

Университет ИТМО
ПИиКТ

Лабораторная работа №1:
«Интерфейсы ввода-вывода общего назначения (GPIO)»

По дисциплине: Проектирование вычислительных систем

Вариант: 3

Выполнили:
Неграш А. В., Р34301
Перевозчиков И. С., Р34301

Преподаватель:
Пинкевич В. Ю.

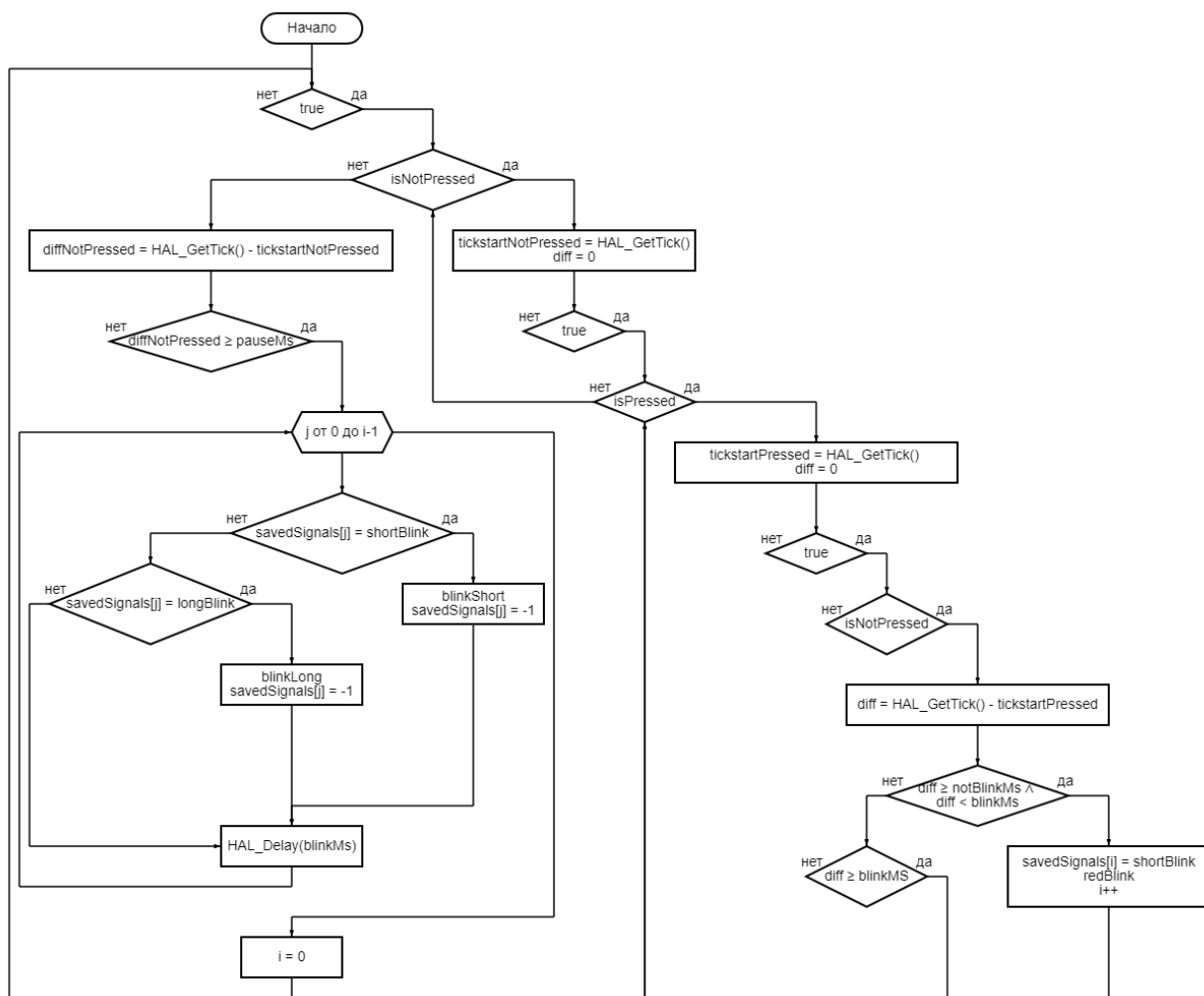
Санкт-Петербург
2023

1. Задание

Разработать и реализовать драйверы управления светодиодными индикаторами и чтения состояния кнопки стенда SDK-1.1M (расположены на боковой панели стенда). Контакты подключения кнопки и светодиодов должны быть настроены в режиме GPIO. Функции и другие компоненты драйверов должны быть универсальными, т.е. пригодными для использования в любом из вариантов задания и не должны содержать прикладной логики программы. Функции драйверов должны быть неблокирующими, то есть не должны содержать ожиданий события (например, нажатия кнопки). Также, в драйверах не должно быть пауз с активным ожиданием функция HAL_Delay() и собственные варианты аналогичной реализации. Обработка нажатия кнопки в программе должна включать программную защиту от дребезга.

Реализовать «передатчик» азбуки Морзе. Последовательность из нажатий кнопки (короткое – точка, длинное – тире) запоминается и после сигнала окончания ввода (долгая пауза) начинает «отправляться» при помощи зелёного светодиода, последовательностью быстрых (точка) и долгих (тире) мерцаний. Во время ввода последовательности после каждого нажатия двухцветный светодиод должен коротким мерцанием индигировать, какой сигнал был введён (мигание жёлтым – точка, мигание красным – тире).

2. Блок-схема прикладного алгоритма



3. Исходный код

```
_Bool isPressed() { return HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_15) == GPIO_PIN_RESET; }
_Bool isNotPressed() { return HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_15) == GPIO_PIN_SET; }

int main(void) {
    HAL_Init();
    SystemClock_Config();
    MX_GPIO_Init();

    uint32_t savedSignals[255];
    uint32_t i = 0;
    uint32_t shortBlink = 0;
    uint32_t longBlink = 1;
    uint32_t notBlinkMs = 100;
    uint32_t blinkMs = 500;
    uint32_t pauseMs = 2000;

    while (1) {
        if (isNotPressed()) {
            uint32_t tickstartNotPressed = HAL_GetTick();
            while (1) {
                if (isPressed()) {
                    uint32_t tickstartPressed = HAL_GetTick();
                    uint32_t diff = 0;
                    while (1) {
                        if (isNotPressed()) {
                            diff = HAL_GetTick() - tickstartPressed;
                            if (diff >= notBlinkMs && diff < blinkMs) {
                                savedSignals[i] = shortBlink;
                                HAL_GPIO_WritePin(GPIOD, GPIO_PIN_15, GPIO_PIN_SET);
                                HAL_Delay(25);
                                HAL_GPIO_WritePin(GPIOD, GPIO_PIN_15, GPIO_PIN_RESET);
                                i++;
                            } else if (diff >= blinkMs) {
                                savedSignals[i] = longBlink;
                                HAL_GPIO_WritePin(GPIOD, GPIO_PIN_14, GPIO_PIN_SET);
                                HAL_Delay(25);
                                HAL_GPIO_WritePin(GPIOD, GPIO_PIN_14, GPIO_PIN_RESET);
                                i++;
                            }
                        }
                        break;
                    }
                }
                break;
            }
        }
    }
}
```

```

uint32_t diffNotPressed = HAL_GetTick() - tickstartNotPressed;
if (diffNotPressed >= pauseMs) {
    for (uint32_t j = 0; j < i; j++) {
        if (savedSignals[j] == shortBlink) {
            HAL_GPIO_WritePin(GPIOD, GPIO_PIN_13, GPIO_PIN_SET);
            HAL_Delay(notBlinkMs);
            HAL_GPIO_WritePin(GPIOD, GPIO_PIN_13, GPIO_PIN_RESET);
            savedSignals[j] = -1;
        } else if (savedSignals[j] == longBlink) {
            HAL_GPIO_WritePin(GPIOD, GPIO_PIN_13, GPIO_PIN_SET);
            HAL_Delay(blinkMs);
            HAL_GPIO_WritePin(GPIOD, GPIO_PIN_13, GPIO_PIN_RESET);
            savedSignals[j] = -1;
        }
        HAL_Delay(blinkMs);
    }
    i = 0;
    break;
}
}
}
}
}

```

4. Вывод

Итак, в процессе выполнения данной лабораторной работы мы ознакомились с устройством станда SDK 1.1M и изучили работу с интерфейсами ввода-вывода в микроконтроллерах, а также разработали собственную программу для управления световыми индикаторами при помощи кнопки, в которой продемонстрировали полученные знания.