

КАФЕДРА №

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

должность, уч. степень, звание

подпись, дата

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ № 2

Общие принципы программного управления микроконтроллером
серии STM32F303xC

по курсу: Программирование встроенных приложений

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ гр. №

подпись, дата

инициалы, фамилия

Санкт-Петербург 2024

Цель работы: Привитие практических навыков по работе с ИСР Keil, ПБВ микроконтроллера и технической документацией.

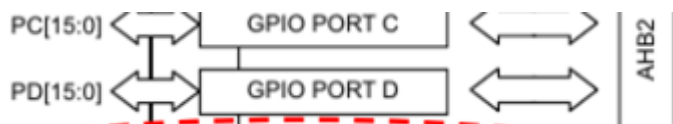
Задание на лабораторную работу:

- 1) Изучить принципы настройки линии порта в режим вывода.
- 2) Создать проект с подключением библиотеки CMSIS. В файлах проекта найти определение стека и области памяти с неупорядоченным хранением данных ('heap'), изменить размеры стека и 'heap' согласно варианту задания (см. табл. 1.1.).
- 3) Написать на языке «си» программу и отладить её работу по переключению уровня сигнала на двух линиях в/в микроконтроллера. Частота получаемых на линиях сигналов типа меандр настраивается согласно варианту (см. табл. 1.1). Значение частоты контролировать осциллографом на выводах отладочной платы. В программе один из выводов настраивать через адреса регистров управления ПБВ, второй с использованием библиотеки CMSIS. Значение счётчика задержки, под заданную вариант частоту переключения линии в/в, подбирать вручную или рассчитать (частота тактирования 8МГц). Сохранить эпюры напряжения для каждой линии в/в.
- 4) Оформить отчёт.

Вариант 24:

Номер варианта	Размер стека, байт	Размер 'heap', байт	Линии в/в	Частота переключений, Гц
24	2560	9472	PD6, PD14	3720; 7440

Перед использованием ПБВ D (GPIOD) необходимо включить его в работу и вывести из состояния сброса. Для этого в подсистеме RCC (RM0316 [1] стр. 123) – сброса и управления тактированием, имеется несколько регистров управления. В спецификации DS9118 [2], на странице 13 рисунок 1, показано, что все ПБВ подключены к шине AHB2:



Для управления тактированием шины AHB2 используется регистр RCC_AHBENR (RM0316 стр.148 подраздел 9.4.6) в котором имеются следующие управляющие биты:

23	22	21	20	19	18	17	16
GPIODEN ⁽¹⁾	GPIODEN	GPIODEN	GPIODEN	GPIODEN	GPIODEN	GPIODEN	GPIODEN ⁽¹⁾
rw	rw	rw	rw	rw	rw	rw	rw

В пояснении указано:

Bit 20 GPIODEN: I/O port D clock enable (бит 20 разрешает тактирование ПБВ D)

Set and cleared by software. (Устанавливается и очищается ПО)

0: I/O port D clock disabled (0: тактирование порта в/в запрещено)

1: I/O port D clock enabled (1: тактирование порта в/в разрешено)

Таким образом, для включения в работу необходимо ПБВ(порт ввода/вывода) D необходимо установить единицу в 20 бит регистра RCC_AHBENR

Выясним адрес этого регистра и изменим состояние 20-го бита на единицу. Сначала выясним раздел памяти, в котором находится этот регистр. По таблице 20 на странице 54 спецификации DS9118, в разделе № 5 Memory mapping (карта памяти), находим область памяти подсистемы RCC:

AHB1	0x4002 1400 - 0x4002 1FFF	3 K	Reserved
	0x4002 1000 - 0x4002 13FF	1 K	RCC

Как видно из таблицы, адрес начала области памяти подсистемы RCC – 0x4002 1000. Далее из описания регистра RCC_AHBENR (RM0316 стр.148 подраздел 9.4.6), находим поле Address offset: 0x14 (Адрес смещения).

9.4.6 AHB peripheral clock enable register (RCC_AHBENR)

Address offset: 0x14

Зная смещение, вычисляем адрес регистра RCC_AHBENR:
 $0x4002\ 1000 + 0x0000\ 0014 = 0x4002\ 1014$.

Закодируем изменение 20-го бита по адресу 0x40021014 на языке си строчкой
 ‘*(uint32_t*)(0x40021014) |= 0x00100000;’ (число, которое было в двоичную, убираю один ноль, получаю: 10000000000000000000 – 20 бит. Переводим в шестнадцатеричную, получаю адрес 20 бита - 0x00100000

Здесь в выражении используется оператор ‘|=’, а не простое присваивание ‘=’, поскольку в регистре уже имеются установленные в единицу биты и их обнуление может привести к системному сбою. uint32_t соответствует типу unsigned int. В общем виде получится программа:

```
#include "RTE_Components.h"
#include CMSIS_device_header
int main()
{
    *(uint32_t*)(0x40021014) |= 0x00100000;
    while(1){ }
}
```

Теперь посмотреть регистры подсистемы RCC, что поменялся 20 бит(GPIODEN) в AHBENR



Таким образом, записав единицу в 20 байт мы включили в работу порт ввода/вывода D.

Теперь установим высокий уровень на PD6. Выделяем из таблицы 72 на странице 230 RM0316 [1], нужный нам режим:

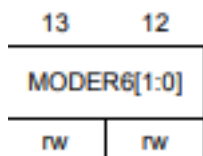
Table 72. Port bit configuration table⁽¹⁾

MODER(i) [1:0]	OTYPER(i)	OSPEEDR(i) [1:0]	PUPDR(i) [1:0]		I/O configuration	
01	0	SPEED	0	0	GP output	PP
	0		0	1	GP output	PP + PU
	0		1	0	GP output	PP + PD
	0		1	1	Reserved	

Названия колонок в таблице соответствует названиям управляющих регистров ПБВ, за исключением последней, что сокращённо описывает сам режим. Выбираем режим работы линии в/в «GP output + PP», что означает General-Purpose output Push-Pull – режим двухтактного выхода.

Для настройки в этот режим PD6 необходимо:

1) В регистр GPIOD_MODER в поле MODER6[1:0] записать 01 (RM0316 стр.237 подраздел 11.4.1), т.е. в 12-ый бит записать единичку, а в 13-ый бит нолик:



2) В регистр GPIOD_OTYPER в 1-й бит записать ноль, но так как после сброса МК он обнуляется, соответственно там и так ноль. Этот шаг можно пропустить.

3) Регистр GPIOD_OSPEEDR настраивает скорость нарастания(спада) фронта сигнала при высоких скоростях переключения вывода (свыше 100КГц), поскольку мы работаем на низких частотах пропускаем и эту настройку.

4) В регистр GPIOD_PUPDR записать нули в биты 12 и 13, но так как после сброса в регистре нули и этот шаг пропускаем. Получается, что для настройки линии PD6 в режим вывода достаточно настроить только один регистр GPIOD_MODER записав в поле MODER6[1:0] значение 0x01, т.е. единицу в 2-й бит. Определим адрес регистра GPIOD_MODER используя ранее рассмотренную таблицу 20 (DS9118 стр. 54):

AHB2	0x4800 0C00 - 0x4800 0FFF	1 K	GPIOD

Адрес начала области памяти ПБВ GPIOD – 0x4800 0C00. Смещение (Address offset) регистра GPIOD_MODER: 0x00. Получаем адрес регистра GPIOD_MODER:

$0x4800\ 0C00 + 0x000\ 0000 = 0x4800\ 0C00$.

И в код программы добавим строчку:

`*(uint32_t*)(0x48000C00) |= 0x00001000;` (1000 – 12 бит, то есть в 0001000000000000 двоичной и 1000 в 16)

Настроив логику работы линии в/в PD6 на вывод, перейдём к непосредственному управлению состоянием сигнала на линии. Используем регистр GPIOD_ODR (RM0316 стр.239 п. 11.4.6) – биты которого управляют состоянием линий, настроенных в режим выхода. Установка 1-го бита этого регистра в единицу – установит на линии PD6 высокий уровень, а сброс в ноль – низкий уровень. Таким образом, управляя состоянием этого бита, мы будем управлять состоянием линии PD6. Так как смещение этого регистра: 0x14, получаем адрес GPIOD_ODR: $0x4800\ 0C00 + 0x14 = 0x4800\ 0C14$.

Исходный код:

```
#include "RTE_Components.h"
#include CMSIS_device_header

void delay(volatile uint32_t count){
    while(count--)
        __NOP();
}

int main()
{
    // Включаем GPIOD
    *((volatile uint32_t*)0x40021014) |= (1 << 20);

    // Устанавливаем PD6 и PD14 в режим вывода
    *((volatile uint32_t*)0x48000C00) |= ((1 << (6 * 2)) | (1 << (14 * 2)));

    while(1){

        // Устанавливаем "1" на PD6 и "0" на PD14
        *((volatile uint32_t*)0x48000C14) ^= (1 << 6);
        *((volatile uint32_t*)0x48000C14) ^= (1 << 14);
        delay(64);

        // Устанавливаем "1" на PD14
        *((volatile uint32_t*)0x48000C14) ^= (1 << 14);
        delay(64);
    }
}
```

}

```
main.c startup_stm32f303xc.s 2.map
1 #include "RTE_Components.h"
2 #include CMSIS_device_header
3
4 void delay(volatile uint32_t count){
5     while(count--){
6         __NOP();
7     }
8
9     int main()
10 {
11     // Включаем GPIOD
12     *((volatile uint32_t*)0x40021014) |= (1 << 20);
13
14     // Устанавливаем PD6 и PD14 в режим вывода
15     *((volatile uint32_t*)0x48000C00) |= ((1 << (6 * 2)) | (1 << (14 * 2)));
16
17 while(1){
18
19     // Устанавливаем "1" на PD6 и "0" на PD14
20     *((volatile uint32_t*)0x48000C14) ^= (1 << 6);
21     *((volatile uint32_t*)0x48000C14) ^= (1 << 14);
22     delay(64);
23
24     // Устанавливаем "1" на PD14
25     *((volatile uint32_t*)0x48000C14) ^= (1 << 14);
26     delay(64);
27 }
28 }
29
```

Таблица трассировки заданных выводов STM32F303xC:

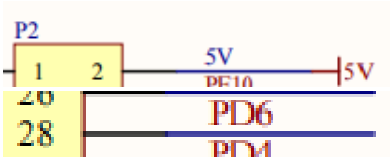
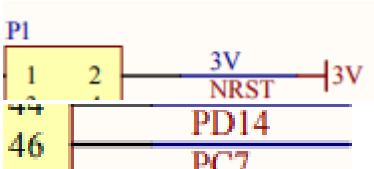
Номер вывода	Обозначение согласно DS9118, стр. 34	Номер разъёмов и выводов на отладочной плате
82	PD6	28 штырь на двухрядном штыревом разъёме P2: 
56	PD14	46 штырь на двурядном штыревом разъёме P1: 

Таблица используемых регистров STM32F303xC с расчётом адресов:

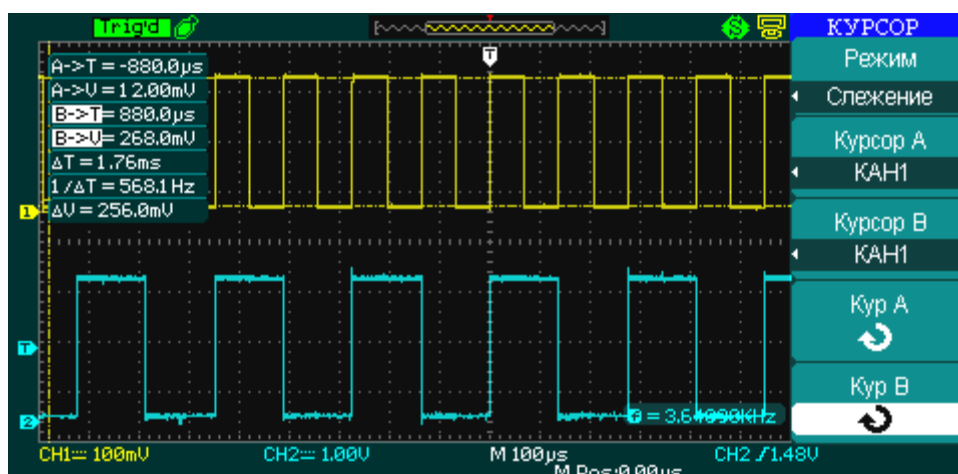
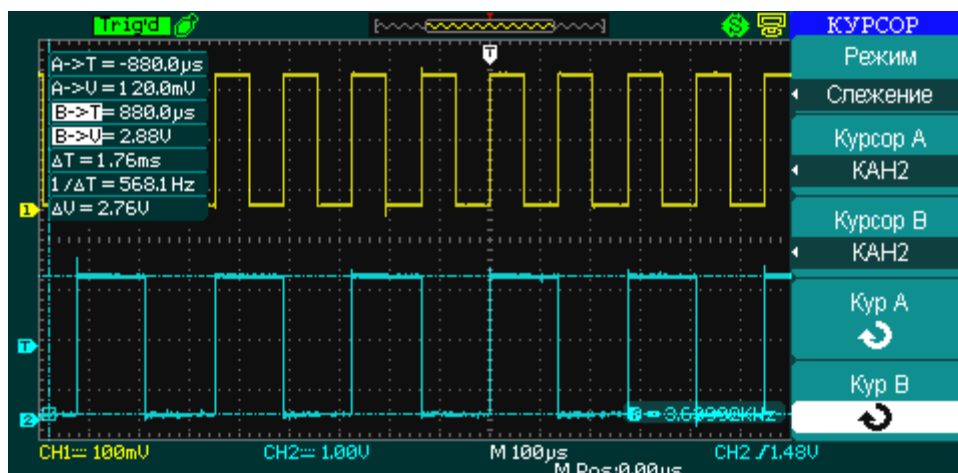
Регистр	Расчёт адреса и ссылки на документацию	Биты и их значение согласно документации			
RCC_AHBENR	$0x4002\ 1000 + 0x0000\ 0014 = 0x4002\ 1014$ DS9118 стр. 54 <table border="1" data-bbox="507 1281 1149 1326"> <tr> <td>0x4002 1000 - 0x4002 13FF</td> <td>1 K</td> <td>RCC</td> </tr> </table> RM0316 стр. 151 Address offset: 0x14	0x4002 1000 - 0x4002 13FF	1 K	RCC	Bit 20 GPIODEN – разрешает работу GPIOD
0x4002 1000 - 0x4002 13FF	1 K	RCC			
GPIOD_MODER	$0x4800\ 0C00 + 0x000\ 0000 = 0x4800\ 0C00$ DS9118, стр. 54 <table border="1" data-bbox="507 1467 1125 1512"> <tr> <td>AHB2 0x4800 0C00 - 0x4800 0FFF</td> <td>1 K</td> <td>GPIOD</td> </tr> </table> RM0316 стр. 240 Address offset: 0x00	AHB2 0x4800 0C00 - 0x4800 0FFF	1 K	GPIOD	Bits 3, 2 MODER[1:0] - управляет режимом работы 1 линии ПВВ
AHB2 0x4800 0C00 - 0x4800 0FFF	1 K	GPIOD			
GPIOD_ODR	$0x4800\ 0C00 + 0x14 = 0x4800\ 0C14$ RM0316 стр. 151 Address offset: 0x14	Управляя состоянием этого бита, мы будем управлять состоянием линии PD1			

Выписка из 25 с указанием размера стека, ‘heap’, затрат оперативной и постоянной памяти проекта:

HEAP	0x20000060	Section	9472	startup_stm32f303xc.o(HEAP)
Heap_Mem	0x20000060	Data	9472	startup_stm32f303xc.o(HEAP)
STACK	0x20002560	Section	2560	startup_stm32f303xc.o(STACK)
Stack_Mem	0x20002560	Data	2560	startup_stm32f303xc.o(STACK)
__initial_sp	0x20002f60	Data	0	startup_stm32f303xc.o(STACK)

609	Total RO	Size (Code + RO Data)	976	(0.95kB)
610	Total RW	Size (RW Data + ZI Data)	12128	(11.84kB)
611	Total ROM	Size (Code + RO Data + RW Data)	976	(0.95kB)

Эпюры напряжений выходов STM32F303xC с указанием амплитуды, частоты, периода каждого сигнала:



Характеристика	Линия PD6
Период	0.27 мс

Частота, Гц	3640
Амплитуда, В	3

Характеристика	Линия PD14
Период	0.14 мс
Частота, Гц	7280
Амплитуда, В	3