实验6：使用CUBE进行中断实验

11610101 韦青茂

实验器材

- 硬件：ARM-STM32开发板，J-Link/St-Link。
- 软件：Win7/Win8/Win10, Keil uVision5

实验要求

1. Use **EXTI** to control the LED: Press **KEY0** to blink **LED0** three times, press **KEY1** to blink **LED1** three times, and transmit the corresponding message by **UART**.
2. Receive **UART** data in **non-blocking** mode: when the **UART** receives the text interrupt, transmits the corresponding message by **UART**.

实验过程

软件代码(main.c)

```
/* USER CODE BEGIN Header */
/**
  ******************************************************************************
  * @file           : main.c
  * @brief          : Main program body
  ******************************************************************************
  * @attention
  *
  * <h2><center>&copy; Copyright (c) 2019 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"
#include "usart.h"
#include "gpio.h"

/* Private includes ----------------------------------------------------------*/
/* USER CODE BEGIN Includes */
#include "string.h"
/* USER CODE END Includes */

/* Private typedef -----------------------------------------------------------*/
/* USER CODE BEGIN PTD */
```

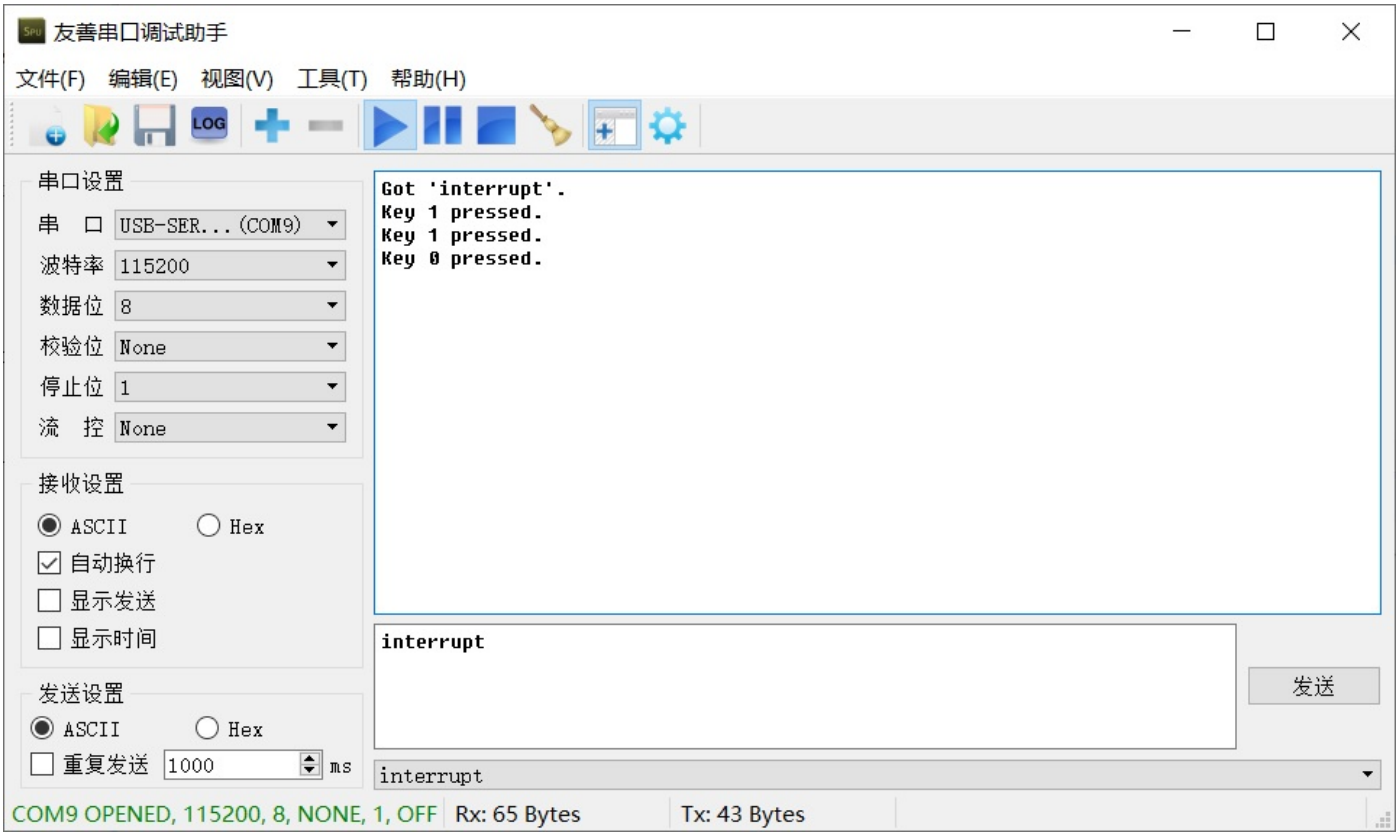
```
HAL_StatusTypeDef HAL_UART_Receive_IT(UART_HandleTypeDef *huart, uint8_t *pData,
uint16_t Size);
/* USER CODE END PTD */

/* 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 */
uint8_t rxBuffer[20];
extern UART_HandleTypeDef huart1;
 uint8_t key0Pressed[] = "Key 0 pressed.\n";
 uint8_t key1Pressed[] = "Key 1 pressed.\n";
 char textForIRQ[] = "interrupt";
/* USER CODE END PV */

/* Private function prototypes -----------------------------------------------*/
void SystemClock_Config(void);
/* USER CODE BEGIN PFP */
void BlinkLed0();
void BlinkLed1();
/* 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();
  MX_USART1_UART_Init();
  /* USER CODE BEGIN 2 */
  HAL_UART_Receive_IT(&huart1, (uint8_t *)rxBuffer, 1);
  /* USER CODE END 2 */

  /* Infinite loop */
  /* USER CODE BEGIN WHILE */
  while (1)
  {
    /* USER CODE END WHILE */

    /* 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};

  /** Initializes the CPU, AHB and APB busses clocks
  */
  RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSI;
  RCC_OscInitStruct.HSIState = RCC_HSI_ON;
  RCC_OscInitStruct.HSICalibrationValue = RCC_HSICALIBRATION_DEFAULT;
  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_CLOCKTYPE_PCLK2;
  RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_HSI;
  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_0) != HAL_OK)
  {
    Error_Handler();
  }
}

/* USER CODE BEGIN 4 */
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{
        HAL_Delay(100);
        switch (GPIO_Pin) {
        case KEY0_Pin:
                HAL_Delay(10);
                if (HAL_GPIO_ReadPin(KEY0_GPIO_Port, KEY0_Pin) == GPIO_PIN_RESET) {
                        HAL_UART_Transmit(&huart1, key0Pressed, 16, 0xffff);
```

```
                        BlinkLed0();
                }
                break;
        case KEY1_Pin:
                HAL_Delay(10);
                if (HAL_GPIO_ReadPin(KEY1_GPIO_Port, KEY1_Pin) == GPIO_PIN_RESET) {
                        HAL_UART_Transmit(&huart1, key1Pressed, 16, 0xffff);
                        BlinkLed1();
                }
                break;
        case KEY_WK_Pin:
                HAL_Delay(10);
                if (HAL_GPIO_ReadPin(KEY_WK_GPIO_Port, KEY_WK_Pin) == GPIO_PIN_SET) {
                        HAL_GPIO_TogglePin(LED0_GPIO_Port, LED0_Pin);
                        HAL_GPIO_TogglePin(LED1_GPIO_Port, LED1_Pin);
                }
                break;
        default:
                break;

        }
}

void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart)
{
        if(huart->Instance==USART1)
        {
        // 要打印的信息
                static uint8_t respond[] = "Got 'interrupt'.\n";
                static unsigned char uRx_Data[1024] = {0};
                static unsigned char uLength = 0;
                if(rxBuffer[0] == '\n' || rxBuffer[0] == '\0')
                {
            // 把末尾的'\n'或者'\r'替换为'\0'
                        uRx_Data[uLength-1] = '\0';
                        if(!strcmp(uRx_Data, textForIRQ)){
                                HAL_UART_Transmit(&huart1, respond, 17, 0xffff);
                        }
                        uLength = 0;
                }
                else
                {
                        uRx_Data[uLength] = rxBuffer[0];
                        uLength++;
                }
        }
}

// 闪烁LED0
void BlinkLed0(){
        HAL_GPIO_WritePin(LED0_GPIO_Port, LED0_Pin, GPIO_PIN_RESET);
        for(int i=0;i<7;i++){
                HAL_GPIO_TogglePin(LED0_GPIO_Port, LED0_Pin);
                HAL_Delay(200);
        }
}
// 闪烁LED1
void BlinkLed1(){
        HAL_GPIO_WritePin(LED1_GPIO_Port, LED1_Pin, GPIO_PIN_RESET);
        for(int i=0;i<7;i++){
                HAL_GPIO_TogglePin(LED1_GPIO_Port, LED1_Pin);
                HAL_Delay(200);
```

```
        }
}
/* 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 */

  /* 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,
     tex: 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****/
```

串口实验：



遇到的问题及解决方法

1. 串口明明发送了"interrupt"却没有匹配成功

因为串口软件发送时会自动在末尾加一个'\r',需要将它删掉

```
// 把末尾的'\n'或者'\r'替换为'\0'
uRx_Data[uLength-1] = '\0';
```

这样打印该字符串时就会以这个手动添加的'\0'作为结束符