Федеральное государственное автономное образовательное учреждение высшего образования

Университет ИТМО

Дисциплина: Проектирование вычислительных систем

Лабораторная работа 1

Вариант 4

Выполнили:
Марков Петр Денисович
Кривоносов Егор Дмитриевич

Группа: P34111

Преподаватель:
Пинкевич Василий Юрьевич

2022 г.

Санкт-Петербург

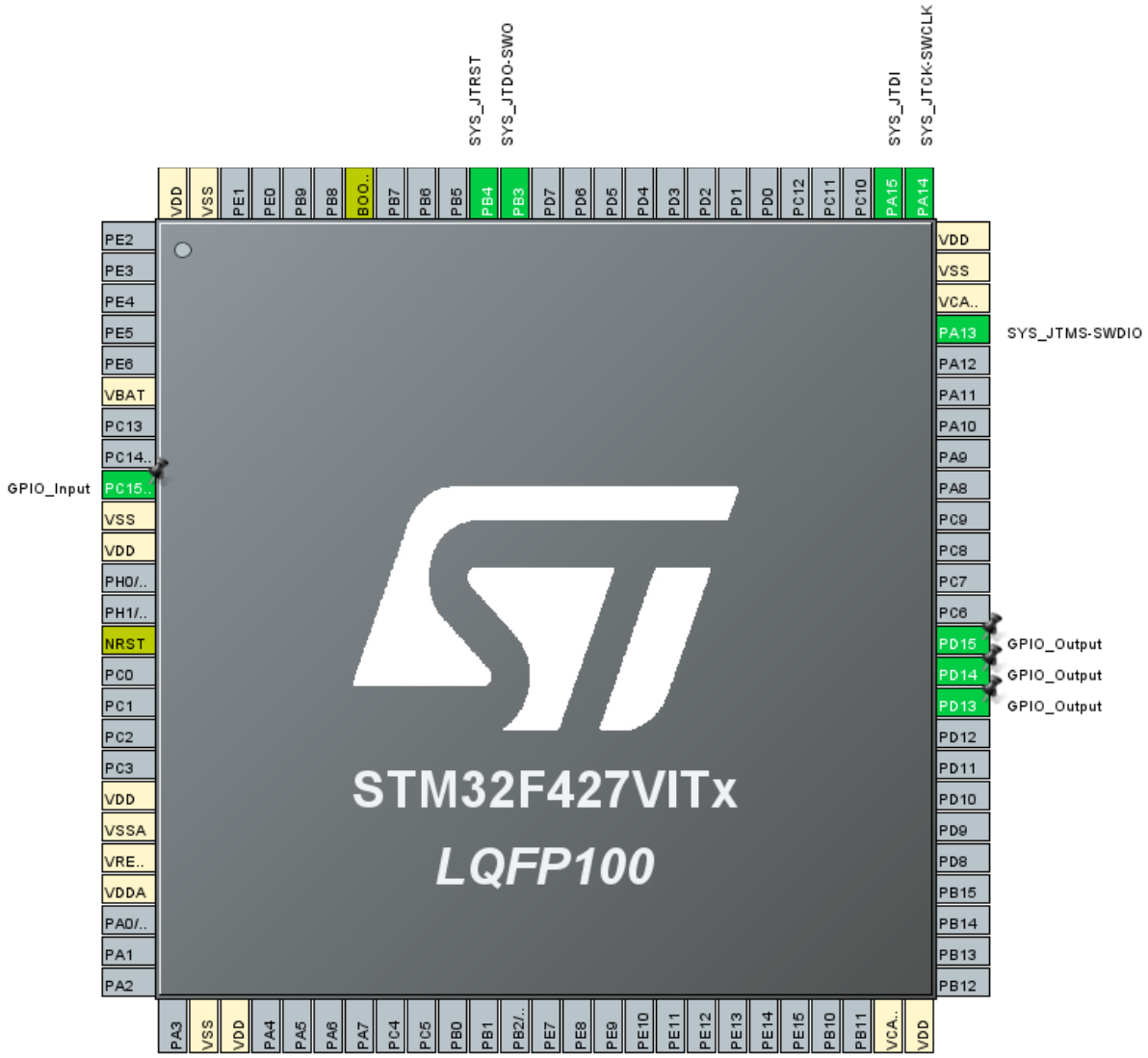
Оглавление

Задание

Разработать и реализовать драйверы управления светодиодными индикаторами и чтения состояния кнопки стенда SDK-1.1M (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

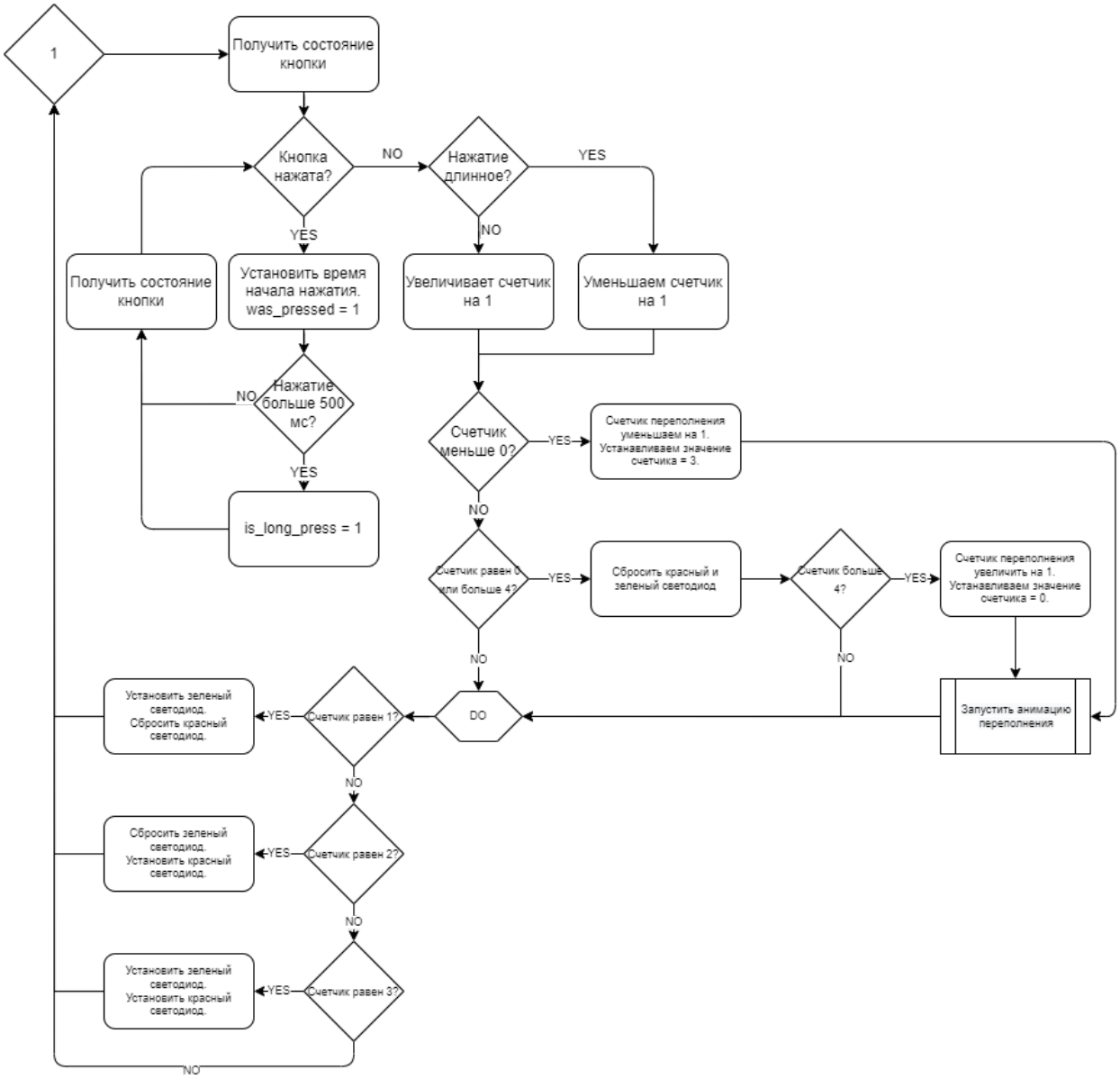


Описание контактов

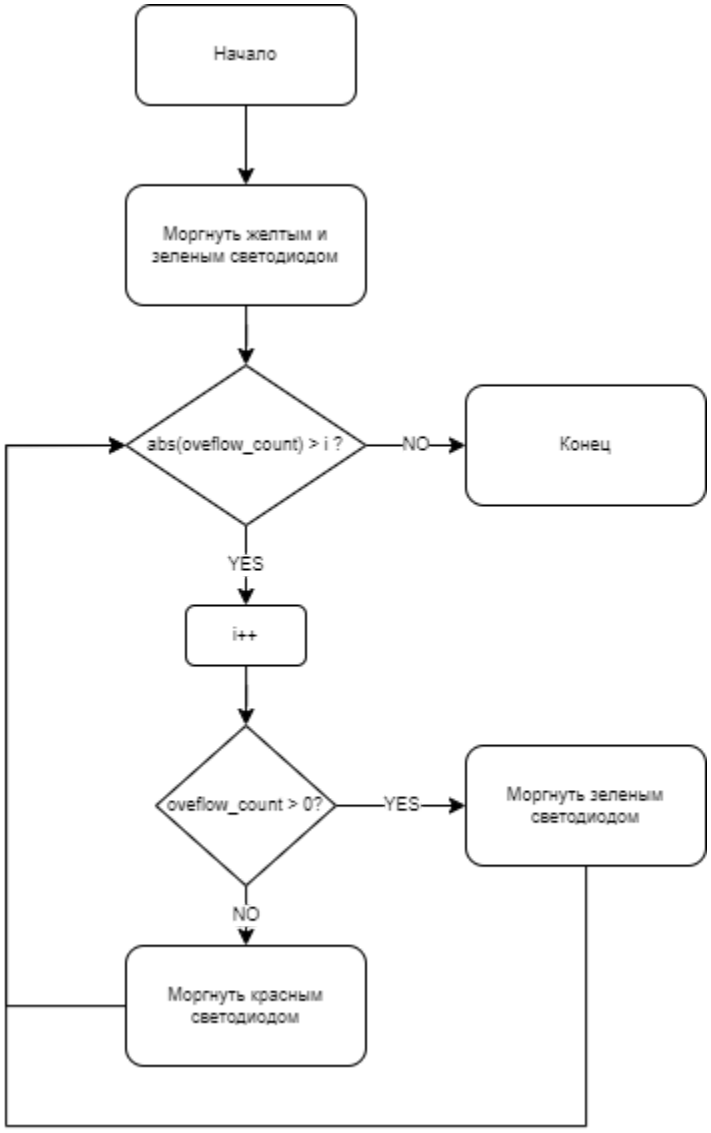
PC15 - перехватывает нажатие кнопки (настроен на GPIO_Input)
PD13 - управляет зеленым светодиодом (настроен на GPIO_Output)
PD14 - управляет желтым светодиодом (настроен на GPIO_Output)
PD15 - управляет красным светодиодом (настроен на GPIO_Output)
PB3, PB4, PA13, PA14, PA15 - J-Tag для отладки

Блок-схемы

Основная схема алгоритма



Подпрограмма “анимация переполнения”



Описание работы алгоритма

Наш алгоритм циклично проверяет состояние кнопки: нажата она или нет. Если кнопка нажата, то фиксируем это и проверяем, как долго происходило нажатие. Если нажатие долгое, то также сохраняем это. Оба эти состояния нужны нам, чтобы при отпускании кнопки мы могли отследить то, что кнопка была нажата и как долго происходило нажатие. При долгом нажатии мы уменьшаем счетчик, при коротком - увеличиваем. Т.к. У нас всего две лампочки, то максимальное число у счетчика = 3. При остальных значениях (все, кроме 0,1,2,3) мы не способны отобразить число корректно, поэтому происходит переполнение в большую или меньшую сторону. Мы запускаем для них отдельную анимацию, показывающую на сколько больше или меньше значение

счетчика отлично от граничных. В зависимости от того, в какую сторону переполнение, горят разные светодиоды: зеленый при переполнении в большую сторону, красный - в меньшую.

Код драйвера

```
void overflow_animation(int overflow_count){
      HAL_GPIO_WritePin(GPIOD, GPIO_PIN_13, GPIO_PIN_RESET);
      HAL_GPIO_WritePin(GPIOD, GPIO_PIN_15, GPIO_PIN_RESET);
      HAL_Delay(250);
      HAL_GPIO_WritePin(GPIOD, GPIO_PIN_13, GPIO_PIN_SET);
      HAL_GPIO_WritePin(GPIOD, GPIO_PIN_14, GPIO_PIN_SET);
      HAL_Delay(250);
      HAL_GPIO_WritePin(GPIOD, GPIO_PIN_13, GPIO_PIN_RESET);
      HAL_GPIO_WritePin(GPIOD, GPIO_PIN_14, GPIO_PIN_RESET);
      HAL_Delay(500);

      for (int i = 0; i < abs(overflow_count); i++){
          if (overflow_count > 0){
             HAL_GPIO_WritePin(GPIOD, GPIO_PIN_13, GPIO_PIN_SET);
             HAL_Delay(250);
             HAL_GPIO_WritePin(GPIOD, GPIO_PIN_13, GPIO_PIN_RESET);
             HAL_Delay(250);
          } else {
             HAL_GPIO_WritePin(GPIOD, GPIO_PIN_15, GPIO_PIN_SET);
             HAL_Delay(250);
             HAL_GPIO_WritePin(GPIOD, GPIO_PIN_15, GPIO_PIN_RESET);
             HAL_Delay(250);
          }
      }
  }

 int count = 0;
 int overflow_count = 0;
 _Bool was_pressed = 0;
 _Bool is_long_press = 0;
 uint32_t wait = 500;

  while (1)
  {
      int start_time;
      int button_state = HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_15);
      was_pressed = 0;
      is_long_press = 0;

      if(!button_state){
          start_time = HAL_GetTick();
      }
      while (!button_state){
```

```
            was_pressed = 1;
            if (HAL_GetTick() - start_time >= wait){
                is_long_press = 1;
            }
            button_state = HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_15);
        }

        if (was_pressed){
            !is_long_press ? count++ : count--;
        }

        if (count < 0){
            count = 3;
            overflow_count--;
            overflow_animation(overflow_count);
        }
        if (count == 0 || count >= 4){
           HAL_GPIO_WritePin(GPIOD, GPIO_PIN_13, GPIO_PIN_RESET);
           HAL_GPIO_WritePin(GPIOD, GPIO_PIN_15, GPIO_PIN_RESET);
           if (count >= 4){
               count = 0;
               overflow_count++;
               overflow_animation(overflow_count);
           }
        }
        if (count == 1){
           HAL_GPIO_WritePin(GPIOD, GPIO_PIN_13, GPIO_PIN_SET);
           HAL_GPIO_WritePin(GPIOD, GPIO_PIN_15, GPIO_PIN_RESET);
        }
        if (count == 2){
           HAL_GPIO_WritePin(GPIOD, GPIO_PIN_13, GPIO_PIN_RESET);
           HAL_GPIO_WritePin(GPIOD, GPIO_PIN_15, GPIO_PIN_SET);
        }
        if (count == 3){
           HAL_GPIO_WritePin(GPIOD, GPIO_PIN_13, GPIO_PIN_SET);
           HAL_GPIO_WritePin(GPIOD, GPIO_PIN_15, GPIO_PIN_SET);
        }
    }
```

Вывод

В ходе лабораторной работы мы познакомились с интерфейсами ввода/вывода общего назначения и реализовали “счетчика” с помощью двух светодиодов.