/\*\*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* File Name : main.c  
\* Description : Main program body  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\*  
\* COPYRIGHT(c) 2017 STMicroelectronics  
\*  
\* Redistribution and use in source and binary forms, with or without modification,  
\* are permitted provided that the following conditions are met:  
\* 1. Redistributions of source code must retain the above copyright notice,  
\* this list of conditions and the following disclaimer.  
\* 2. Redistributions in binary form must reproduce the above copyright notice,  
\* this list of conditions and the following disclaimer in the documentation  
\* and/or other materials provided with the distribution.  
\* 3. Neither the name of STMicroelectronics nor the names of its contributors  
\* may be used to endorse or promote products derived from this software  
\* without specific prior written permission.  
\*  
\* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"  
\* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE  
\* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE  
\* DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE  
\* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL  
\* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR  
\* SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER  
\* CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,  
\* OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE  
\* OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.  
\*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\*/  
/\* Includes ------------------------------------------------------------------\*/  
#include "main.h"  
#include "stm32f0xx\_hal.h"  
  
/\* USER CODE BEGIN Includes \*/  
  
/\* USER CODE END Includes \*/  
  
/\* Private variables ---------------------------------------------------------\*/  
  
/\* USER CODE BEGIN PV \*/  
/\* Private variables ---------------------------------------------------------\*/  
  
/\* USER CODE END PV \*/  
  
/\* Private function prototypes -----------------------------------------------\*/  
void SystemClock\_Config(void);  
void Error\_Handler(void);  
  
/\* USER CODE BEGIN PFP \*/  
/\* Private function prototypes -----------------------------------------------\*/  
  
/\* USER CODE END PFP \*/  
  
/\* USER CODE BEGIN 0 \*/  
  
/\* USER CODE END 0 \*/  
  
int main(void)  
{  
HAL\_Init(); // Reset of all peripherals, init the Flash and Systick  
SystemClock\_Config(); //Configure the system clock  
  
/\* Enable Peripheral Clocks in RCC ---------------------------------------\*/  
RCC->AHBENR |= RCC\_AHBENR\_GPIOAEN;  
RCC->AHBENR |= RCC\_AHBENR\_GPIOCEN;  
RCC->APB2ENR |= RCC\_APB2ENR\_SYSCFGCOMPEN;  
  
/\* Configure GPIOC (LED) Pins --------------------------------------------\*/  
GPIOC->MODER |= (1 << 18) | (1 << 16) | (1 << 14) | (1 << 12); // Set PC9-PC6 to output mode  
GPIOC->BSRR = (1 << 8) | (1 << 6); // Set PC6 & PC8 high (BSRR doesn't need bitwise)  
  
/\* Configure GPIOA (Button) Pin ------------------------------------------\*/  
GPIOA->PUPDR |= (1 << 1); // Set PA0 to input mode, low speed and pull-down  
  
/\* Configure the SysTick Timer and Interrupt -----------------------------\*/  
SysTick\_Config(HAL\_RCC\_GetHCLKFreq()/1000);   
NVIC\_SetPriority(SysTick\_IRQn, 0); // Change to priority 2 to allow the EXTI to starve the SysTick.  
  
/\* Configure the EXTI PA0 Interrupt ---------------------------------------\*/  
SYSCFG->EXTICR[0] |= SYSCFG\_EXTICR1\_EXTI0\_PA;  
EXTI->IMR |= (1 << 0);  
EXTI->RTSR |= (1 << 0);  
NVIC\_SetPriority(EXTI0\_1\_IRQn, 1);  
NVIC\_EnableIRQ(EXTI0\_1\_IRQn);  
  
while (1) {  
\_\_WFI(); // Put processor to sleep until interrupt  
}   
}  
  
/\*\* System Clock Configuration  
\*/  
void SystemClock\_Config(void)  
{  
  
RCC\_OscInitTypeDef RCC\_OscInitStruct;  
RCC\_ClkInitTypeDef RCC\_ClkInitStruct;  
  
/\*\*Initializes the CPU, AHB and APB busses clocks   
\*/  
RCC\_OscInitStruct.OscillatorType = RCC\_OSCILLATORTYPE\_HSI;  
RCC\_OscInitStruct.HSIState = RCC\_HSI\_ON;  
RCC\_OscInitStruct.HSICalibrationValue = 16;  
RCC\_OscInitStruct.PLL.PLLState = RCC\_PLL\_NONE;  
if (HAL\_RCC\_OscConfig(&RCC\_OscInitStruct) != HAL\_OK)  
{  
Error\_Handler();  
}  
  
/\*\*Initializes the CPU, AHB and APB busses clocks   
\*/  
RCC\_ClkInitStruct.ClockType = RCC\_CLOCKTYPE\_HCLK|RCC\_CLOCKTYPE\_SYSCLK  
|RCC\_CLOCKTYPE\_PCLK1;  
RCC\_ClkInitStruct.SYSCLKSource = RCC\_SYSCLKSOURCE\_HSI;  
RCC\_ClkInitStruct.AHBCLKDivider = RCC\_SYSCLK\_DIV1;  
RCC\_ClkInitStruct.APB1CLKDivider = RCC\_HCLK\_DIV1;  
  
if (HAL\_RCC\_ClockConfig(&RCC\_ClkInitStruct, FLASH\_LATENCY\_0) != HAL\_OK)  
{  
Error\_Handler();  
}  
  
/\*\*Configure the Systick interrupt time   
\*/  
//HAL\_SYSTICK\_Config(HAL\_RCC\_GetHCLKFreq()/1000);  
  
/\*\*Configure the Systick   
\*/  
HAL\_SYSTICK\_CLKSourceConfig(SYSTICK\_CLKSOURCE\_HCLK);  
  
/\* SysTick\_IRQn interrupt configuration \*/  
//HAL\_NVIC\_SetPriority(SysTick\_IRQn, 0, 0);  
}  
  
/\* USER CODE BEGIN 4 \*/  
  
/\* USER CODE END 4 \*/  
  
/\*\*  
\* @brief This function is executed in case of error occurrence.  
\* @param None  
\* @retval None  
\*/  
void Error\_Handler(void)  
{  
/\* USER CODE BEGIN Error\_Handler \*/  
/\* User can add his own implementation to report the HAL error return state \*/  
while(1)   
{  
}  
/\* USER CODE END Error\_Handler \*/   
}  
  
#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  
  
/\*\*  
\* @}  
\*/   
  
/\*\*  
\* @}  
\*/   
  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* (C) COPYRIGHT STMicroelectronics \*\*\*\*\*END OF FILE\*\*\*\*/