МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

АЭРОКОСМИЧЕСКОГО ПРИБО6РОСТРОЕНИЯ»

КАФЕДРА № 43

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| доцент |  |  |  | А.А. Попов |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ |
| «*Настройка системы прерываний микроконтроллера*» |
| по курсу: Программирование встроенных приложений |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4136 |  |  |  | Н.С. Бобрович |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2024

**1. Цель работы**

Привитие практических навыков по работе с системой прерываний ядра ARMv7-M, библиотекой CMSIS, технической документацией.

**2. Задание**

Таблица 1 – Варианты заданий лабораторной работы №4.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Вариант | Номер линии PC | Количество групп и подгрупп приоритетов | | Номера приоритетов до настройки группировки приоритетов и после, соответственно порядку линий | | Приоритет SysTick |
| Групп | Подгрупп | до | После №группы(№подгруппы) |

Скриншот 22-05-2024 132321

1. Изучить подраздел B3.4 (стр. B3-624) руководства ARMv7-M и раздел №14 руководства RM0316.
2. Используя библиотеку CMSIS, написать и отладить работу программы на языке «си», в которую входит настройка внешних прерываний (EXTI) на четыре линии ПВВ C, согласно номеру варианта таблицы 1. Настроить приоритет системного таймера SysTick и в дальнейшем не изменять.
3. Не меняя приоритеты и группировку приоритетов (оставляем по умолчанию после сброса) проверить в режиме отладки порядок обработки:
   * одновременно установленных четырёх внешних прерываний (в точке останова – Stop, нажатием кнопок установить четыре прерывания одновременно, делаем шаг отладки – Step (F11), продолжаем выполнение программы – Run (F5));
   * поступления прерываний последовательно (последовательно нажимаем нужные кнопки).

Отразить в выводах выявленный порядок обработки прерываний: внешних – от кнопок; внутреннего – от системного таймера, прерывают ли прерывания друг друга и если да, то в каком порядке.

1. Не меняя группировку приоритетов, установить приоритеты согласно варианту таблицы 1 (столбец «до»), исследовать изменение порядка обработки прерываний аналогично п. 3).
2. Настроить приоритеты прерываний, используя группы и подгруппы, согласно варианту таблицы 1 (столбец «после»), исследовать изменение порядка обработки прерываний аналогично п. 3).

**3. Исходный код программы**

**Файл .c:**

#include "RTE\_Components.h" // Component selection

#include CMSIS\_device\_header // Device header

#include <stdio.h>

static volatile uint32\_t ui\_count100ms = 0;

void delay(void) {

volatile uint32\_t i = 6000000;

while (i > 0)

i--;

}

int main(void) {

uint32\_t priGroup = 0, PreemptPriority = 0, SubPriority = 0;

//настройка частоты 72 МГц

SET\_BIT(RCC->CR, RCC\_CR\_HSEON);

while ((RCC->CR & RCC\_CR\_HSERDY) == 0) {}

FLASH->ACR = FLASH\_ACR\_PRFTBE | FLASH\_ACR\_LATENCY\_1;

RCC->CFGR |= (uint32\_t)(RCC\_CFGR\_PLLSRC\_HSE\_PREDIV | RCC\_CFGR\_PLLMUL9);

SET\_BIT(RCC->CR, RCC\_CR\_PLLON);

while ((RCC->CR & RCC\_CR\_PLLRDY) == 0) {}

RCC->CFGR &= (uint32\_t)((uint32\_t)~(RCC\_CFGR\_SW));

RCC->CFGR |= (uint32\_t)RCC\_CFGR\_SW\_PLL;

while ((RCC->CFGR & (uint32\_t)RCC\_CFGR\_SWS) != (uint32\_t)RCC\_CFGR\_SWS\_PLL) {}

SystemCoreClockUpdate();//проверяем частоту SystemCoreClock

printf("clk=%d\n", SystemCoreClock);

SET\_BIT(RCC->APB2ENR, RCC\_APB2ENR\_SYSCFGEN);//разрешаем тактирование SYSCFG

SET\_BIT(RCC->AHBENR, RCC\_AHBENR\_GPIOCEN); //GPIOC

CLEAR\_BIT(GPIOC->MODER, GPIO\_MODER\_MODER0 | GPIO\_MODER\_MODER2 |

GPIO\_MODER\_MODER3 | GPIO\_MODER\_MODER4); //PC0,2,3,4 In

SET\_BIT(GPIOC->PUPDR, GPIO\_PUPDR\_PUPDR0\_0 | GPIO\_PUPDR\_PUPDR2\_0 |

GPIO\_PUPDR\_PUPDR3\_0 | GPIO\_PUPDR\_PUPDR4\_0);//Pull up PC0,2,3,4

SET\_BIT(GPIOC->MODER, GPIO\_MODER\_MODER1\_0 | GPIO\_MODER\_MODER5\_0 |

GPIO\_MODER\_MODER6\_0 | GPIO\_MODER\_MODER7\_0);//PC1,5,6,7 Out

SET\_BIT(GPIOC->OTYPER, GPIO\_OTYPER\_OT\_1 | GPIO\_OTYPER\_OT\_5 |

GPIO\_OTYPER\_OT\_6 | GPIO\_OTYPER\_OT\_7); //режим с открытым стоком

SET\_BIT(GPIOC->BRR, GPIO\_BRR\_BR\_1 | GPIO\_BRR\_BR\_5 | GPIO\_BRR\_BR\_6 | GPIO\_BRR\_BR\_7); //притягиваем к нулю

///ВСТАВКА

NVIC\_SetPriorityGrouping(5); //2 bits for group -- NVIC\_PriorityGroup\_2 (4group 4 sub)

priGroup = NVIC\_GetPriorityGrouping();

printf("Priority Group=%d\r\n", priGroup);

NVIC\_SetPriority(EXTI2\_TSC\_IRQn, 2 << 2 | 2);

NVIC\_DecodePriority(NVIC\_GetPriority(EXTI2\_TSC\_IRQn), priGroup, &PreemptPriority, &SubPriority);

printf("EXTI2 Preempt Priority=%d \tSubPriority=%d\r\n", PreemptPriority, SubPriority);

NVIC\_SetPriority(EXTI3\_IRQn, 1 << 2 | 1);

NVIC\_DecodePriority(NVIC\_GetPriority(EXTI3\_IRQn), priGroup, &PreemptPriority, &SubPriority);

printf("EXTI3 Preempt Priority=%d \tSubPriority=%d\r\n", PreemptPriority, SubPriority);

NVIC\_SetPriority(EXTI4\_IRQn, 1 << 2 | 0);

NVIC\_DecodePriority(NVIC\_GetPriority(EXTI4\_IRQn), priGroup, &PreemptPriority, &SubPriority);

printf("EXTI4 Preempt Priority=%d \tSubPriority=%d\r\n", PreemptPriority, SubPriority);

NVIC\_SetPriority(EXTI9\_5\_IRQn, 0 << 2 | 0);

NVIC\_DecodePriority(NVIC\_GetPriority(EXTI9\_5\_IRQn), priGroup, &PreemptPriority, &SubPriority);

printf("EXTI6 Preempt Priority=%d \tSubPriority=%d\r\n", PreemptPriority, SubPriority);

//ВСТАВКА КОНЕЦ

printf("Press any key\r\n");

//прерывание на спад сигнала

SET\_BIT(EXTI->FTSR, EXTI\_FTSR\_FT2 | EXTI\_FTSR\_FT3 | EXTI\_FTSR\_FT4 | EXTI\_FTSR\_FT6);

//разрешаем прерывания внешних линий 2,3,4,6

SET\_BIT(EXTI->IMR, EXTI\_IMR\_IM2 | EXTI\_IMR\_IM3 | EXTI\_IMR\_IM4 | EXTI\_IMR\_IM6);

// выбираем в качестве внешних входов EXTI линии:

//EXTI2=PC2 EXTI3=PC3 EXTI4=PC4 EXTI6=PC6

SYSCFG->EXTICR[0] = SYSCFG\_EXTICR1\_EXTI2\_PC | SYSCFG\_EXTICR1\_EXTI3\_PC;

SYSCFG->EXTICR[1] = SYSCFG\_EXTICR2\_EXTI4\_PC | SYSCFG\_EXTICR2\_EXTI6\_PC;

NVIC\_EnableIRQ(EXTI2\_TSC\_IRQn);

NVIC\_EnableIRQ(EXTI3\_IRQn);

NVIC\_EnableIRQ(EXTI4\_IRQn);

NVIC\_EnableIRQ(EXTI9\_5\_IRQn);

SysTick\_Config(0x6DDD00);//прерывание каждые 100мсек

NVIC\_SetPriority(SysTick\_IRQn, 6);

while (1) {}

}

void SysTick\_Handler(void)//обработчик прерывание системного таймера

{

ui\_count100ms++;

8

if (ui\_count100ms % 3 == 0)//выводим каждые 0,3 секунды

ITM\_SendChar('o');

}

void EXTI0\_TSC\_IRQHandler(void)

{

EXTI->PR = EXTI\_PR\_PR2;

ITM\_SendChar('0');

delay();

ITM\_SendChar('a');

ITM\_SendChar('\n');

}

void EXTI2\_IRQHandler(void)

{

EXTI->PR = EXTI\_PR\_PR3;

ITM\_SendChar('2');

delay();

ITM\_SendChar('b');

ITM\_SendChar('\n');

}

void EXTI3\_IRQHandler(void)

{

EXTI->PR = EXTI\_PR\_PR4;

ITM\_SendChar('3');

delay();

ITM\_SendChar('c');

ITM\_SendChar('\n');

}

void EXTI4\_IRQHandler(void)

{

EXTI->PR = EXTI\_PR\_PR6;

ITM\_SendChar('4');

delay();

ITM\_SendChar('d');

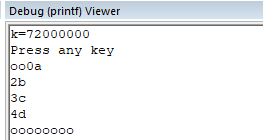
ITM\_SendChar('\n');

}

**4. Задание**

**Пункт 3:**

Согласно заданию установлены нулевые приоритеты (закомментированы строки NVIC\_SetPriority), группировка приоритетов отсутствует (закомментирована строка NVIC\_SetPriorityGrouping). Случай возникновения 4-х одновременных прерываний и порядок их обработки на снимке экрана:



Расшифровка:

Группировка приоритетов = 0 (без групп)

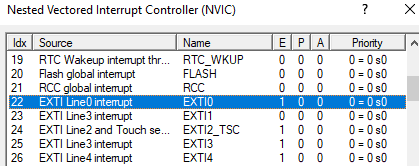
1 – начало EXTI1 | а – конец EXTI1

2 – начало EXTI2 | b – конец EXTI2

3 – начало EXTI3 | c – конец EXTI3

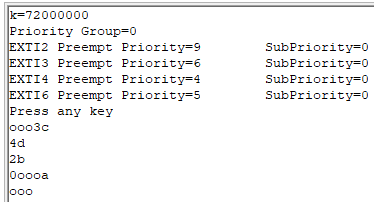
5 – начало EXTI5 | d – конец EXTI5

При последовательном поступлении прерываний обработка ведется в порядке поступления, последовательно, без прерываний друг друга.



**Пункт 4:**

Согласно заданию п. 4) установлены приоритеты EXTI0(PC1)=83, EXTI2(PC2)=98, EXTI3(PC3)=112, EXTI4(PC5)=97. Результат на снимке экрана:



Расшифровка:

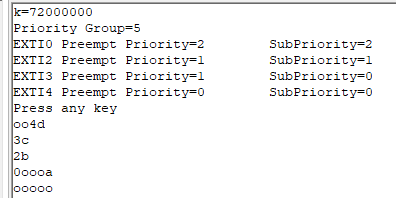
Группировка приоритетов = 0 (без групп)

При одновременном возникновении, последовательности обработки прерываний следующая: EXTI3>EXTI5>EXTI2>EXTI1

Выполняется согласно установленному приоритету.

**Пункт 5:**

Согласно заданию 5) установлена группировка приоритетов 2, 8 групп, в каждой 2 подгруппы.



Расшифровка:

Выполняются в порядке группировки: EXTI5>EXTI2>EXTI2>EXTI1

Выполняется согласно установленному приоритету.