

681.51(07)

М - 545

№5109

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИИ  
Федеральное государственное автономное образовательное  
учреждение высшего профессионального образования  
«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»



Кафедра синергетики и процессов управления

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ**  
к выполнению лабораторной работы №2  
по дисциплине

**МИКРОПРОЦЕССОРНЫЕ  
ИНФОРМАЦИОННО-УПРАВЛЯЮЩИЕ  
СИСТЕМЫ**

Для студентов направления подготовки  
27.03.03 Системный анализ и управление

Таганрог 2016

УДК 081.51.01(07.07)

Составитель: Скляр А.А.

Методические указания к выполнению лабораторных работ №2 по дисциплине «Микропроцессорные информационно-управляющие системы». – Таганрог: Изд-во ЮФУ, 2016. -14 с.

Представлен лабораторный практикум по ряду базовых разделов дисциплины «Микропроцессорные информационно-управляющие системы». К каждой работе прилагаются краткие теоретические сведения и рекомендации по использованию среды разработки IAR Embedded Workbench.

Указания предназначены для студентов направления подготовки 27.03.03 Системный анализ и управление.

## Лабораторная работа № 2

### Использование портов ввода/вывода.

**Целью** данной работы является изучение основ работы с портами ввода/вывода ARM-контроллера и организации их взаимодействия.

#### Теоретический материал

Порты ввода-вывода являются основными средствами связи микроконтроллера с внешними устройствами. С их помощью можно опрашивать состояние кнопок, контактов реле, уровней сигнала и т.п., а также формировать сигналы управления различными устройствами, например светодиодами, реле и т.д. Порты ввода-вывода имеются практически в любом микроконтроллере независимо от его архитектуры.

Все порты ввода-вывода STM32 являются 16-разрядными и называются GPIO, от General Purpose Input/Output (входы/выходы общего назначения). Порты обозначаются буквами латинского алфавита A, B, C и т.д., например Port A, Port B или Port G. Любой из единичных каналов этих портов может быть сконфигурирован как вход или как выход.

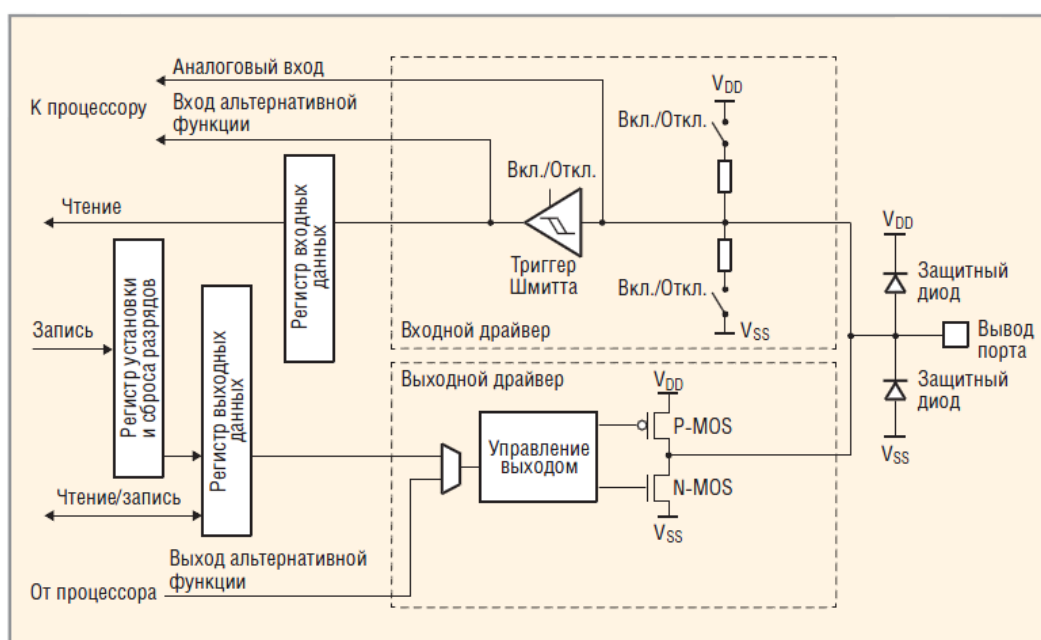


Рис. 1 Схема порта ввода/вывода

Если канал порта настроен на ввод информации, он может функционировать как цифровой или аналоговый вход (см. рис. 1). В режиме цифрового входа канал порта подключается с помощью внутреннего подтягивающего резистора к плюсу источника питания или к нулевому потенциалу.

Настройка порта на вывод (см. рис. 1) требует задания частоты тактирования, определяющей его максимальное быстродействие и тип выхода.

Ниже перечислены возможные типы выхода:

- двухтактный выход;
- выход с открытым стоком;
- двухтактный выход с альтернативной функцией;
- выход с открытым стоком и альтернативной функцией.

В режиме *двухтактный выход* с помощью транзисторов P-MOS и N-MOS выход порта может принимать значения высокого потенциала в качестве логической единицы или нулевого потенциала в качестве логического нуля.

Если порт сконфигурирован как *выход с открытым стоком*, то для управления состоянием линии используется только нижний транзистор N-MOS, а верхний транзистор P-MOS находится в отключенном состоянии.

Последние два режима (*двухтактный выход с альтернативной функцией* и *выход с открытым стоком и альтернативной функцией*) применяются для выводов, которые используют какой-либо функциональный блок периферийного устройства, например SPI, I<sup>2</sup>C, USART, DAC и т.п.

### Конфигурирование портов ввода-вывода в STM32F4-DISCO

Для конфигурирования портов и работы с ними микроконтроллер STM32F429ZI имеет девять регистров для каждого порта. Эти регистры имеют следующие названия и назначение:

- **MODER** – регистр, задающий режим работы порта.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MODER15[1:0]		MODER14[1:0]		MODER13[1:0]		MODER12[1:0]		MODER11[1:0]		MODER10[1:0]		MODER9[1:0]		MODER8[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODER7[1:0]		MODER6[1:0]		MODER5[1:0]		MODER4[1:0]		MODER3[1:0]		MODER2[1:0]		MODER1[1:0]		MODER0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Данные биты изменяются программой для настройки режима порта ввода-вывода:

- 00: режим ввода (режим по умолчанию),
- 01: режим вывода,
- 10: режим альтернативных функций,
- 11: аналоговый режим.
- **OTYPER** – регистр для задания типа выхода.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OT15	OT14	OT13	OT12	OT11	OT10	OT9	OT8	OT7	OT6	OT5	OT4	OT3	OT2	OT1	OT0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Биты 31:16 зарезервированы и не должны изменяться.

Биты 15:0 изменяются программой для задания тип выхода порта ввода-вывода:

- 0: Двухтактный выход (режим по умолчанию),
- 1: Выход с открытым стоком.

- **OSPEEDR** – регистр задания скорости работы порта.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OSPEEDR15 [1:0]		OSPEEDR14 [1:0]		OSPEEDR13 [1:0]		OSPEEDR12 [1:0]		OSPEEDR11 [1:0]		OSPEEDR10 [1:0]		OSPEEDR9 [1:0]		OSPEEDR8 [1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OSPEEDR7[1:0]		OSPEEDR6[1:0]		OSPEEDR5[1:0]		OSPEEDR4[1:0]		OSPEEDR3[1:0]		OSPEEDR2[1:0]		OSPEEDR1 [1:0]		OSPEEDR0 [1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Данные биты записываются программой для настройки скорости работы портов ввода-вывода.

- 00: низкая скорость,
- 01: средняя скорость,
- 10: высокая скорость,
- 11: высшая скорость.

- **PUPDR** – регистр задания режимов pull-up/pull-down;

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PUPDR15[1:0]		PUPDR14[1:0]		PUPDR13[1:0]		PUPDR12[1:0]		PUPDR11[1:0]		PUPDR10[1:0]		PUPDR9[1:0]		PUPDR8[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PUPDR7[1:0]		PUPDR6[1:0]		PUPDR5[1:0]		PUPDR4[1:0]		PUPDR3[1:0]		PUPDR2[1:0]		PUPDR1[1:0]		PUPDR0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Данные биты записываются программой для задания режимов pull-up или pull-down.

- 00: Отключение pull-up и pull-down,
- 01: Режим pull-up,
- 10: Режим pull-down,
- 11: Зарезервировано.

- **IDR** – регистр для чтения физического состояния выводов порта;

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDR15	IDR14	IDR13	IDR12	IDR11	IDR10	IDR9	IDR8	IDR7	IDR6	IDR5	IDR4	IDR3	IDR2	IDR1	IDR0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

Биты 31:16 зарезервированы и не должны изменяться.

Биты 15:0 являются только для чтения и доступ к ним может быть получен только путем считывания целого машинного слова. Данные биты содержат значение входа соответствующего порта.

- **ODR** – регистр для записи данных непосредственно в порт

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ODR15	ODR14	ODR13	ODR12	ODR11	ODR10	ODR9	ODR8	ODR7	ODR6	ODR5	ODR4	ODR3	ODR2	ODR1	ODR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Биты 31:16 зарезервированы и не должны изменяться.

Биты 15:0 предназначены для передачи сигнала в порт. Данные биты записываются и читаются программой.

- **BSRR** – регистр установки и сброса разрядов порта;

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BS15	BS14	BS13	BS12	BS11	BS10	BS9	BS8	BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

Биты 31:16 BR<sub>y</sub>: В порту *x* сбрасывается *y* бит ( $y = 0..15$ ). Данные биты доступны только для записи:

0: Не совершать ни каких действий с соответствующим битом в регистре ODR<sub>x</sub>.

1: Сбросить соответствующий бит в регистре ODR<sub>x</sub>.

Биты 15:0 BS<sub>y</sub>: В порту *x* устанавливается *y* бит ( $y = 0..15$ ). Данные биты доступны только для записи:

0: Не совершать ни каких действий с соответствующим битом в регистре ODR<sub>x</sub>.

1: Установить соответствующий бит в регистре ODR<sub>x</sub>.

*Примечание: если установлены одновременно и BS<sub>y</sub> и BR<sub>y</sub>, то значения BS<sub>y</sub> имеют более высокий приоритет.*

- **LCKR** – регистр для блокировки установленной конфигурации порта.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															LCKK
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCK15	LCK14	LCK13	LCK12	LCK11	LCK10	LCK9	LCK8	LCK7	LCK6	LCK5	LCK4	LCK3	LCK2	LCK1	LCK0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Биты 31:17 зарезервированы и не должны изменяться.

Бит 16 – Блокировочный ключ, который может быть всегда прочитан. Значение данного ключа изменятся с использованием специальной последовательности:

WR LCKR[16] = '1' + LCKR[15:0]

WR LCKR[16] = '0' + LCKR[15:0]

WR LCKR[16] = '1' + LCKR[15:0]

RD LCKR

RD LCKR[16] = '1' (данная операция чтения необходима для

подтверждения блокировки порта)

*Примечание: во время выполнения данной последовательности значения битов LCK[15:0] не должны изменяться, иначе блокировка порта не состоится.*

Биты 15:0 – биты блокировки соответствующего порта. Могут быть изменены при условии, что бит LCKR равен 0.

0: Конфигурация порта не заблокирована.

1: Конфигурация порта заблокирована.

- **AFR** – регистр задания альтернативной функции портам ввода/вывода. В данной лабораторной работе не используется.

### Подготовка проекта в EWARM

Для создания программы управления портами ввода-вывода микроконтроллера STM32F429ZI необходимо настроить проект в среде разработки IAR EWARM. Для этого необходимо выполнить следующую последовательность шагов:

1. Создание проекта на языке C:

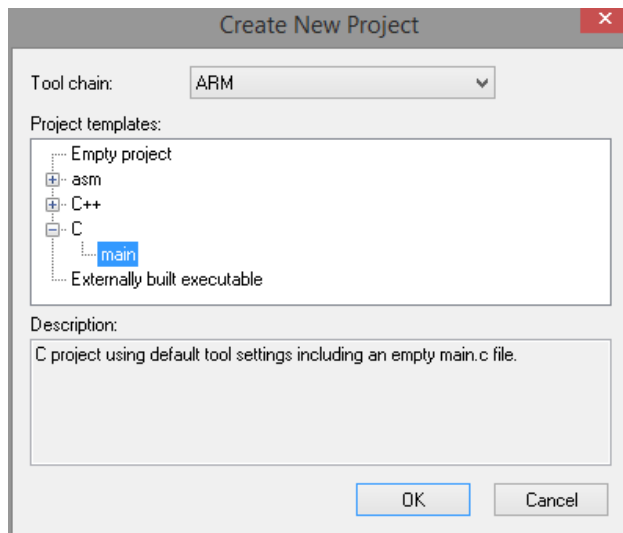


Рис. 2 Окно создания проекта

2. Настроить проект – выбрать ядра микроконтроллера Cortex-M4F

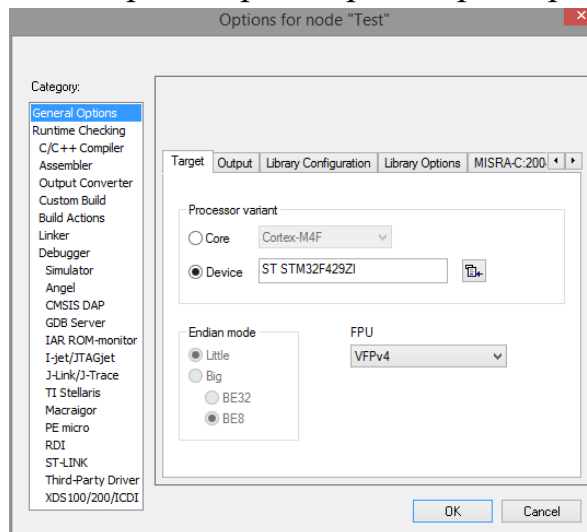


Рис. 3 Окно настройки ядра процессора

3. Настроить язык C/C++ – отключить оптимизацию кода

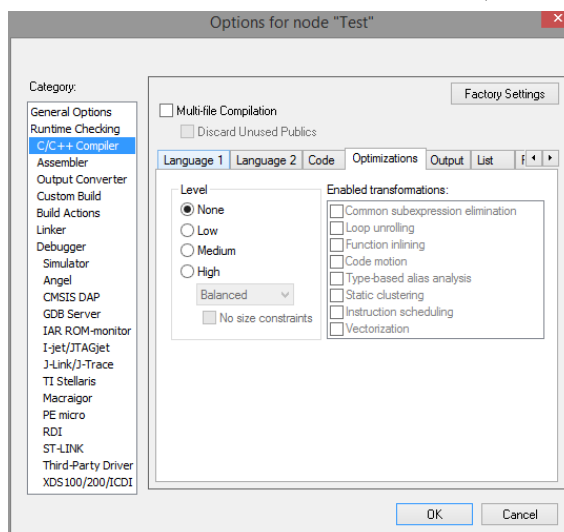


Рис. 4 Окно настройки компилятора



4. Настроить язык C/C++ – добавить в подключаемые директории значение «\$PROJ\_DIR\$», а также в определения значение «STM32F429\_439xx»:

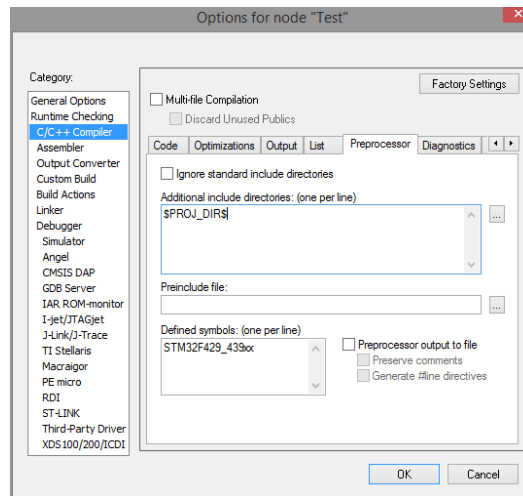


Рис. 5 Окно настройки компилятора

5. Настроить отладчик – выбрать «ST-LINK»

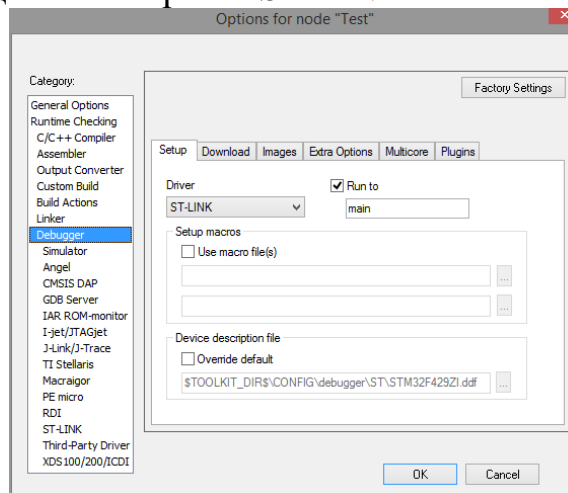


Рис. 6 Окно настройки отладчика

6. Настроить отладчик – во вкладке Download – выбрать значение «Verify load» и «Use flash load»

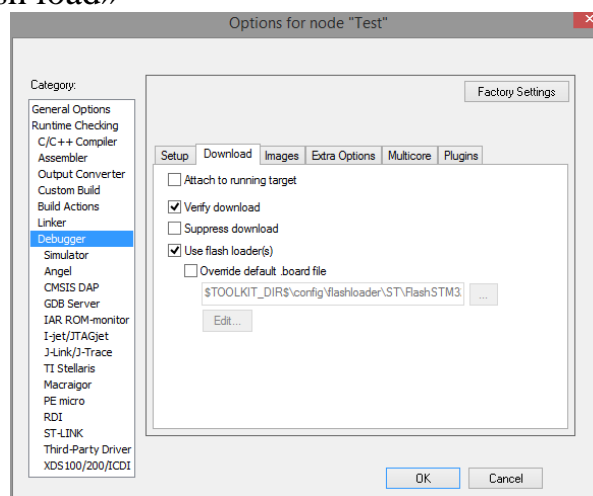


Рис. 7 Окно настройки отладчика

7. Настроить ST-LINK – в поле «Reset» выбрать значение «connect during reset»; в поле «Interface» выбрать значение «SWD».

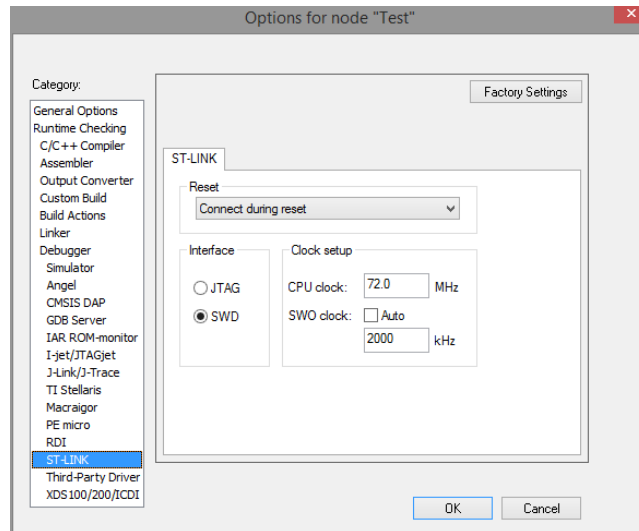


Рис. 8 Окно настройки ST-LINK

8. Настроить CMSIS – выбрать Use CMSIS (подключение стандартных библиотек Cortex-M4F)

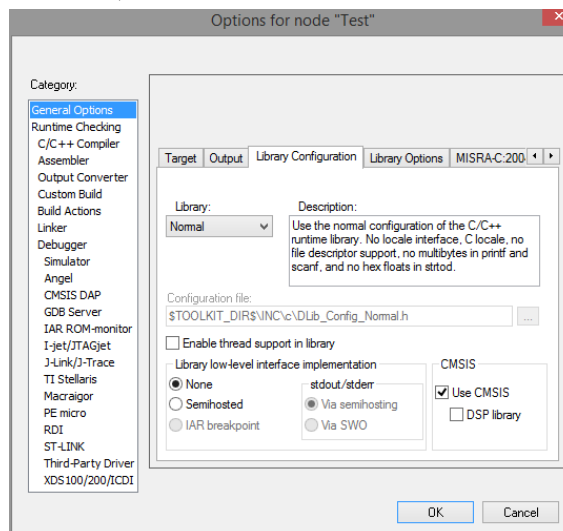


Рис. 9 Окно главных настроек проекта

## Инициализация процессора

После того, как среда разработки программы управления портами ввода-вывода настроена, необходимо перейти к формированию файлов инициализации процессора STM32F429ZI. Для этого необходимо выполнить следующую последовательность шагов:

1. Запустить генератор кода STM32F4xx\_Clock\_Configuration\_V1.1.0

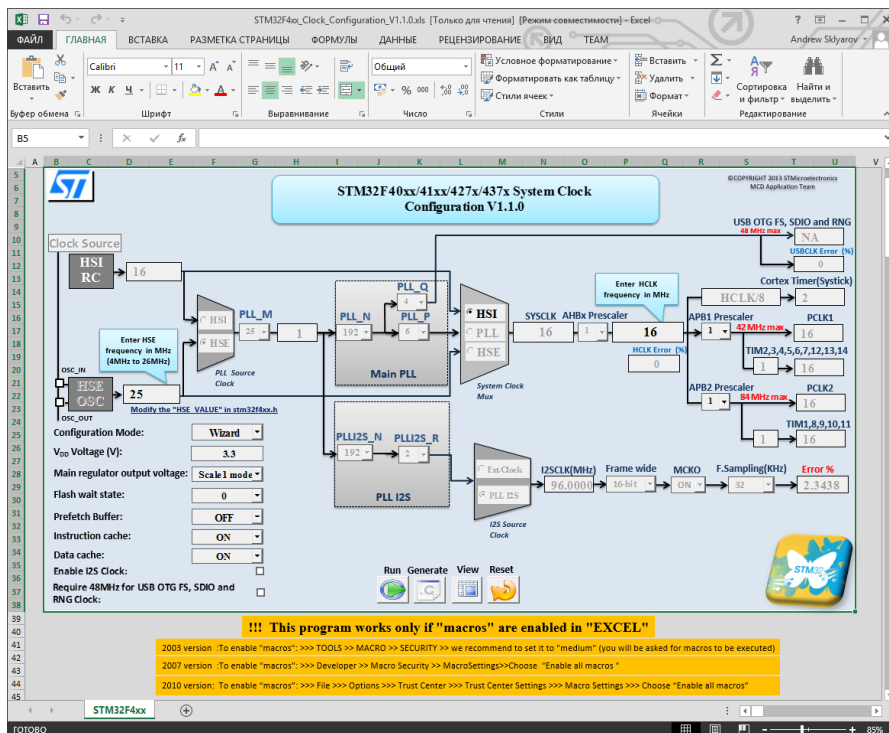


Рис. 10 Генератор кода для инициализации частоты контроллера

2. Сгенерировать файл `system_stm32f4xx.c` для частоты HCLK = 168 МГц.
3. Подключить файл `system_stm32f4xx.c` к проекту.

### Пример программы управления портами ввода-вывода

Рассмотрим пример программы, которая демонстрирует работу портов ввода/вывода. Она переключает состояние светодиода LD4 (см рис. 11) после прохождения некоторого количества процессорного времени.

```
#include "stm32f4xx.h"

#define GPIO_OTYPER_PUSH_PULL 0x00000000UL
#define GPIO_OTYPER_OPENDRAIN 0x00000001UL

#define GPIO_MODER_INPUT 0x00000000UL
#define GPIO_MODER_OUTPUT 0x00000001UL
#define GPIO_MODER_ALTER 0x00000002UL
#define GPIO_MODER_ANALOG 0x00000003UL

#define GPIO_OSPEEDER_LOW 0x00000000UL
#define GPIO_OSPEEDER_MEDIUM 0x00000001UL
#define GPIO_OSPEEDER_FAST 0x00000002UL
#define GPIO_OSPEEDER_HIGH 0x00000003UL

#define GPIO_PUPDR_NOPUPD 0x00000000UL
#define GPIO_PUPDR_PULLUP 0x00000001UL
#define GPIO_PUPDR_PULLDOWN 0x00000002UL

#define GPIO_AFR_AF0 0x00000000UL
#define GPIO_AFR_AF1 0x00000001UL
#define GPIO_AFR_AF2 0x00000002UL
#define GPIO_AFR_AF3 0x00000003UL
#define GPIO_AFR_AF4 0x00000004UL
#define GPIO_AFR_AF7 0x00000007UL
#define GPIO_AFR_AF9 0x00000009UL
```

```

void SystemInit(void);

void main(void)
{
    SystemInit();

    //*****инициализация порта светодиода LD4*****
    // подключение тактового генератора к общей шине с портам
    RCC->AHB1ENR |= RCC_AHB1ENR_GPIOGEN;

    //номер порта PG14 = LD4
    unsigned int num_port = 14;

    // отчищаем биты от прежнего значения AFR для порта PG14
    GPIOG->AFR[num_port >> 3] &= ~(0x0000000FUL << ((num_port & 0x07) * 4));
    // AFR = AF0 или не использовать альтернативные функции
    GPIOG->AFR[num_port >> 3] |= GPIO_AFR_AF0 << ((num_port & 0x07) * 4);

    // отчищаем биты от прежнего значения
    GPIOG->MODER &= ~(0x00000003UL << (num_port * 2));
    // mode = output или MODER14 = [29 бит = 0; 28 бит = 1]
    GPIOG->MODER |= GPIO_MODER_OUTPUT << (num_port * 2);

    // отчищаем биты от прежнего значения
    GPIOG->OSPEEDR &= ~(0x00000003UL << (num_port * 2));
    // OSPEEDER = HIGH или OSPEEDER14 = [29 бит = 1; 28 бит = 1]
    GPIOG->OSPEEDR |= GPIO_OSPEEDER_HIGH << (num_port * 2);

    // отчищаем биты от прежнего значения
    GPIOG->OTYPER &= ~(0x00000001UL << num_port);
    // OTYPER = PUSH_PULL или OT14 = [14 бит = 0]
    GPIOG->OTYPER |= GPIO_OTYPER_PUSH_PULL << num_port;

    // отчищаем биты от прежнего значения
    GPIOG->PUPDR &= ~(0x00000003UL << (num_port * 2));
    // PUPDR = NOPUPD или PUPDR14 = [29 бит = 0; 28 бит = 0]
    GPIOG->PUPDR |= GPIO_PUPDR_NOPUPD << (num_port * 2);
    //*****

    while (1)
    {
        GPIOG->BSRRL = 1 << 14; //LedsTurnON() или BS14 = [14 бит = 1]
        for (int i = 0; i < 1500000; i++); // delay

        GPIOG->BSRRH = 1 << 14; //LedsTurnOFF() или BR14 = [30 бит = 1]
        for (int i = 0; i < 1500000; i++); // delay
    }
}

```

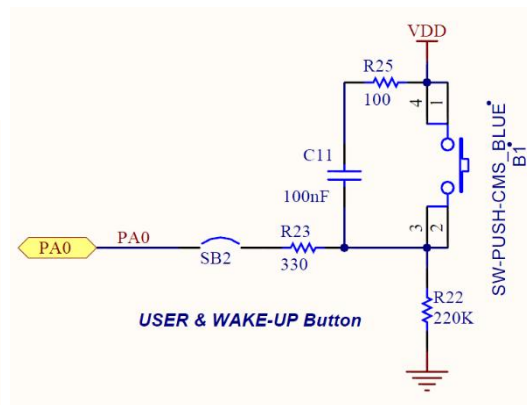
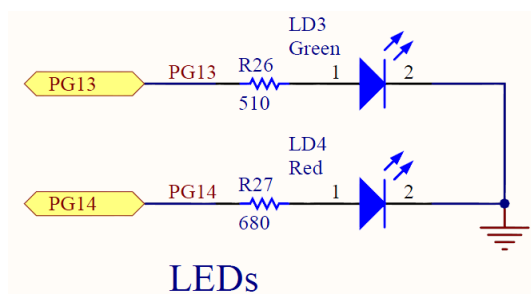


Рис. 11 Схемы подключения периферийных устройств (LD3 – зеленый светодиод, LD4 – красный светодиод, B1 – пользовательская кнопка)

Схема подключения периферийных устройств показана на рис. 11. Из данной схемы следует что светодиод LD3 находится на порту PG13, светодиод LD4 находится на порту PG14, а кнопка B1 находится на порту PA0.

Для работы с кнопкой B1 необходимо настроить порт PA0 (обращение к регистрам порта необходимо осуществлять через GPIOA) на режим ввода (MODER0 = [1 бит = 0; 0 бит = 0]) с максимальной частотой тактирования (OSPEEDER0 = [1 бит = 1; 0 бит = 1]) и без использования подтягивающего резистора (PUPDR0 = [1 бит = 0; 0 бит = 0]). Также для запуска работы порта PA0 необходимо подключить тактовый генератор к общей шине с портом A и настроить значение регистра AFR таким образом, чтобы альтернативные функции не использовались:

```
// подключение тактового генератора к общей шине с портами
RCC->AHB1ENR |= RCC_AHB1ENR_GPIOAEN;
unsigned int num_port = 0; // номер порта PA0 = USER & WAKE-UP Button

// отчищаем биты от прежнего значения
GPIOA->AFR[num_port >> 3] &= ~(0x000000FUL << ((num_port & 0x07) * 4));

// AFR = AF0 или не использовать альтернативные функции
GPIOA->AFR[num_port >> 3] |= GPIO_AFR_AF0 << ((num_port & 0x07) * 4);
```

### Порядок выполнения работы

1. На основе кода примера приложения создать новый проект в среде программирования IAR Embedded Workbench IDE.
2. Написать программу управления портами ввода-вывода в соответствии с вариантом задания.
3. Скомпилировать и отладить написанную программу в среде IAR Embedded Workbench IDE.
4. Провести тестирование работоспособности программы на отладочной плате STM32F4- DISCO.

### Содержание отчета

1. Цель работы.
2. Описание последовательности действий для создания проекта.
3. Исходный текст программы с комментариями.
4. Выводы по работе.

## Варианты заданий

№ варианта	Описание
1	Реализовать схему переменного включения/выключения светодиодов LD3 и LD4, расположенных на плате.
2	Реализовать схему включения/выключения светодиода LD3 по кнопке В1, расположенной на плате.
3	Реализовать схему включения/выключения светодиода LD4 по кнопке В1, расположенной на плате.
4	Реализовать схему одновременного включения/выключения светодиодов LD3 и LD4 по кнопке В1, расположенной на плате.
5	Реализовать схему переменного включения/выключения светодиодов LD3 и LD4 по кнопке В1, расположенной на плате.
6	Реализовать схему одновременного включения/выключения светодиодов LD3 и LD4, расположенных на плате.
7 <sup>1</sup>	Реализовать схему включения/выключения светодиода LD3, расположенного на плате.

---

<sup>1</sup> Задание с минимальным уровнем сложности