_Pin|HTS221\_DRDY\_EXTI15\_Pin

|PMOD\_IRQ\_EXTI12\_Pin;

GPIO\_InitStruct.Mode = GPIO\_MODE\_IT\_RISING;

GPIO\_InitStruct.Pull = GPIO\_NOPULL;

HAL\_GPIO\_Init(GPIOD, &GPIO\_InitStruct);

/\*Configure GPIO pins : USB\_OTG\_FS\_PWR\_EN\_Pin SPBTLE\_RF\_SPI3\_CSN\_Pin PMOD\_RESET\_Pin STSAFE\_A100\_RESET\_Pin \*/

GPIO\_InitStruct.Pin = USB\_OTG\_FS\_PWR\_EN\_Pin|SPBTLE\_RF\_SPI3\_CSN\_Pin|PMOD\_RESET\_Pin|STSAFE\_A100\_RESET\_Pin;

GPIO\_InitStruct.Mode = GPIO\_MODE\_OUTPUT\_PP;

GPIO\_InitStruct.Pull = GPIO\_NOPULL;

GPIO\_InitStruct.Speed = GPIO\_SPEED\_FREQ\_LOW;

HAL\_GPIO\_Init(GPIOD, &GPIO\_InitStruct);

/\*Configure GPIO pins : VL53L0X\_XSHUT\_Pin LED3\_WIFI\_\_LED4\_BLE\_Pin \*/

GPIO\_InitStruct.Pin = VL53L0X\_XSHUT\_Pin|LED3\_WIFI\_\_LED4\_BLE\_Pin;

GPIO\_InitStruct.Mode = GPIO\_MODE\_OUTPUT\_PP;

GPIO\_InitStruct.Pull = GPIO\_NOPULL;

GPIO\_InitStruct.Speed = GPIO\_SPEED\_FREQ\_LOW;

HAL\_GPIO\_Init(GPIOC, &GPIO\_InitStruct);

/\*Configure GPIO pins : VL53L0X\_GPIO1\_EXTI7\_Pin LSM3MDL\_DRDY\_EXTI8\_Pin \*/

GPIO\_InitStruct.Pin = VL53L0X\_GPIO1\_EXTI7\_Pin|LSM3MDL\_DRDY\_EXTI8\_Pin;

GPIO\_InitStruct.Mode = GPIO\_MODE\_IT\_RISING;

GPIO\_InitStruct.Pull = GPIO\_NOPULL;

HAL\_GPIO\_Init(GPIOC, &GPIO\_InitStruct);

/\*Configure GPIO pin : USB\_OTG\_FS\_VBUS\_Pin \*/

GPIO\_InitStruct.Pin = USB\_OTG\_FS\_VBUS\_Pin;

GPIO\_InitStruct.Mode = GPIO\_MODE\_INPUT;

GPIO\_InitStruct.Pull = GPIO\_NOPULL;

HAL\_GPIO\_Init(USB\_OTG\_FS\_VBUS\_GPIO\_Port, &GPIO\_InitStruct);

/\*Configure GPIO pins : USB\_OTG\_FS\_ID\_Pin USB\_OTG\_FS\_DM\_Pin USB\_OTG\_FS\_DP\_Pin \*/

GPIO\_InitStruct.Pin = USB\_OTG\_FS\_ID\_Pin|USB\_OTG\_FS\_DM\_Pin|USB\_OTG\_FS\_DP\_Pin;

GPIO\_InitStruct.Mode = GPIO\_MODE\_AF\_PP;

GPIO\_InitStruct.Pull = GPIO\_NOPULL;

GPIO\_InitStruct.Speed = GPIO\_SPEED\_FREQ\_VERY\_HIGH;

GPIO\_InitStruct.Alternate = GPIO\_AF10\_OTG\_FS;

HAL\_GPIO\_Init(GPIOA, &GPIO\_InitStruct);

/\*Configure GPIO pins : INTERNAL\_SPI3\_SCK\_Pin INTERNAL\_SPI3\_MISO\_Pin INTERNAL\_SPI3\_MOSI\_Pin \*/

GPIO\_InitStruct.Pin = INTERNAL\_SPI3\_SCK\_Pin|INTERNAL\_SPI3\_MISO\_Pin|INTERNAL\_SPI3\_MOSI\_Pin;

GPIO\_InitStruct.Mode = GPIO\_MODE\_AF\_PP;

GPIO\_InitStruct.Pull = GPIO\_NOPULL;

GPIO\_InitStruct.Speed = GPIO\_SPEED\_FREQ\_VERY\_HIGH;

GPIO\_InitStruct.Alternate = GPIO\_AF6\_SPI3;

HAL\_GPIO\_Init(GPIOC, &GPIO\_InitStruct);

/\*Configure GPIO pin : PMOD\_SPI2\_SCK\_Pin \*/

GPIO\_InitStruct.Pin = PMOD\_SPI2\_SCK\_Pin;

GPIO\_InitStruct.Mode = GPIO\_MODE\_AF\_PP;

GPIO\_InitStruct.Pull = GPIO\_NOPULL;

GPIO\_InitStruct.Speed = GPIO\_SPEED\_FREQ\_VERY\_HIGH;

GPIO\_InitStruct.Alternate = GPIO\_AF5\_SPI2;

HAL\_GPIO\_Init(PMOD\_SPI2\_SCK\_GPIO\_Port, &GPIO\_InitStruct);

/\*Configure GPIO pins : PMOD\_UART2\_CTS\_Pin PMOD\_UART2\_RTS\_Pin PMOD\_UART2\_TX\_Pin PMOD\_UART2\_RX\_Pin \*/

GPIO\_InitStruct.Pin = PMOD\_UART2\_CTS\_Pin|PMOD\_UART2\_RTS\_Pin|PMOD\_UART2\_TX\_Pin|PMOD\_UART2\_RX\_Pin;

GPIO\_InitStruct.Mode = GPIO\_MODE\_AF\_PP;

GPIO\_InitStruct.Pull = GPIO\_NOPULL;

GPIO\_InitStruct.Speed = GPIO\_SPEED\_FREQ\_VERY\_HIGH;

GPIO\_InitStruct.Alternate = GPIO\_AF7\_USART2;

HAL\_GPIO\_Init(GPIOD, &GPIO\_InitStruct);

/\*Configure GPIO pins : ST\_LINK\_UART1\_TX\_Pin ST\_LINK\_UART1\_RX\_Pin \*/

GPIO\_InitStruct.Pin = ST\_LINK\_UART1\_TX\_Pin|ST\_LINK\_UART1\_RX\_Pin;

GPIO\_InitStruct.Mode = GPIO\_MODE\_AF\_PP;

GPIO\_InitStruct.Pull = GPIO\_NOPULL;

GPIO\_InitStruct.Speed = GPIO\_SPEED\_FREQ\_VERY\_HIGH;

GPIO\_InitStruct.Alternate = GPIO\_AF7\_USART1;

HAL\_GPIO\_Init(GPIOB, &GPIO\_InitStruct);

/\*Configure GPIO pins : ARD\_D15\_Pin ARD\_D14\_Pin \*/

GPIO\_InitStruct.Pin = ARD\_D15\_Pin|ARD\_D14\_Pin;

GPIO\_InitStruct.Mode = GPIO\_MODE\_AF\_OD;

GPIO\_InitStruct.Pull = GPIO\_NOPULL;

GPIO\_InitStruct.Speed = GPIO\_SPEED\_FREQ\_VERY\_HIGH;

GPIO\_InitStruct.Alternate = GPIO\_AF4\_I2C1;

HAL\_GPIO\_Init(GPIOB, &GPIO\_InitStruct);

/\* EXTI interrupt init\*/

HAL\_NVIC\_SetPriority(*EXTI9\_5\_IRQn*, 0, 0);

HAL\_NVIC\_EnableIRQ(*EXTI9\_5\_IRQn*);

HAL\_NVIC\_SetPriority(*EXTI15\_10\_IRQn*, 0, 0);

HAL\_NVIC\_EnableIRQ(*EXTI15\_10\_IRQn*);

}

/\* USER CODE BEGIN 4 \*/

/\* USER CODE END 4 \*/

/\*\*

\* @brief This function is executed in case of error occurrence.

\* @retval None

\*/

**void** **Error\_Handler**(**void**)

{

/\* USER CODE BEGIN Error\_Handler\_Debug \*/

/\* User can add his own implementation to report the HAL error return state \*/

\_\_disable\_irq();

**while** (1)

{

}

/\* USER CODE END Error\_Handler\_Debug \*/

}

**#ifdef** USE\_FULL\_ASSERT

/\*\*

\* @brief Reports the name of the source file and the source line number

\* where the assert\_param error has occurred.

\* @param file: pointer to the source file name

\* @param line: assert\_param error line source number

\* @retval None

\*/

**void** assert\_failed(uint8\_t \*file, uint32\_t line)

{

/\* USER CODE BEGIN 6 \*/

/\* User can add his own implementation to report the file name and line number,

ex: printf("Wrong parameters value: file %s on line %d\r\n", file, line) \*/

/\* USER CODE END 6 \*/

}

**#endif** /\* USE\_FULL\_ASSERT \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* (C) COPYRIGHT STMicroelectronics \*\*\*\*\*END OF FILE\*\*\*\*/