Федеральное государственное автономное образовательное

учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Дисциплина: Проектирование вычислительных систем

Лабораторная работа 1

#### Вариант 4

**Выполнили:** Барсуков Максим Андреевич,

группа P3415

Стригалев Никита Сергеевич,

группа P3412

**Преподаватель:**

Пинкевич Василий Юрьевич

2025 г.

Санкт-Петербург

# Содержание

[**Задание 3**](#_9gtkkyqf3v0l)

[**Используемые контакты 4**](#_ae7x5l8yijsz)

[Описание контактов 4](#_zdp6by6vd5io)

[**Блок-схемы 5**](#_4i43kas9316o)

[Схема алгоритма: 5](#_y1r641vwu3x6)

[Подпрограмма анимации переполнения: 6](#_yrvsrjbxlgqe)

[**Описание работы алгоритма 7**](#_d66vuu4q5j37)

[**Код драйвера 8**](#_r3hm7c8yhuum)

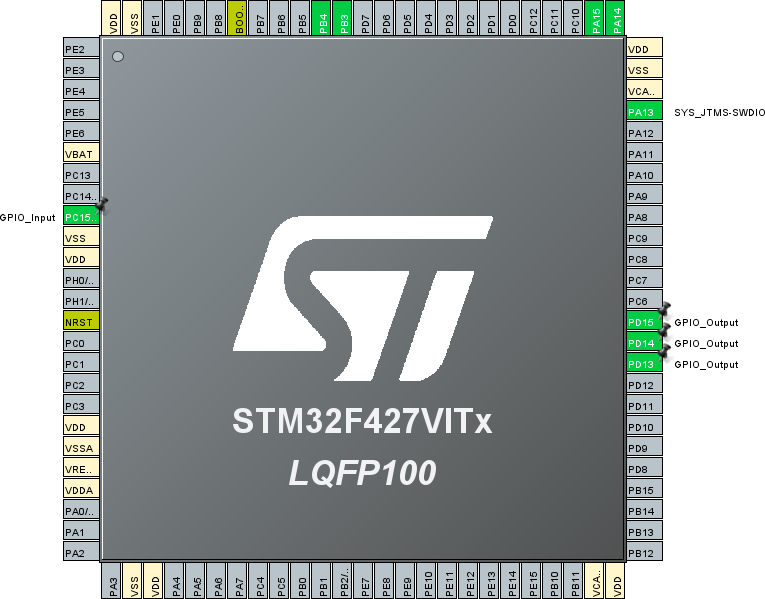
[**Вывод 10**](#_l4bkvn8lufa2)

# Задание

##### Разработать и реализовать драйверы управления светодиодными индикаторами и чтения состояния кнопки стенда SDK-1.1М (индикаторы и кнопка расположены на боковой панели стенда). Написать программу с использованием разработанных драйверов в соответствии с вариантом задания.

##### Реализовать двоичный двухразрядный счетчик на светодиодах с возможностью вычитания (использовать зеленый светодиод и один из двух цветов двухцветного). Быстрое нажатие кнопки должно прибавлять единицу к отображаемому на светодиодах двоичному числу. По переполнению счетчика должна отображаться простая анимация: мигание обоими светодиодами, затем количество миганий зеленым светодиодом, равное количеству переполнений с момента перезагрузки микроконтроллера. Долгое нажатие кнопки должно вычитать единицу из отображаемого на светодиодах двоичного числа. Если происходит вычитание из нуля, количество переполнений уменьшается на единицу, и отображается анимация, аналогичная анимации при переполнении.

# Используемые контакты



### Описание контактов

###### **PC15** - перехватывает нажатие кнопки (настроен на GPIO\_Input)

###### **PD13** - управляет зеленым светодиодом (настроен на GPIO\_Output)

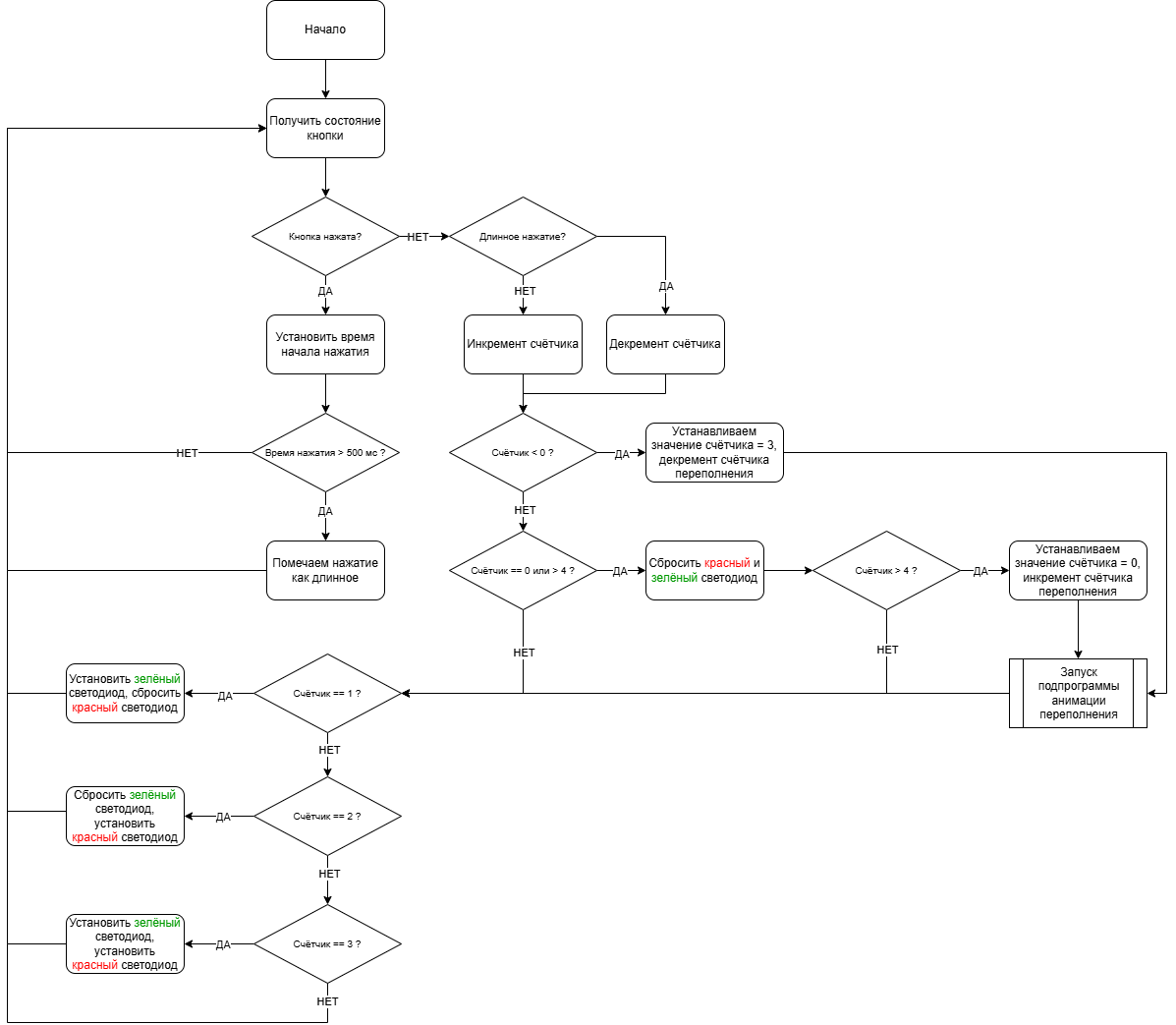
###### **PD14** - управляет желтым светодиодом (настроен на GPIO\_Output)

###### **PD15** - управляет красным светодиодом (настроен на GPIO\_Output)

###### **PB3, PB4, PA13, PA14, PA15** - J-Tag для отладки

# Блок-схемы

## Схема алгоритма:



## Подпрограмма анимации переполнения:

## 

# Описание работы алгоритма

###### Алгоритм работает в цикле, постоянно отслеживая состояние кнопки. При обнаружении нажатия система регистрирует этот факт и измеряет его длительность. Различаются два типа нажатий: короткое и долгое. В момент отпускания кнопки анализируется тип произошедшего нажатия: краткое нажатие увеличивает значение счётчика, а долгое — уменьшает его.

###### Счётчик ограничен диапазоном от 0 до 3, так как для отображения доступны только две лампочки. При попытке выйти за эти границы (например, при значении 4 или -1) возникает переполнение. В этом случае активируется специальная анимация, которая сигнализирует о выходе за пределы диапазона: зелёный светодиод указывает на переполнение "сверху", а красный — "снизу".

# Код драйвера

void turn\_off\_leds() {

HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_13, GPIO\_PIN\_RESET);

HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_14, GPIO\_PIN\_RESET);

HAL\_Delay(150);

}

void pulse\_dual\_leds() {

HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_13, GPIO\_PIN\_SET);

HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_14, GPIO\_PIN\_SET);

HAL\_Delay(150);

HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_13, GPIO\_PIN\_RESET);

HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_14, GPIO\_PIN\_RESET);

HAL\_Delay(150);

}

void pulse\_single\_led() {

HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_13, GPIO\_PIN\_SET);

HAL\_Delay(150);

HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_13, GPIO\_PIN\_RESET);

HAL\_Delay(150);

}

void show\_overflow\_feedback(int times) {

turn\_off\_leds();

pulse\_dual\_leds();

for (int i = 0; i < times; i++) {

pulse\_single\_led();

}

}

int main(void)

{

int led\_state = 0;

int overflow\_level = 0;

const int press\_threshold = 500;

HAL\_Init();

SystemClock\_Config();

while (1)

{

int button\_is\_down = !HAL\_GPIO\_ReadPin(GPIOC, GPIO\_PIN\_15);

int tick\_start = 0;

if (button\_is\_down) {

tick\_start = HAL\_GetTick();

while (!HAL\_GPIO\_ReadPin(GPIOC, GPIO\_PIN\_15));

int press\_time = HAL\_GetTick() - tick\_start;

if (press\_time < press\_threshold) {

led\_state++;

if (led\_state > 3) {

led\_state = 0;

overflow\_level++;

show\_overflow\_feedback(overflow\_level);

}

} else {

led\_state--;

if (led\_state < 0) {

led\_state = 3;

overflow\_level--;

if (overflow\_level < 0) {

overflow\_level = 0;

} else {

show\_overflow\_feedback(overflow\_level);

}

}

}

}

switch (led\_state) {

case 0:

HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_13, GPIO\_PIN\_RESET);

HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_14, GPIO\_PIN\_RESET);

break;

case 1:

HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_13, GPIO\_PIN\_SET);

HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_14, GPIO\_PIN\_RESET);

break;

case 2:

HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_13, GPIO\_PIN\_RESET);

HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_14, GPIO\_PIN\_SET);

break;

case 3:

HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_13, GPIO\_PIN\_SET);

HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_14, GPIO\_PIN\_SET);

break;

}

}

}

# Вывод

#### В результате выполнения лабораторной работы были освоены принципы работы с портами ввода-вывода общего назначения (GPIO). На практике эти знания были применены для разработки и программирования цифрового счётчика, использующего два светодиода в качестве индикатора состояния.