```c
/* USER CODE BEGIN Header */
/**
  ******************************************************************************
  * @file    stm32f0xx_it.c
  * @brief   Interrupt Service Routines.
  ******************************************************************************
  * @attention
  *
  * Copyright (c) 2022 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 "stm32f0xx_it.h"
/* Private includes ----------------------------------------------------------*/
/* USER CODE BEGIN Includes */
/* USER CODE END Includes */

/* Private typedef -----------------------------------------------------------*/
/* USER CODE BEGIN TD */

/* USER CODE END TD */

/* 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 -----------------------------------------------*/
/* USER CODE BEGIN PFP */

/* USER CODE END PFP */

/* Private user code ---------------------------------------------------------*/
/* USER CODE BEGIN 0 */

/* USER CODE END 0 */

/* External variables --------------------------------------------------------*/

/* USER CODE BEGIN EV */
/* USER CODE END EV */

/******************************************************************************/
/*           Cortex-M0 Processor Interruption and Exception Handlers          */
/******************************************************************************/



void EXTI0_1_IRQHandler(void)
{
      EXTI->PR |= EXTI_PR_PIF0;     /* "ACK" the interrupt */
}

char RecvChar(USART_TypeDef *Def);
void WriteChar(USART_TypeDef *Def, char Cur);
void SetColor(char c, char val);



static char Count[2];
static uint8_t Status = 0;

void WriteAsyncChar(USART_TypeDef *Def, char Cur)
{
      Def->TDR = Cur;
}

void USART3_4_IRQHandler(void)
{
      if ((USART3->ISR & USART_ISR_RXNE) == USART_ISR_RXNE)
      {
            Count[Status] = USART3->RDR;
            Status++;
      }

      if (Status == 2)
      {
            if (Count[1] == '0' || Count[1] == '1')
            {
                  SetColor(Count[0], Count[1] == '1');
                  Status = 0;
            }
            else
            {
                  SetColor('e', 0); /* Submit an error */
                  Status = 0;
            }
      }
      USART3->ICR |= (USART_ISR_ORE | USART_ISR_RXNE);
}



/**
  * @brief This function handles Non maskable interrupt.
  */
void NMI_Handler(void)
{
  /* USER CODE BEGIN NonMaskableInt_IRQn 0 */

  /* USER CODE END NonMaskableInt_IRQn 0 */
  /* USER CODE BEGIN NonMaskableInt_IRQn 1 */
  while (1)
  {
  }
  /* USER CODE END NonMaskableInt_IRQn 1 */
}

/**
  * @brief This function handles Hard fault interrupt.
  */
void HardFault_Handler(void)
{
  /* USER CODE BEGIN HardFault_IRQn 0 */

  /* USER CODE END HardFault_IRQn 0 */
  while (1)
  {
    /* USER CODE BEGIN W1_HardFault_IRQn 0 */
    /* USER CODE END W1_HardFault_IRQn 0 */
  }
}

/**
  * @brief This function handles System service call via SWI instruction.
  */
void SVC_Handler(void)
{
  /* USER CODE BEGIN SVC_IRQn 0 */

  /* USER CODE END SVC_IRQn 0 */
  /* USER CODE BEGIN SVC_IRQn 1 */

  /* USER CODE END SVC_IRQn 1 */
}

/**
  * @brief This function handles Pendable request for system service.
  */
void PendSV_Handler(void)
{
  /* USER CODE BEGIN PendSV_IRQn 0 */

  /* USER CODE END PendSV_IRQn 0 */
  /* USER CODE BEGIN PendSV_IRQn 1 */

  /* USER CODE END PendSV_IRQn 1 */
}

/**
  * @brief This function handles System tick timer.
  */
void SysTick_Handler(void)
{
  /* USER CODE BEGIN SysTick_IRQn 0 */

  /* USER CODE END SysTick_IRQn 0 */
  HAL_IncTick();
  /* USER CODE BEGIN SysTick_IRQn 1 */

  /* USER CODE END SysTick_IRQn 1 */
}

/******************************************************************************/
/* STM32F0xx Peripheral Interrupt Handlers                                    */
/* Add here the Interrupt Handlers for the used peripherals.                  */
/* For the available peripheral interrupt handler names,                      */
/* please refer to the startup file (startup_stm32f0xx.s).                    */
/******************************************************************************/

/* USER CODE BEGIN 1 */

/* USER CODE END 1 */
```