

**Московский государственный технический университет им. Н. Э. Баумана**

**Кафедра ИУ-4  
«Проектирование и технология производства ЭС»**

# **Журнал практических работ**

**по курсу: «Проектирование  
микропроцессорных систем»**

**Для студентов приборостроительных специальностей**

**20\_24/25 учебный год**

Студент \_\_\_\_\_  
(фамилия, и. о.)

Группа \_\_\_\_\_ ИУ4-73Б

Преподаватель Леонидов В. В. Допуск к экзамену (зачету) \_\_\_\_\_ Подпись \_\_\_\_\_  
(фамилия, и. о.) (кол-во баллов)

**Москва  
2024**

Отчет по лабораторной работе № 1 «Основы UAL»			
дата	Оценка (max 5)	Бонус за сложность	подпись

**Цели работы:** На языке Unified Assembler Language (UAL) разработать код программы для отладочной платы NUCLEO-F103RB с микроконтроллером STM32F103RBT6, позволяющую изменять уровень яркости светодиода между 3 уровнями по нажатию кнопки.

### Задачи работы:

Составить теоретическое описание ШИМ сигнала и его характеристик,

Разработать код программы по управлению ШИМ сигналом,

Получить осциллограмму ШИМ сигнала при различных уровнях яркости светодиода.

### Теоретическая часть

Широкоимпульсная модуляция (ШИМ, англ. pulse-width modulation (PWM)) — метод управления мощностью методом пульсирующего включения и выключения потребителя энергии.

Основной причиной применения ШИМ является стремление к повышению КПД при построении вторичных источников питания электронной аппаратуры и в других узлах, например, ШИМ используется для регулировки яркости подсветки мониторов и дисплеев в телефонах и т.п.

Важными характеристиками ШИМ сигнала являются:

амплитуда ( $U$ ),

частота ( $f$ ),

скважность ( $S$ ) - отношение периода ( $T$ ) к длительности сигнала "высокого" напряжения ( $t_{вкл}$ ); или же коэффициент заполнения ( $D$ ) - обратная скважности величина.

Примеры 3 разных значений скважности отображены на рисунке 1.

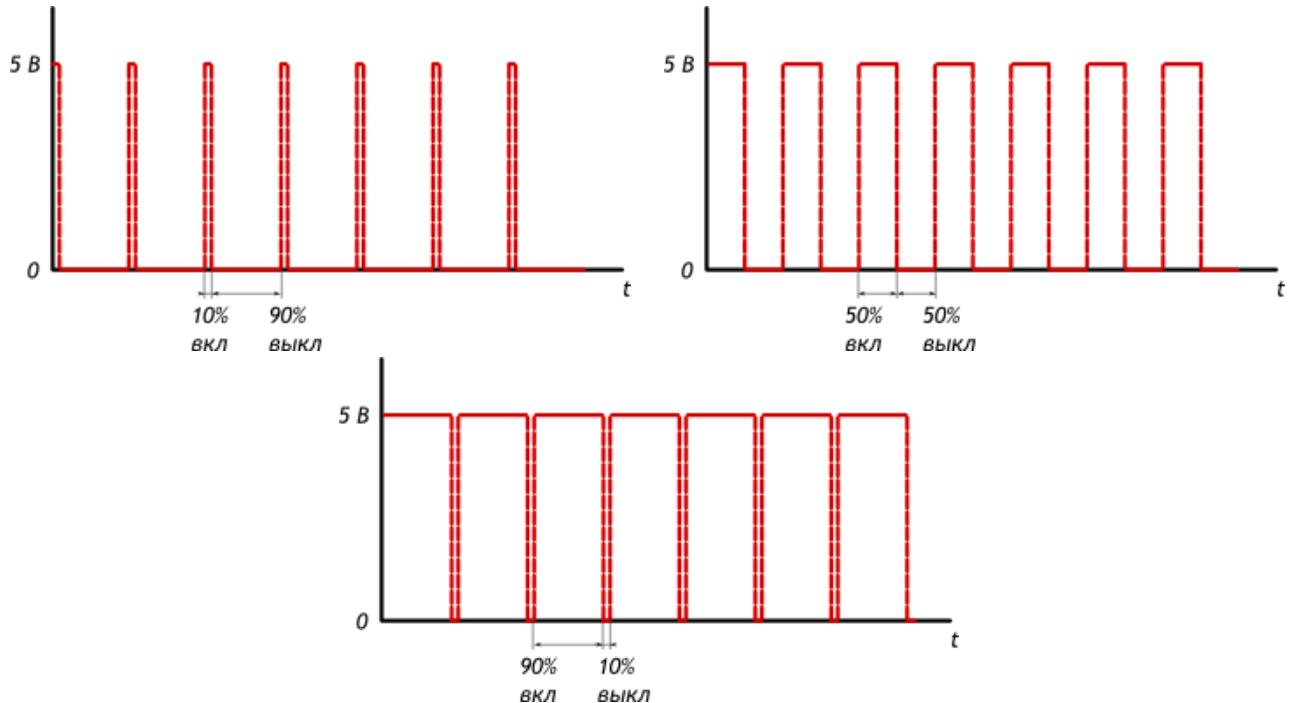


Рисунок 1 - примеры различных значений скважности

Чем выше коэффициент заполнения (или ниже скважность), тем больше мощность передаваемого сигнала и ярче светится светодиод.

## Практическая часть

Листинг 1 - основная часть программы

Reset\_Handler:

```
ldr      r0, =estack
mov      sp, r0          /* set stack pointer */

ldr      r0, =PERIPH_BB_BASE + \
           (RCC_APB2ENR-PERIPH_BASE)*32 + \
           2*4                @ вычисляем адрес для
                           2-го бита регистра RCC_APB2ENR
mov      r1, #1            @ включаем тактирование порта А (во 2-й
                           бит RCC_APB2ENR пишем '1')
str      r1, [r0]          @ загружаем это значение
ldr      r0, =GPIOA_CRL    @ адрес порта
mov      r1, #0x0003        @ 4-битная маска настроек для Output mode
                           50MHz, Push-Pull ("0011")
ldr      r2, [r0]          @ считатьпорт
bfi      r2, r1, #20, #4   @ скопировать биты маски в позицию PIN5
str      r2, [r0]          @ загрузить результат в регистр настройки
                           порта
ldr      r0, =GPIOA_BSRR    @ адрес порта выходных сигналов
ldr      r3, =PERIPH_BB_BASE + \
           (RCC_APB2ENR-PERIPH_BASE)*32 + \
           4*4                @ вычисляем адрес для BitBanding 2-го
                           бита регистра RCC_APB2ENR
mov      r4, #1            @ включаем тактирование порта С
str      r4, [r3]          @ загружаем это значение
ldr      r3, =GPIOC_CRH    @ адрес порта
mov      r4, #0b1000        @ 4-битная маска настроек для Input
                           mode 50MHz, with pull-up/pull-down
ldr      r5, [r3]          @ считатьпорт
bfi      r5, r4, #20, #4   @ скопировать биты маски в позицию PIN5
str      r5, [r3]          @ загрузить результат в регистр настройки
                           порта
mov      r9, #1            @ r9 означает текущее состояние яркости

loop:
ldr      r7, =GPIOC_IDR    @ Бесконечный цикл
ldr      r6, [r7]          @ считать входные значения
and      r6, #0x2000        @ загружаем в порт
cmp      r6, #0
beq      button_pressed   button_pressed
bge      button_unpressed button_unpressed

button_pressed:
ldr      r10, =0x00FF        @ псевдоинструкция Thumb (загрузить
                           константу в регистр)
bl       delay             @ задержка
ldr      r6, [r7]          @ загружаем в порт
and      r6, #0x2000
cmp      r6, #0
it eq
addeq   r9, #0x001

button_unpressed:
cmp      r9, #0x003
it eq
moveq   r9, #0
bl       led_light          led_light
b       loop

led_light:                  @ функция отправки 1
                           ШИМ сигнала, сперва нужно записать в R9
                           уровень яркости
```

```

push {lr}
cmp r9, #0
beq led_light_little
cmp r9, #1
beq led_light_medium
cmp r9, #2
beq led_light_large
led_light_little:
    ldr r1, =GPIO_BSRR_BS5      @ устанавливаем вывод в '1'
    str r1, [r0]                @ загружаем впорт
    ldr r10, =0x00FF
    bl delay                   @ задержка
    ldr r1, =GPIO_BSRR_BR50    @ сбрасываем в '0'
    str r1, [r0]                @ загружаем впорт
    ldr r10, =0x0F00            @ псевдоинструкция Thumb (загрузить
                                константу в регистр)
    bl delay                   @ задержка
    b led_light_end
led_light_medium:
    ldr r1, =GPIO_BSRR_BS5      @ устанавливаем вывод в '1'
    str r1, [r0]                @ загружаем впорт
    ldr r10, =0x05FF            @ псевдоинструкция Thumb (загрузить
                                константу в регистр)
    bl delay                   @ задержка
    ldr r1, =GPIO_BSRR_BR50    @ сбрасываем в '0'
    str r1, [r0]                @ загружаем впорт
    ldr r10, =0x0A00            @ псевдоинструкция Thumb (загрузить
                                константу в регистр)
    bl delay                   @ задержка
    b led_light_end
led_light_large:
    ldr r1, =GPIO_BSRR_BS5      @ устанавливаем вывод в '1'
    str r1, [r0]                @ загружаем впорт
    ldr r10, =0x0FF0            @ псевдоинструкция Thumb (загрузить
                                константу в регистр)
    bl delay                   @ задержка
    ldr r1, =GPIO_BSRR_BR50    @ сбрасываем в '0'
    str r1, [r0]                @ загружаем впорт
    ldr r10, =0x000F            @ псевдоинструкция Thumb (загрузить
                                константу в регистр)
    bl delay                   @ задержка
    b led_light_end
led_light_end:
    pop {lr}
    bx lr                      @ возвращаемся к началу цикла

delay:
    subs r10, #1                @ функция задержки, сперва нужно записать
    it NE                      @ SUB с установкой флагов результата
    bne delay                  @ переход, если Z==0 (результат вычитания
                                не равен нулю)
    bx lr                      @ выход из подпрограммы (переход к адресу
                                в регистре LR - вершина стека)

```

Результаты работы программы представлены на осциллографмах на рисунках 2-5.

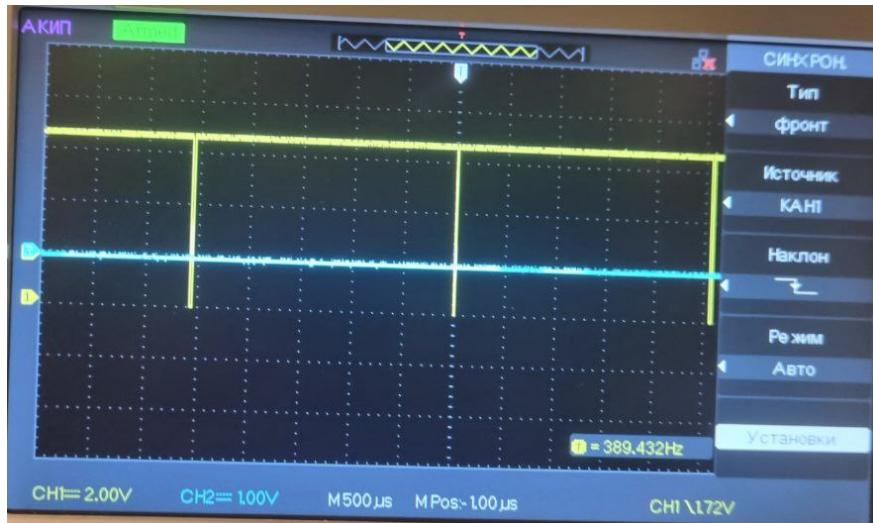


Рисунок 2 - ШИМ сигнал при максимальной яркости

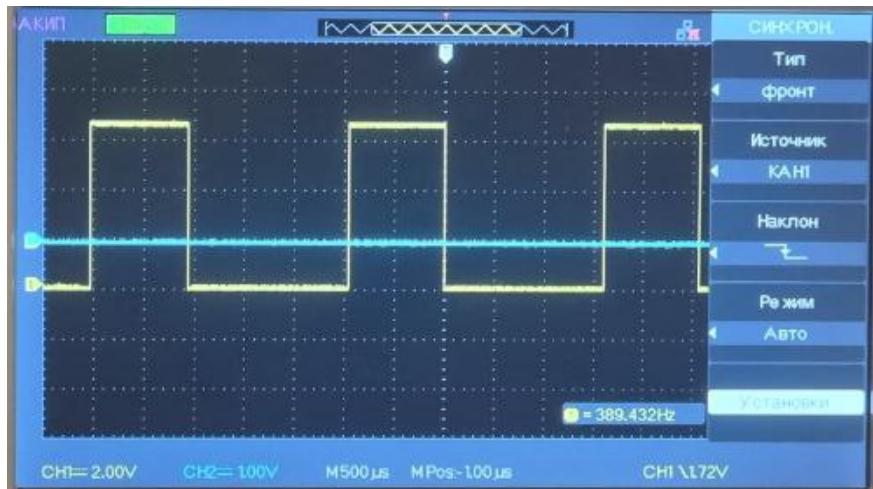


Рисунок 3 - ШИМ сигнал при средней яркости

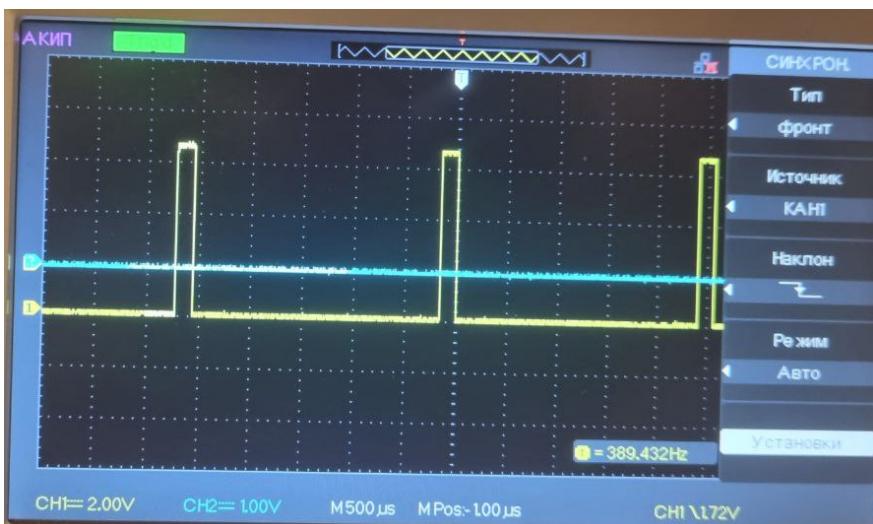


Рисунок 4- ШИМ сигнал при минимальной яркости

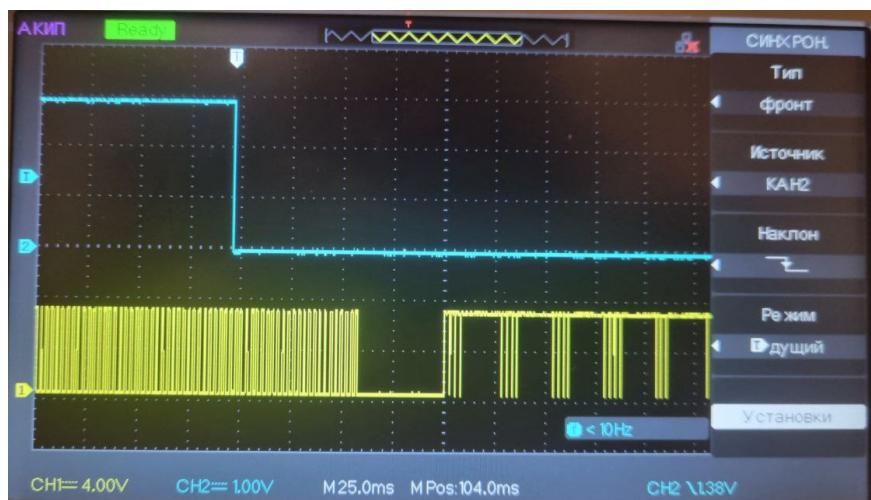


Рисунок 5 - изменении яркости при нажатии кнопки

Представленные выше осциллографы в полной мере отражают принцип работы кода, представленного в листинге 1.

### Выводы

В ходе выполнения лабораторной работы успешно разработана программа на языке UAL для контроля яркости свечения светодиода путём нажатия на кнопку, получены навыки работы с отладочной платой Nucleo.