

Министерство науки и высшего образования Российской Федерации  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧЕРЕЖДЕНИЕ ВЫСШЕГО  
ОБРАЗОВАНИЯ

“НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО”

Факультет ПИиКТ, кафедра ВТ



**УНИВЕРСИТЕТ ИТМО**

ОТЧЁТ

По лабораторной работе №4

“Интерфейс I2C и матричная клавиатура”

По предмету: Проектирование вычислительных систем

Вариант 1

Студенты:

Андрейченко Леонид Вадимович

Степанов Михаил Алексеевич

Группа Р34301

Преподаватель:

Пинкевич Василий Юрьевич

Санкт-Петербург

2023

## Цель работы

- Получить базовые знания об интерфейсе I2C и особенностях передачи данных по данному интерфейсу.
- Получить базовые знания об устройстве и принципах работы контроллера интерфейса I2C в микроконтроллерах и получить навыки его программирования.

## Задание

Разработать программу, которая использует интерфейс I2C для считывания нажатий кнопок клавиатуры стенда SDK-1.1.

Подсистема опроса клавиатуры должна удовлетворять следующим требованиям:

- реализуется защита от дребезга;
- нажатие кнопки фиксируется сразу после того, как было обнаружено, что кнопка нажата (с учетом защиты от дребезга), а не в момент отпускания кнопки; если необходимо, долгое нажатие может фиксироваться отдельно;
- кнопка, которая удерживается дольше, чем один цикл опроса, не считается повторно нажатой до тех пор, пока не будет отпущена (нет повторений);
- распознается и корректно обрабатывается множественное нажатие (при нажатии более чем одной кнопки считается, что ни одна кнопка не нажата, если это не противоречит требованиям к программе);
- всем кнопкам назначаются коды от 1 до 12 (порядок на усмотрение исполнителей).

Программа должна иметь два режима работы, переключение между которыми производится по нажатию кнопки на боковой панели стенда:

- режим тестирования клавиатуры;
- прикладной режим.
- Уведомление о смене режима выводится в UART.
- В режиме тестирования клавиатуры программа выводит в UART коды нажатых кнопок.
- В прикладном режиме программа обрабатывает нажатия кнопок и выполняет действия в соответствии с вариантом задания.

## Организация программы

Основная программа представляет из себя цикл, в котором происходит проверка текущего режима работы программы.

Если установлен мод тестирования, то по нажатию на кнопку происходит её вывод через uart.

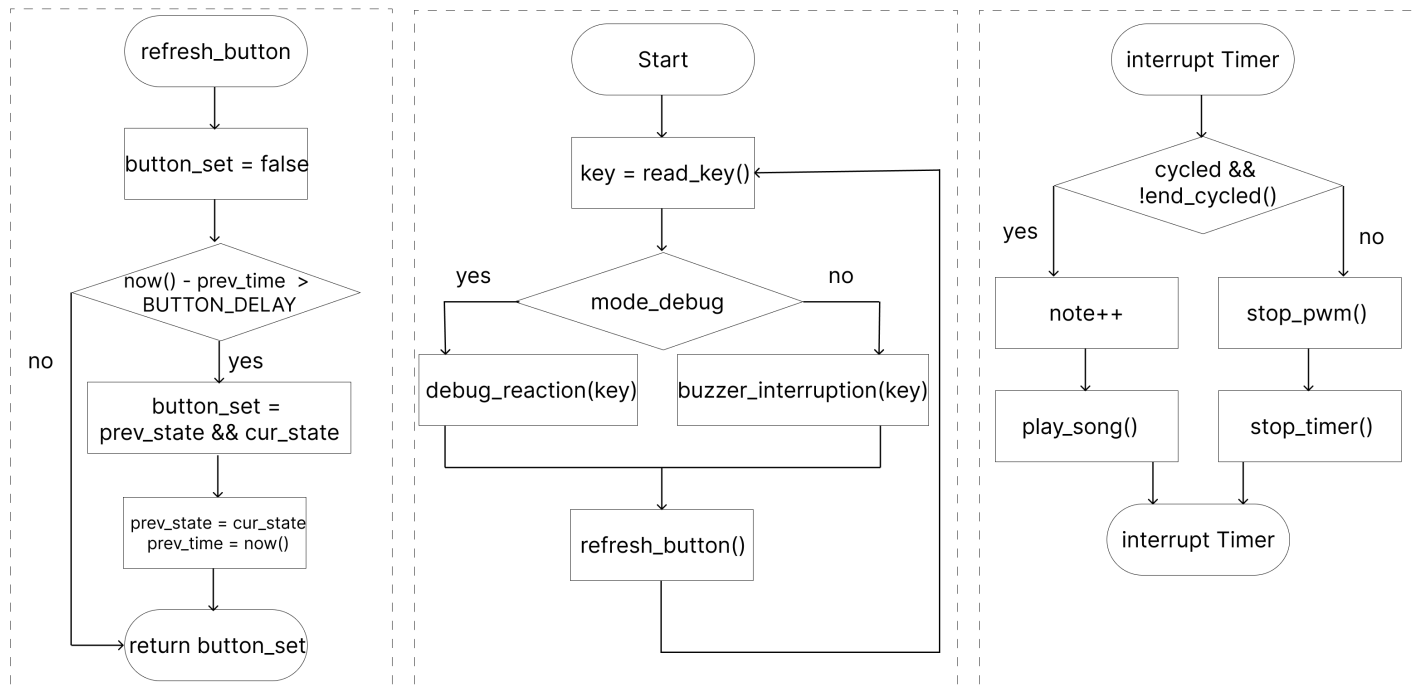
Считывание состояния кнопок происходит с учетом их дребезга.

Если установлен прикладной режим, то по нажатию кнопок происходит запуск соответствующего действия из 3й лабораторной работы. Нумерация клавиатуры начинается с нижнего правого угла. Номера кнопок и их действия:

- 1 - уменьшение продолжительности проигрыша звука
- 2 - проиграть все ноты
- 3 - уменьшить октаву

- 4 - увеличить продолжительности проигрыша звука
- 5 - проиграть текущую ноту
- 6 - увеличить октаву

Блок-Схема алгоритма (рисунок 1)



## Исходный код

```

while (1) {
    if (!check_button()) {
        if (ready) {
            mode_info(&huart6);
            mode = !mode;
            ready = 0;
        }
    } else {
        ready = 1;
    }
    uint8_t key = get_keyboard(&hi2c1);
    if (mode)
        keyboard_test(key);
    else
        keyboard_application(key);

    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
}
  
```

-----

## Основной цикл в функции main

```
HAL_StatusTypeDef Set_Keyboard(I2C_HandleTypeDef *hi2c) {
    HAL_StatusTypeDef ret = HAL_OK;
    uint8_t buf = 0;

    ret |= PCA9538_Write_Register(hi2c, KBRD_ADDR, POLARITY_INVERSION, &buf);
    ret |= PCA9538_Write_Register(hi2c, KBRD_ADDR, OUTPUT_PORT, &buf);

    return ret;
}
```

```
uint8_t Check_Row(I2C_HandleTypeDef *hi2c, uint8_t Nrow) {
    uint8_t Nkey = 0x00;
    uint8_t buf = Nrow;
    uint8_t kbd_in;

    Set_Keyboard(hi2c);

    PCA9538_Write_Register(hi2c, KBRD_ADDR, CONFIG, &buf);
    HAL_Delay(1);
    PCA9538_Read_Inputs(hi2c, KBRD_ADDR, &buf);

    kbd_in = buf & 0x70;
    Nkey = 0;
    if (kbd_in != 0x70) {
        if (!(kbd_in & 0x10)) {
            Nkey |= 0x04;
        }
        if (!(kbd_in & 0x20)) {
            Nkey |= 0x02;
        }
        if (!(kbd_in & 0x40)) {
            Nkey |= 0x01;
        }
    } else
        Nkey = 0x00;

    return Nkey;
}
```

```
uint8_t get_key(I2C_HandleTypeDef *hi2c) {
    uint8_t Row[4] = {ROW4, ROW3, ROW2, ROW1};
```

```

multi_click = 0;
uint8_t pressed_button = 0;

uint8_t key;
for (int i = 0; i < 4; i++) {
    key = Check_Row(hi2c, Row[i]);
    if ((pressed_button && key) || ((key - 1) & key)) multi_click = 1;
    if (key == 0x01) {
        // right
        pressed_button = i * 3 + 1;
    } else if (key == 0x02) {
        // central
        pressed_button = i * 3 + 2;
    } else if (key == 0x04) {
        // left
        pressed_button = i * 3 + 3;
    }
}
return pressed_button;
}

```

---

#### Функции считывания состояния клавиш

```

void keyboard_test(uint8_t key) {
    if (key == 0) {
        if (multi_clicked() && !prev_multi) {
            prev_multi = 1;
            UART_Transmit(&huart6, (uint8_t*)"Warning: Few keys pressed!\n\r");
        }
        return;
    }
    prev_multi = 0;
    char s[64];
    sprintf(s, "Number of pressed button is %d \n\r", key);
    UART_Transmit(&huart6, (uint8_t*)s);
}

```

---

#### Функция вывода текущей нажатой клавиши

```

void keyboard_application(uint8_t key) {
    switch (key) {
        case 12:
        case 11:
        case 10:

```

```

case 9:
case 8:
case 7:
case 5:
    play_song(notes[key - 1]);
    send_long_info(&huart6, notes[key - 1], get_octave(), get_duration());
    break;
case 2:
    play_all_song();
    send_info(&huart6, get_octave(), get_duration());
    break;
case 6:
    increase_octave();
    send_info(&huart6, get_octave(), get_duration());
    break;
case 3:
    decrease_octave();
    send_info(&huart6, get_octave(), get_duration());
    break;
case 4:
    increase_duration();
    send_info(&huart6, get_octave(), get_duration());
    break;
case 1:
    decrease_duration();
    send_info(&huart6, get_octave(), get_duration());
    break;
default:
    break;
}
}

```

---

### Функция работы прикладного режима

### Выводы

- Получили базовые знания об интерфейсе I2C и особенностях передачи данных по данному интерфейсу.
- Получили базовые знания об устройстве и принципах работы контроллера интерфейса I2C в микроконтроллерах и получили навыки его программирования.