

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО»

Институт компьютерных наук и технологий

Высшая школа искусственного интеллекта

Дисциплина:
ПРОГРАММИРОВАНИЕ МИКРОКОНТРОЛЛЕРОВ

ОТЧЕТ

По лабораторной работе №7

Тема: Использование таймеров STM32F200 в режиме
сравнивания выходных величин

Обучающийся гр. 3530201/10001

Нгуен Куок Дат

Руководитель _____ Вербова Наталья Михайловна

Санкт-Петербург 2022

Содержание

1	Цель и постановка задачи	2
1.1	Цель работы	2
1.2	Постановка задачи	2
2	Выполнение задания	2

1 Цель и постановка задачи

1.1 Цель работы

Закрепить навыки работы с низкоуровневыми библиотеками и промежуточным программным обеспечением микроконтроллера. Ознакомиться со способами управления аппаратными таймерами STM32F200. Ознакомиться с приемами отладки программ.

1.2 Постановка задачи

Разработать программу для микроконтроллера (МК) STM32F200, которая включает и выключает светодиоды: один при достижении содержимым таймера заданных значений, а другой при достижении заданных значений содержимым программного счетчика.

2 Выполнение задания

Код программы

```
#include "stm32f2xx_hal.h"

GPIO_InitTypeDef GPIO_InitStruct;

TIM_HandleTypeDef htim;

TIM_OC_InitTypeDef outputChannelInit;

void InitializeGPIO()
{
```

```

RCC->AHB1ENR |= RCC_AHB1ENR_GPIODEN;

/* GPIO base configuration */

GPIO_InitStruct.Pin = (GPIO_PIN_12|GPIO_PIN_15);
GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;;
GPIO_InitStruct.Speed = GPIO_SPEED_LOW;
GPIO_InitStruct.Alternate = GPIO_AF2_TIM4;
HAL_GPIO_Init(GPIOD, &GPIO_InitStruct);
}

```

```

void InitializeTimer()
//1.6MHz baseclock
{
    RCC->APB1ENR |= RCC_APB1ENR_TIM4EN ;

    /* Time base configuration */

    htim.Instance = TIM4;
    htim.Init.Period = 0x500;
    htim.Init.Prescaler = 1250ul;
    htim.Init.ClockDivision = 0;
    htim.Init.RepetitionCounter = 0;
    htim.Init.CounterMode = TIM_COUNTERMODE_UP;
    HAL_TIM_Base_Init(&htim);
}

```

```

void InitializePWMChannel()

```

```

{
    /* Channel base configuration */

    outputChannelInit.OCMode = TIM_OCMode_PWM1;

    outputChannelInit.Pulse = 0x400;

    outputChannelInit.OCFastMode = TIM_OCFAST_DISABLE;

    outputChannelInit.OCpolarity = TIM_OCPOLARITY_HIGH;

    HAL_TIM_OC_Init(&htim);

    HAL_TIM_OC_ConfigChannel(&htim, &outputChannelInit,
TIM_CHANNEL_1);

    /* Enable TIM peripheral counter */

    HAL_TIM_OC_Start(&htim, TIM_CHANNEL_1);
}

void InitializePWMChannel2()
{
    /* Channel 2 base configuration */

    outputChannelInit.OCMode = TIM_OCMode_PWM1;

    outputChannelInit.Pulse = 0x100;

    outputChannelInit.OCFastMode = TIM_OCFAST_DISABLE;

    outputChannelInit.OCpolarity = TIM_OCPOLARITY_HIGH;

    HAL_TIM_OC_Init(&htim);

    HAL_TIM_OC_ConfigChannel(&htim, &outputChannelInit,
TIM_CHANNEL_4);

    /* Enable TIM peripheral counter */

    HAL_TIM_OC_Start(&htim, TIM_CHANNEL_4);
}

```

```

int main()
{
    InitializeGPIO();
    InitializeTimer();
    InitializePWMChannel();
    InitializePWMChannel2();
    for (;;)
    {
    }
}

```

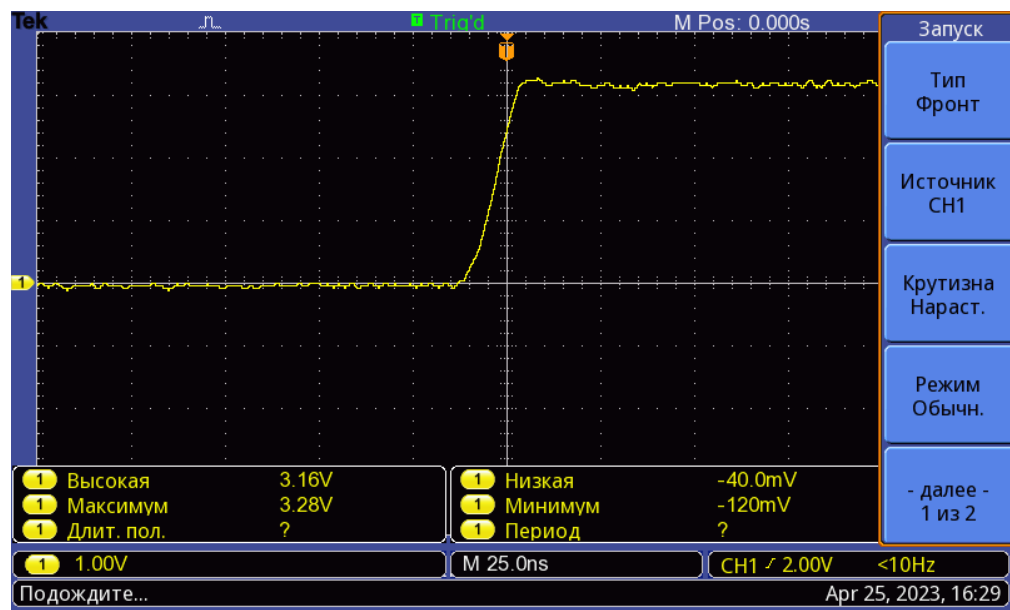


Рис. 1: Фронт нарастания

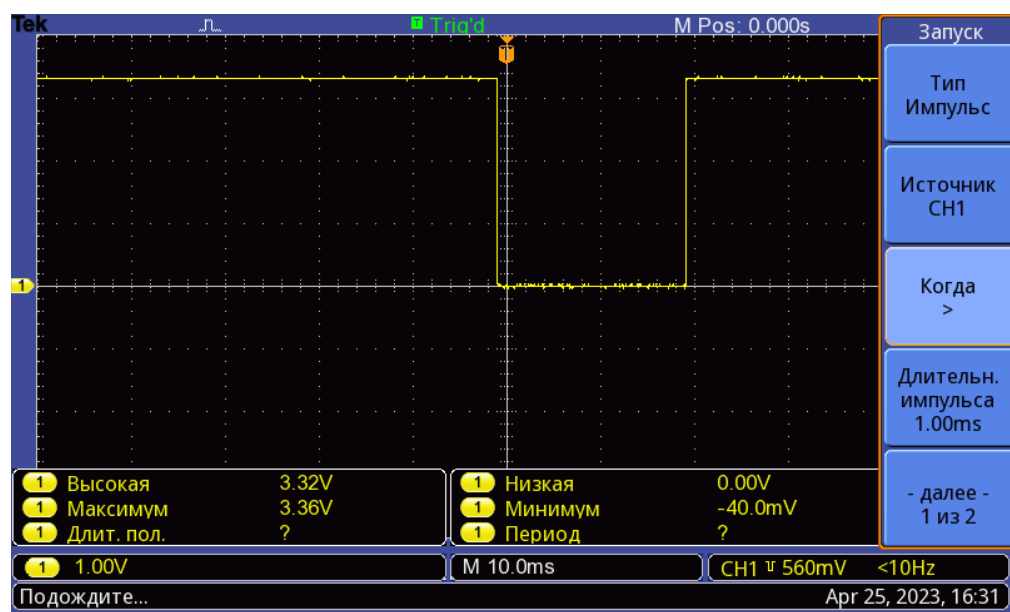


Рис. 2: Импульс, отрицательная полярность, $> 1\text{ms}$