

КАФЕДРА №

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

должность, уч. степень, звание

подпись, дата

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №3

Настройка подсистемы тактирования

Микроконтроллера

по курсу: Программирование встроенных приложений

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

подпись, дата

инициалы, фамилия

Санкт-Петербург 2024

1. Задание

24	44 (HSI)	PA15	PB0
----	----------	------	-----

2. Листинг

```
#include "RTE_Components.h" // Component selection
#include CMSIS_device_header // Device header
#define DELAY {__nop();__nop();__nop();__nop();} // Задержка на 4 такта
int main(void)

{
    // Supportive variables (compiler don't modify)
    volatile uint32_t StartUpCounter = 0, HSEStatus = 0;

    // Turn on HSE oscillator
    SET_BIT(RCC->CR, RCC_CR_HSEON);

    // Wait for HSE to turn on
    do {
        HSEStatus = RCC->CR & RCC_CR_HSERDY;
        StartUpCounter++;
    } while ((HSEStatus == 0) && (StartUpCounter != 0x15000));

    // Check if HSE is working
    if ((RCC->CR & RCC_CR_HSERDY) != RESET) {
        // Configure FLASH
        FLASH->ACR = 0;

        // AHB Pre = 4
        RCC->CFGR |= (uint32_t) RCC_CFGR_HPRE_DIV4;

        // Configure PLL for 44 MHz
        CLEAR_BIT(RCC->CR, RCC_CR_PLLON);
        RCC->CFGR |= (uint32_t)(RCC_CFGR_PLLSRC_HSE_PREDIV |
RCC_CFGR_PLLMUL16);
        RCC->CFGR2 |= (uint32_t)RCC_CFGR2_PREDIV_DIV3;
        RCC->CFGR |= (uint32_t)RCC_CFGR_PLLXTPRE_HSE_PREDIV_DIV2;
```

```

// Turn on PLL and wait for it to stabilize
SET_BIT(RCC->CR, RCC_CR_PLLON);
while ((RCC->CR & RCC_CR_PLLRDY) == 0) {}

// Set PLL as clock source for the MC
RCC->CFGR |= (uint32_t)RCC_CFGR_SW_PLL;
while ((RCC->CFGR & (uint32_t)RCC_CFGR_SWS) != (uint32_t)RCC_CFGR_SWS_PLL)
{}
} else {
    while (1) {} // HSE doesn't launch
}

// Update system core clock
SystemCoreClockUpdate();

// Configure MCO to HSI
SET_BIT(RCC->CFGR, RCC_CFGR_MCO_HSI);

// Enable clocking for GPIOA and GPIOB
SET_BIT(RCC->AHBENR, RCC_AHBENR_GPIOAEN);
SET_BIT(GPIOA -> MODER, GPIO_MODER_MODER8_1);
CLEAR_BIT(GPIOA -> AFR[1], GPIO_AFRH_AFRH0_Msk);

SET_BIT(RCC->AHBENR, RCC_AHBENR_GPIOBEN);

// Set PA15 and PB0 to output mode
GPIOA->MODER&=~GPIO_MODER_MODER15;
SET_BIT(GPIOA->MODER, GPIO_MODER_MODER15_0);
SET_BIT(GPIOB->MODER, GPIO_MODER_MODER0_0);

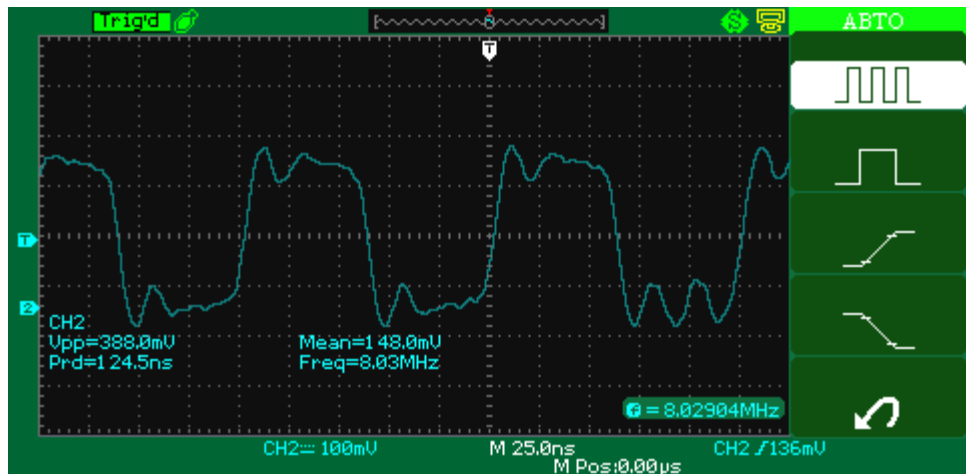
// Configure PA15 and PB0 output types and pull-up/pull-down settings
SET_BIT(GPIOA->OTYPER, GPIO_OTYPER_OT_15);

SET_BIT(GPIOB->PUPDR, GPIO_PUPDR_PUPDR0_1);
SET_BIT(GPIOA->PUPDR, GPIO_PUPDR_PUPDR15_0);

```

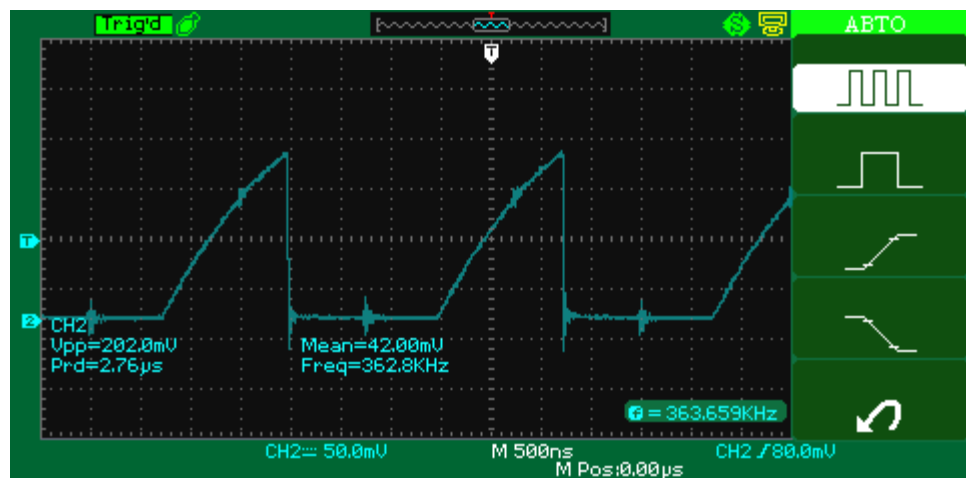
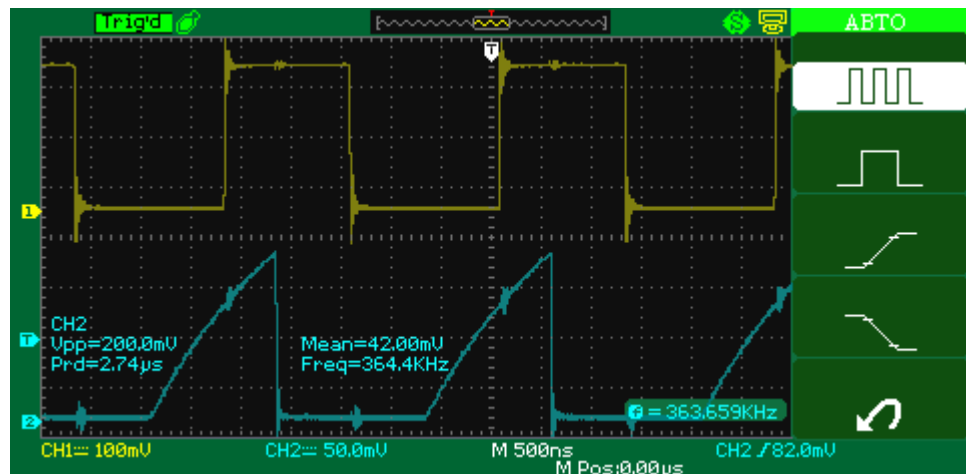
```
while (1) {  
    // Set PA15 and PB0  
    GPIOA->BSRR = GPIO_ODR_15;  
    GPIOB->BSRR = GPIO_ODR_0;  
  
    // Delay  
    //DELAY; DELAY;  
  
    // Reset PA15 and PB0  
    GPIOA->BRR = GPIO_ODR_15;  
    GPIOB->BRR = GPIO_ODR_0;  
  
    // Delay  
    //DELAY; DELAY;  
}  
}
```

3. Осциллограмму вывода MCO (PA8)

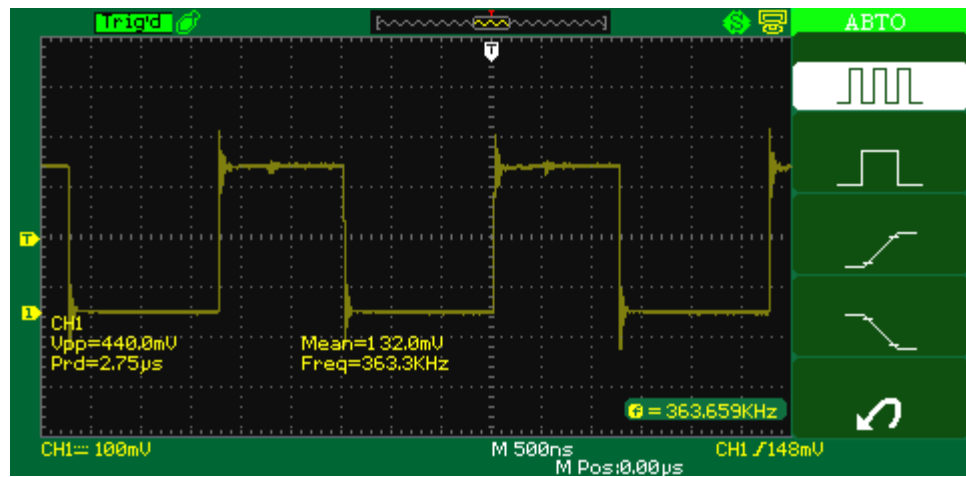


4. Осциллограммы

а) Без оптимизации

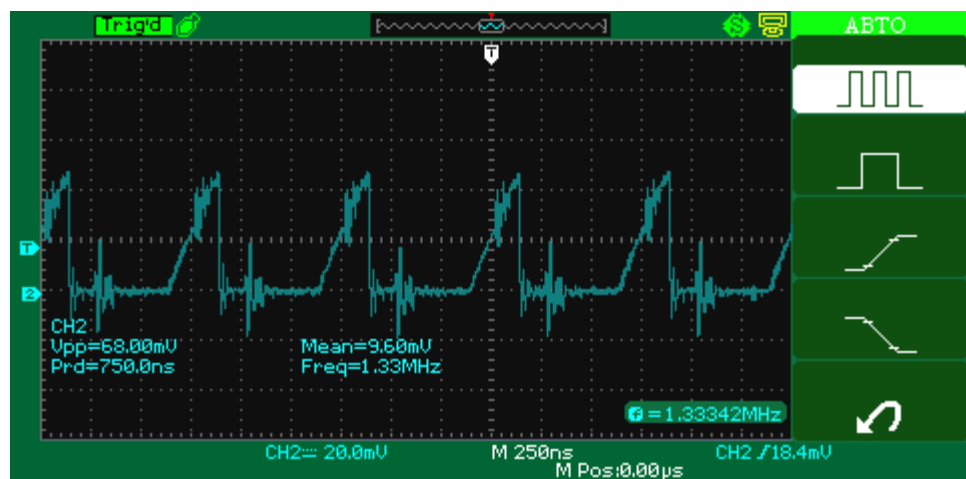
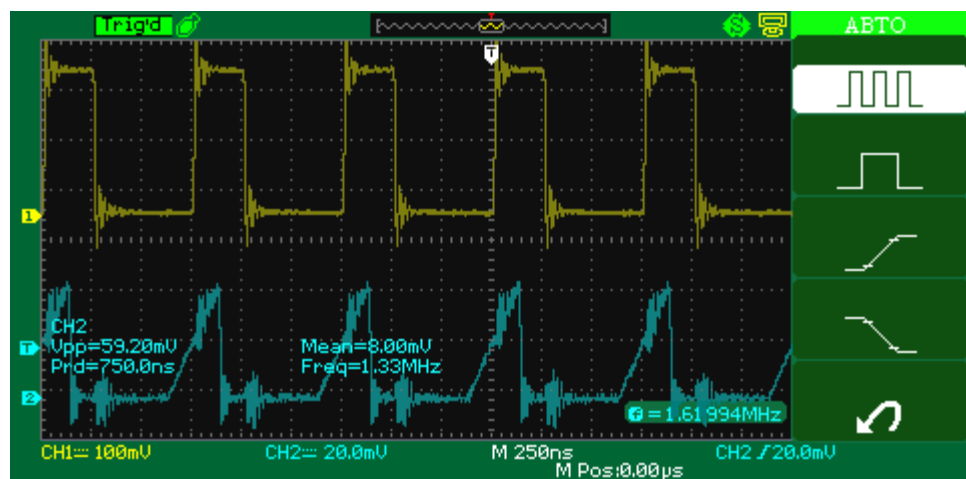


Порт PA15.

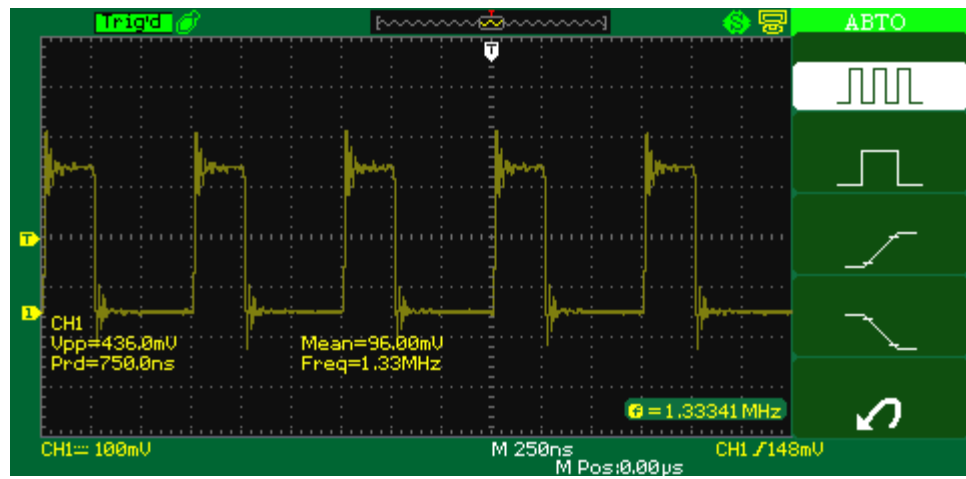


Порт PB0.

б) С оптимизацией

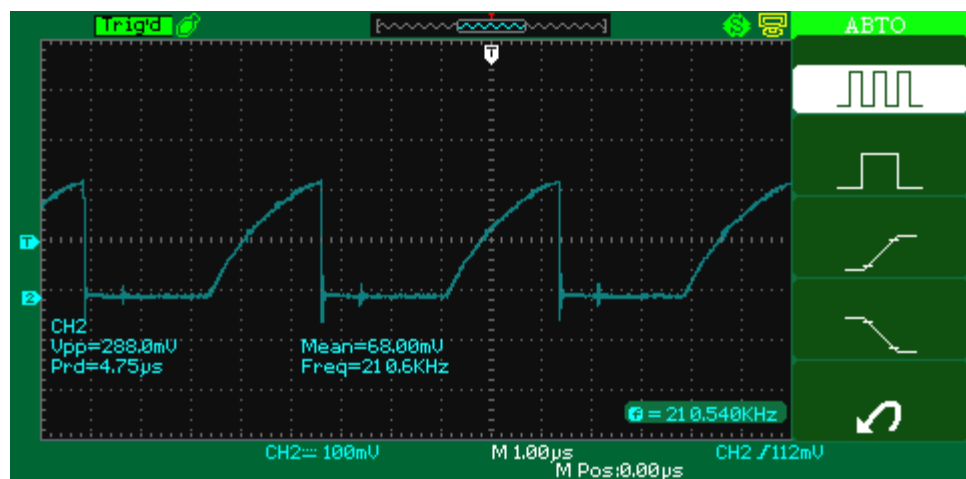
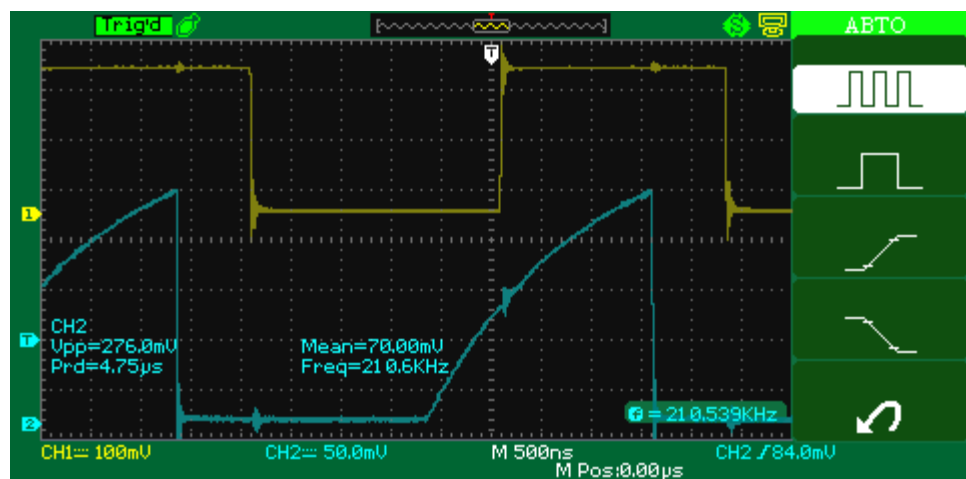


Порт PA15.

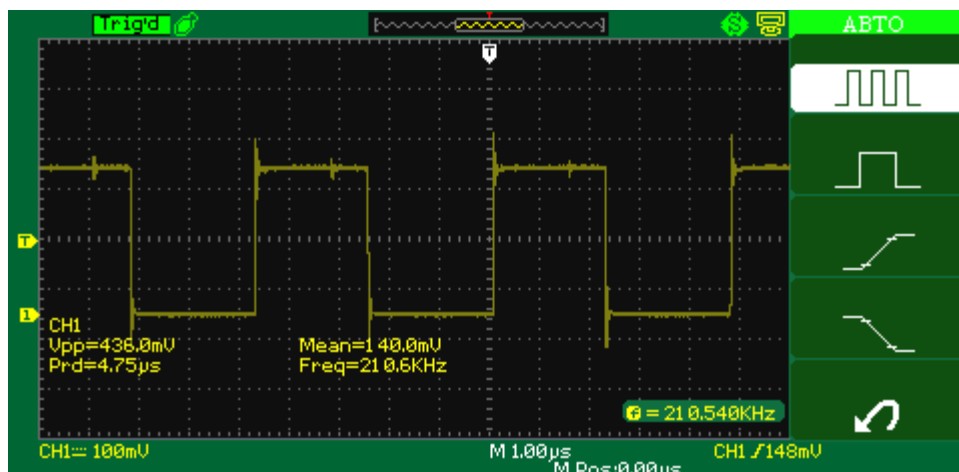


Порт PB0.

в)с задержкой



Порт PA15.



Порт PB0.

5. Сводная таблица

Линия	Измеряемый параметр				
	Частота следования импульсов, КГц	Амплитуда, В	Длительность импульса, нс	Время нарастания фронта, нс	Время спада среза, нс
Оптимизация компилятора отключена 'Level 0(-O0)'					
PA15	363	0,48	2845	1000	100
PB0	363	3	2845	100	100
Оптимизация компилятора включена 'Level 3(-O3)'					
PA15	1333	0,41	752	250	50
PB0	1333	3	752	50	50
Добавлена задержка (~ 0,000009 сек)					
PA15	210	0.67	4800	2150	200
PB0	210	3	4800	200	200

6. Расчет соотношения HCLK с частотой переключения выводов

Без оптимизации: $16\,000\,000 / 363\,000 = 44$

С оптимизацией: $16\,000\,000 / 1\,334\,000 = 12$