**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет «ЛЭТИ»**

**Кафедра САУ**

отчет

**по лабораторной работе №5**

**по дисциплине «Операционные системы реального времени»**

**Тема: «ПРЕРЫВАНИЯ В ОС FreeRTOS»**

**1 - Варинат**

|  |  |  |
| --- | --- | --- |
| Студенты гр. 7492 |  | Дружинин А.А. |
|  |  | Репин П.О. |
| Преподаватель |  | Гречухин М.Н. |

Санкт-Петербург

2021

**Цель работы:** познакомятся с использованием прерываний в ОС FreeRTOS.

**Задание:**

В таблице 1 приведены варианты к лабораторной работе.

|  |  |  |
| --- | --- | --- |
| Вариант | Режим 1 | Режим 2 |
| 1 | Мигание диодом с частотой 2 Гц | Мигание диодом с частотой 10 Гц |
| 2 | Бегущий огонь по одному диоду влево | Бегущий огонь по одному диоду вправо |
| 3 | Бегущий огонь по два диода, сходящийся к центру | Бегущий огонь по два диода, расходящийся от центра |
| 4 | Мигание диодом сигнала SOS (3 коротких вспышки, три длинных вспышки, три коротких вспышки) | Отключение диода (диод не мигает) |

Таблица 1 – варианты заданий.

Создать две функции-задачи. Одна из них будет отвечать за визуализацию на выбранной периферии по указанным согласно таблице режимам. Вторая будет являться обработчиком прерываний (IST).

**Код программы:**

#include "stm32f4xx.h" // Device header

#include "FreeRTOS.h" // ARM.FreeRTOS::RTOS:Core

#include "FreeRTOSConfig.h" // ARM.FreeRTOS::RTOS:Config

#include "task.h" // ARM.FreeRTOS::RTOS:Core

#include "semphr.h"

#include "queue.h"

//xSemaphoreHandle x\_INTERRUPT;

SemaphoreHandle\_t x\_INTERRUPT;

xQueueHandle queue;

void exti\_init()

{

SYSCFG->EXTICR[3] |= SYSCFG\_EXTICR4\_EXTI13\_PC;

EXTI->RTSR = EXTI\_RTSR\_TR13;

//EXTI->PR = EXTI\_PR\_PR13;

EXTI->IMR = EXTI\_IMR\_MR13;

NVIC\_EnableIRQ(EXTI15\_10\_IRQn);

NVIC\_SetPriority(EXTI15\_10\_IRQn,5);

\_\_enable\_irq();

}

void EXTI15\_10\_IRQHandler(void)

{

BaseType\_t needCS = pdFALSE;

EXTI->PR = EXTI\_PR\_PR13;

xSemaphoreGiveFromISR(x\_INTERRUPT, &needCS);

if(needCS == pdTRUE)

{

portYIELD\_FROM\_ISR(needCS);

}

}

void IST(void \*pvParams)

{

int mode = 0;

while(1)

{

if((xSemaphoreTake(x\_INTERRUPT, portMAX\_DELAY)) == pdPASS)

{

mode=!mode;

xQueueSendToFrontFromISR(queue,&mode,0);

}

}

}

void vTask1(void \*pvParams)

{

int mode = 0;

while(1)

{

xQueueReceive(queue,&mode,0);

if(mode)

{

GPIOA->ODR |= GPIO\_ODR\_ODR\_5;//led is

vTaskDelay(250);

GPIOA->ODR &= ~(GPIO\_ODR\_ODR\_5);//led is

vTaskDelay(250);

}

else

{

GPIOA->ODR |= GPIO\_ODR\_ODR\_5;//led is

vTaskDelay(50);

GPIOA->ODR &= ~(GPIO\_ODR\_ODR\_5);//led is

vTaskDelay(50);

}

}

}

int main()

{

RCC->AHB1ENR |=RCC\_AHB1ENR\_GPIOAEN; //gpio is on

RCC->AHB1ENR |=RCC\_AHB1ENR\_GPIOCEN;

GPIOA->MODER |= GPIO\_MODER\_MODER5\_0; //5 OUTPUT

RCC->APB2ENR |= RCC\_APB2ENR\_SYSCFGEN;

exti\_init();

queue = xQueueCreate(16,sizeof(int));

x\_INTERRUPT = xSemaphoreCreateBinary();

if(x\_INTERRUPT != NULL)

{

xTaskCreate(IST, "interrupt", configMINIMAL\_STACK\_SIZE, NULL, 4, NULL);

xTaskCreate(vTask1, "flash", configMINIMAL\_STACK\_SIZE, NULL, 2, NULL);

vTaskStartScheduler();

}

while(1)

{

}

}

**Описание кода программы:**

void EXTI15\_10\_IRQHandler(void) – функция обработки прерываний, в ней сбрасывается флаг прерываний и отдается бинарный семафор для дальнейшей обработки функцией IST.

void IST – задача обработки прерываний, она меняет режим работы светодиода.

void vTask1 – задача для реализации режима работы светодиода, она принимает режим работы через очередь.

void exti\_init()- инициализация прерываний.

**Работа программы:**

При загрузке на плату выполняется мигание светодиода с частотой 10Гц, при нажатии на кнопку частота меняется с 10 до 2Гц.

**Вывод:** выполнив лабораторную работу, мы познакомились с использованием прерываний в ОС FreeRTOS.