ГУАП

КАФЕДРА № 43

ОТЧЕТ  
 ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| доц. канд. техн. наук |  |  |  | Попов А.А. |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

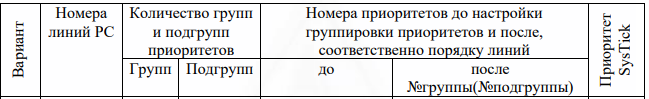
|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ |
| НАСТРОЙКА СИСТЕМЫ ПРЕРЫВАНИЙ МИКРОКОНТРОЛЛЕРА |
| по дисциплине: ПРОГРАММИРОВАНИЕ ВСТРОЕННЫХ ПРИЛОЖЕНИЙ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4134 |  | 15.04.2024 |  | Самарин Д. В. |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2024

1. **Вариант задания**



Следующая программа выполнена согласно требованиям п. 4) содержания работы.

**Листинг**

#include "RTE\_Components.h"

#include CMSIS\_device\_header

#include "stdio.h"

static volatile uint32\_t ui\_count100ms=0;

void delay(void){

volatile uint32\_t i=6000000;

while(i > 0)

i--;}

// K3 K7 K10 K15

//9 омера линий PC (0,1,3,5) Групп - 8 Подгрупп - 2 до (71,86,100,85)

//6(1), 7(0), 4(1), 6(0) Приоритет 6

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\_MODER1|

GPIO\_MODER\_MODER3|GPIO\_MODER\_MODER5); //PC0,1,3,5 In

SET\_BIT(GPIOC->PUPDR,GPIO\_PUPDR\_PUPDR0\_0|GPIO\_PUPDR\_PUPDR1\_0|

GPIO\_PUPDR\_PUPDR3\_0|GPIO\_PUPDR\_PUPDR5\_0);//Pull up PC0,1,3,5

SET\_BIT(GPIOC->MODER,GPIO\_MODER\_MODER2\_0|GPIO\_MODER\_MODER4\_0|

GPIO\_MODER\_MODER6\_0|GPIO\_MODER\_MODER7\_0);//PC2,4,6,7 Out

SET\_BIT(GPIOC->OTYPER, GPIO\_OTYPER\_OT\_2|GPIO\_OTYPER\_OT\_4|

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

SET\_BIT(GPIOC->BRR,GPIO\_BRR\_BR\_2|GPIO\_BRR\_BR\_4|GPIO\_BRR\_BR\_6|GPIO\_BRR\_BR\_7); //притягиваем к нулю

//1 бит под подгруппы и 4-1 = 3 бита под группы

//NVIC\_SetPriorityGrouping(4);

priGroup = NVIC\_GetPriorityGrouping();

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

//6(1)

//NVIC\_SetPriority(EXTI0\_IRQn,13);

//NVIC\_SetPriority(EXTI0\_IRQn,71);

NVIC\_DecodePriority(NVIC\_GetPriority(EXTI0\_IRQn),priGroup,&PreemptPriority,&SubPriority);

printf("EXTI0 Preempt Priority=%d \tSubPriority=%d\r\n",PreemptPriority,SubPriority);

//7 (0)

//NVIC\_SetPriority(EXTI1\_IRQn,14);

//NVIC\_SetPriority(EXTI1\_IRQn,86);

NVIC\_DecodePriority(NVIC\_GetPriority(EXTI1\_IRQn),priGroup,&PreemptPriority,&SubPriority);

printf("EXTI1 Preempt Priority=%d \tSubPriority=%d\r\n",PreemptPriority,SubPriority);

//4(1)

//NVIC\_SetPriority(EXTI3\_IRQn,9);

//NVIC\_SetPriority(EXTI3\_IRQn,100);

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

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

//6(0)

//NVIC\_SetPriority(EXTI9\_5\_IRQn,12);

//NVIC\_SetPriority(EXTI9\_5\_IRQn,80);

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

printf("EXTI9\_5 Preempt Priority=%d \tSubPriority=%d\r\n",PreemptPriority,SubPriority);

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

//прерывание на спад сигнала

SET\_BIT(EXTI->FTSR,EXTI\_FTSR\_FT0|EXTI\_FTSR\_FT1|EXTI\_FTSR\_FT3|EXTI\_FTSR\_FT5);

//разрешаем прерывания внешних линий 0,1,3,5

SET\_BIT(EXTI->IMR,EXTI\_IMR\_IM0|EXTI\_IMR\_IM1|EXTI\_IMR\_IM3|EXTI\_IMR\_IM5);

// выбираем в качестве внешних входов EXTI линии:

//EXTI0=PC0 EXTI1=PC1 EXTI3=PC3 EXTI5=PC5

SYSCFG->EXTICR[0]=SYSCFG\_EXTICR1\_EXTI0\_PC|SYSCFG\_EXTICR1\_EXTI1\_PC|SYSCFG\_EXTICR1\_EXTI3\_PC;

SYSCFG->EXTICR[1]=SYSCFG\_EXTICR2\_EXTI5\_PC;

NVIC\_EnableIRQ(EXTI0\_IRQn);

NVIC\_EnableIRQ(EXTI1\_IRQn);

NVIC\_EnableIRQ(EXTI3\_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++;

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

ITM\_SendChar('o');}

void EXTI0\_IRQHandler(void)

{

EXTI->PR = EXTI\_PR\_PR0;

ITM\_SendChar('0');

delay();

ITM\_SendChar('a');

ITM\_SendChar('\n');

}

void EXTI1\_IRQHandler(void)

{

EXTI->PR = EXTI\_PR\_PR1;

ITM\_SendChar('1');

delay();

ITM\_SendChar('b');

ITM\_SendChar('\n');

}

void EXTI3\_IRQHandler(void)

{

EXTI->PR = EXTI\_PR\_PR3;

ITM\_SendChar('3');

delay();

ITM\_SendChar('c');

ITM\_SendChar('\n');

}

void EXTI9\_5\_IRQHandler(void)

{

EXTI->PR = EXTI\_PR\_PR5;

ITM\_SendChar('5');

delay();

ITM\_SendChar('d');

ITM\_SendChar('\n');

}

/\*int main(void){

uint32\_t cnt=0;

int i=0;

ITM\_SendChar('s');

ITM\_SendChar('T');

ITM\_SendChar('M');

ITM\_SendChar('\n');

printf("Integer \tHexadecimal\n");

while(1)

{

printf("%08d\t%08X\n", cnt, cnt);

cnt++;

for(int i=0; i<0x100000; i++)

\_\_NOP();

}

}\*/

1. **Анализ порядка обработки прерываний системного таймера**

Анализ работы системы прерываний при различных настройках приоритетов без влияния системного таймера (строчка SysTick\_Config(0x6DDD00); – закомментирована):

1. Согласно заданию п. 3) 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:

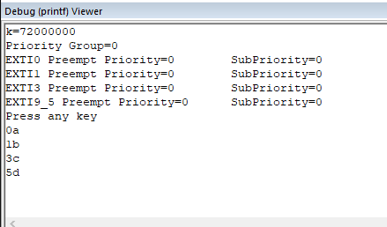


Рисунок 1 – демонстрация прерываний при нулевых приоритетах

Группировка приоритетов = 0 (без подгрупп)

Для линий внешних прерываний EXTI2, EXTI3, EXTI4, EXTI6 приоритеты равны нулю

0 - начало EXTI0 | a - конец EXTI0

1 - начало EXTI1 | b - конец EXTI1

3 - начало EXTI3 | c - конец EXTI3

5 - начало EXTI5 | d - конец EXTI5

При одновременном возникновении 4-х прерываний их обработка происходит в порядке: EXTI0(PC0) > EXTI1(PC1) > EXTI3(PC3) > EXTI5(PC5).

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

Демонстрация окна просмотра NVIC.

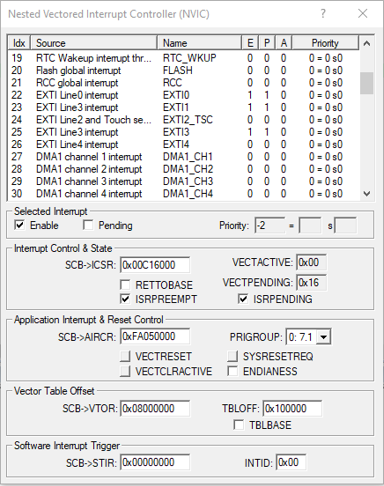


Рисунок 2 – демонстрация отложенных прерываний при одновременном нажатии 4-ех.

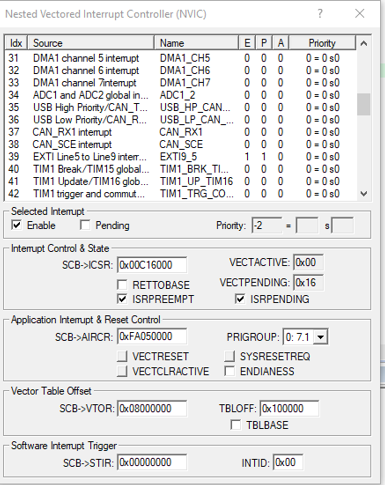


Рисунок 3 – демонстрация отложенных прерываний при одновременном нажатии 4-ех.

2. Согласно заданию п. 4) установлены приоритеты EXTI0(PC0)=71, EXTI1(PC1)=86, EXTI3(PC3)=100, EXTI5(PC5)=85. Результат на снимке экрана:

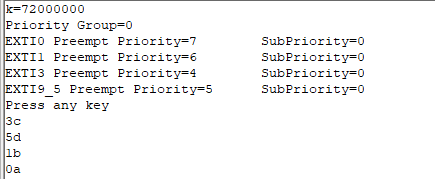


Рисунок 3 – демонстрация отложенных прерываний при установленных приоритетах и одновременном нажатии 4-ех.

При одновременном возникновении, последовательность обработки прерываний следующая: EXTI3 (4) > EXTI5 (5) EXTI1 (6) > EXTI0 (7)

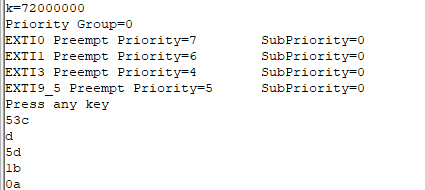


Рисунок 4 – демонстрация отложенных прерываний при установленных приоритетах и последовательном прерывании 4-ех.

При последовательном поступлении, прерывания прерывают работу друг друга в следующем порядке:

EXTI4 прерывает любые другие

EXTI5 прерывает EXTI1, EXTI0

EXTI1 прерывает EXTI0

1. Согласно заданию 5) установлена группировка приоритетов 7, одна группа, в которой 16 подгрупп. Результат на снимке экрана:

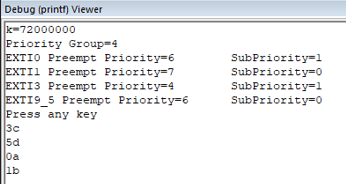


Рисунок 5 – демонстрация отложенных прерываний при установленных группах и субприоритетах.

Группировка приоритетов. 8 групп и 2 подгруппы.

Группа(подгруппа)

EXTI0 6(1)  
EXTI1 7(0)  
EXTI3 4(1)

EXTI5 6(0)

При одновременном возникновении, последовательность обработки прерываний следующая: EXTI3>EXTI5>EXTI0>EXTI1

При последовательном поступлении, при одинаковых группах, выполняется последовательно без прерываний, при разных последовательно согласно приоритету.

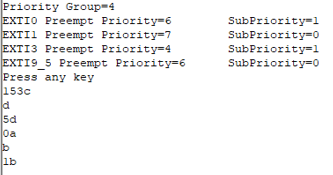


Рисунок 6 – демонстрация отложенных прерываний при установленных группах и субприоритетах.