ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Кафедра вычислительной техники



Лабораторная работа №2

по дисциплине «Периферийные устройства информационных систем»

CooCox CoIDE

Группа: АВТ-819 Преподаватель: Токарев В. Г.

Студент: Ванин К.Е.

Новосибирск

2021

**Цель работы:**

Целью данной работы является ознакомление студентов с механизмом

аппаратных прерываний микроконтроллера STM32F103C8, с принципом создания функций-обработчиков аппаратных прерываний.

**Задание:**

- изучить теоретический материал по системе прерываний микроконтроллера

STM32F103C8;

- доработать проект из Лабораторной работы 1, заменив программную задержку

аппаратной с использованием прерываний от системного таймера SysTick;

- доработать проект из Лабораторной работы 1, заменив программное чтение

состояний кнопок PA0 и PA15 процедурами обработки внешних прерываний по линиям

EXTI0\_IRQn и EXTI15\_10\_IRQn;

- отладить встраиваемое приложение, используя точки останова для верификации

корректной обработки прерываний.

- оформить отчет с использованием скриншотов экрана работающего в режиме

отладки приложения.

**Ход работы:**

Изучили теоретический материал по системе прерываний.

Доработали проект из лабораторной работы 1, заменив программную задержку аппаратной с использованием прерываний от системного таймера SysTick. Заменили программное чтение состояний кнопок PA0 и PA15 процедурами обработки внешних прерываний по линиям EXTI0\_IRQn и EXTI15\_10\_IRQn.

Продемонстрируем обработку кнопки PA0. При нажатии на кнопку происходит обработка в блоке EXTI0\_IRQHandler:

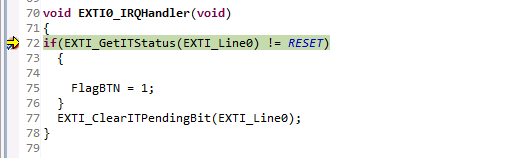


Рисунок 1 – обработка кнопки PA0

Продемонстрируем обработку кнопки PA15. При нажатии на кнопку происходит обработка в блоке EXTI15\_10\_IRQHandler:

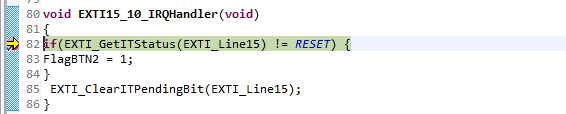


Рисунок 2 – обработка кнопки PA15

**Листинг программы:**

**#include** "stm32f10x.h"

**#include** "stm32f10x\_gpio.h"

**#include** "stm32f10x\_rcc.h"

**#include** "stm32f10x.h"

**#include** "stm32f10x\_exti.h"

**#include** "misc.h"

**static** \_\_IO uint32\_t TimingDelay;

**int** isActive = 0;

**void** **SysTick\_Handler**(**void**)

{

**if** (TimingDelay != 0x00) TimingDelay--;

}

**void** **Delay**(\_\_IO uint32\_t nCount)

{

TimingDelay = nCount;

**while**(TimingDelay != 0);

}

**void** **EXTI0\_IRQHandler**(**void**)

{

**if**(EXTI\_GetITStatus(EXTI\_Line0) != *RESET*)

{

isActive = 1;

}

EXTI\_ClearITPendingBit(EXTI\_Line0);

}

**void** **EXTI15\_10\_IRQHandler**(**void**)

{

**if**(EXTI\_GetITStatus(EXTI\_Line15) != *RESET*)

{

isActive = 0;

}

EXTI\_ClearITPendingBit(EXTI\_Line15);

}

**void** **GPIO\_Blink**(**void**)

{

**int** i;

GPIO\_InitTypeDef GPIO\_InitStructure;

EXTI\_InitTypeDef EXTI\_InitStructure;

NVIC\_InitTypeDef NVIC\_InitStructure;

RCC\_APB2PeriphClockCmd(RCC\_APB2Periph\_GPIOC|RCC\_APB2Periph\_GPIOB| RCC\_APB2Periph\_GPIOA, *ENABLE*);

GPIO\_InitStructure.GPIO\_Pin = GPIO\_Pin\_13;

GPIO\_InitStructure.GPIO\_Mode = *GPIO\_Mode\_Out\_PP*;

GPIO\_Init(GPIOC, &GPIO\_InitStructure);

GPIO\_InitStructure.GPIO\_Pin = GPIO\_Pin\_6 |GPIO\_Pin\_7|GPIO\_Pin\_8|GPIO\_Pin\_9;

GPIO\_InitStructure.GPIO\_Mode = *GPIO\_Mode\_Out\_PP*;

GPIO\_InitStructure.GPIO\_Speed = *GPIO\_Speed\_50MHz*;

GPIO\_Init(GPIOB, &GPIO\_InitStructure);

GPIO\_InitStructure.GPIO\_Pin = GPIO\_Pin\_15|GPIO\_Pin\_0;

GPIO\_InitStructure.GPIO\_Mode = *GPIO\_Mode\_IN\_FLOATING*;

GPIO\_Init(GPIOA, &GPIO\_InitStructure);

SysTick\_Config(SystemCoreClock / 1000);

GPIO\_EXTILineConfig(GPIO\_PortSourceGPIOA, GPIO\_PinSource0);

EXTI\_InitStructure.EXTI\_Line = EXTI\_Line0;

EXTI\_InitStructure.EXTI\_Mode = *EXTI\_Mode\_Interrupt*;

EXTI\_InitStructure.EXTI\_Trigger = *EXTI\_Trigger\_Rising*;

EXTI\_InitStructure.EXTI\_LineCmd = *ENABLE*;

EXTI\_Init(&EXTI\_InitStructure);

NVIC\_InitStructure.NVIC\_IRQChannel = *EXTI0\_IRQn*;

NVIC\_InitStructure.NVIC\_IRQChannelPreemptionPriority = 0x0F;

NVIC\_InitStructure.NVIC\_IRQChannelSubPriority = 0x0F;

NVIC\_InitStructure.NVIC\_IRQChannelCmd = *ENABLE*;

NVIC\_Init(&NVIC\_InitStructure);

GPIO\_EXTILineConfig(GPIO\_PortSourceGPIOA, GPIO\_PinSource15);

EXTI\_InitStructure.EXTI\_Line = EXTI\_Line15;

EXTI\_InitStructure.EXTI\_Mode = *EXTI\_Mode\_Interrupt*;

EXTI\_InitStructure.EXTI\_Trigger = *EXTI\_Trigger\_Rising*;

EXTI\_InitStructure.EXTI\_LineCmd = *ENABLE*;

EXTI\_Init(&EXTI\_InitStructure);

NVIC\_InitStructure.NVIC\_IRQChannel = EXTI15\_10\_IRQn;

NVIC\_InitStructure.NVIC\_IRQChannelPreemptionPriority = 0x0F;

NVIC\_InitStructure.NVIC\_IRQChannelSubPriority = 0x0F;

NVIC\_InitStructure.NVIC\_IRQChannelCmd = *ENABLE*;

NVIC\_Init(&NVIC\_InitStructure);

**while** (1)

{

== 1){

GPIOB->ODR ^= GPIO\_Pin\_6;

Delay(100);

GPIOB->ODR ^= GPIO\_Pin\_6;

GPIOB->ODR ^= GPIO\_Pin\_7;

Delay(100);

GPIOB->ODR ^= GPIO\_Pin\_7;

GPIOB->ODR ^= GPIO\_Pin\_8;

Delay(100);

GPIOB->ODR ^= GPIO\_Pin\_8;

GPIOB->ODR ^= GPIO\_Pin\_9;

Delay(100);

GPIOB->ODR ^= GPIO\_Pin\_9;

}

**else** **if**(isActive == 0){

GPIOB->ODR ^= GPIO\_Pin\_9;

Delay(100);

GPIOB->ODR ^= GPIO\_Pin\_9;

GPIOB->ODR ^= GPIO\_Pin\_8;

Delay(100);

GPIOB->ODR ^= GPIO\_Pin\_8;

GPIOB->ODR ^= GPIO\_Pin\_7;

Delay(100);

GPIOB->ODR ^= GPIO\_Pin\_7;

GPIOB->ODR ^= GPIO\_Pin\_6;

Delay(100);

GPIOB->ODR ^= GPIO\_Pin\_6;

}

Delay(100);

}

}

**Вывод:**

Мы изучили теоретический материал по системе прерываний микроконтроллера STM32F103C8.

Доработали проект из Лабораторной работы 1 с помощью замены программной задержки аппаратной с использованием прерываний от системного таймера SysTick, а также замены программного чтения

состояний кнопок PA0 и PA15 процедурами обработки внешних прерываний по линиям EXTI0\_IRQn и EXTI15\_10\_IRQn.

Провели отладку встраиваемого приложения с помощью точек останов для верификации корректной обработки прерываний.