# Университет ИТМО

Проектирование вычислительных систем

**Лабораторная работа №4**

Интерфейс $I^2C$ и матричная клавиатура

ФИО студентов:
Готовко Алексей Владимирович
Руденко Илья Александрович
Вариант: 5
Направление подготовки: 09.03.04 (СППО)
Учебная группа: P34101
ФИО преподавателя: Пинкевич Василий Юрьевич

Санкт-Петербург
2024г.

# Содержание

# 1 Цели работы

1. Получить базовые знания об интерфейсе $I^2C$ и особенностях передачи данных по данному интерфейсу.

2. Получить базовые знания об устройстве и принципах работы контроллера интерфейса $I^2C$ в микроконтроллерах и получить навыки его программирования.

# 2 Вариант задания

Задания аналогичны вариантам лабораторной работы №3, за исключением того, что ввод символов должен выполняться не с клавиатуры через UART, а с помощью клавиатуры стенда. Выбор кнопок клавиатуры стенда, играющих роль кнопок клавиатуры компьютера, должен выполняться по усмотрению исполнителей. В отчете необходимо привести описание функций кнопок в реализованной программе.

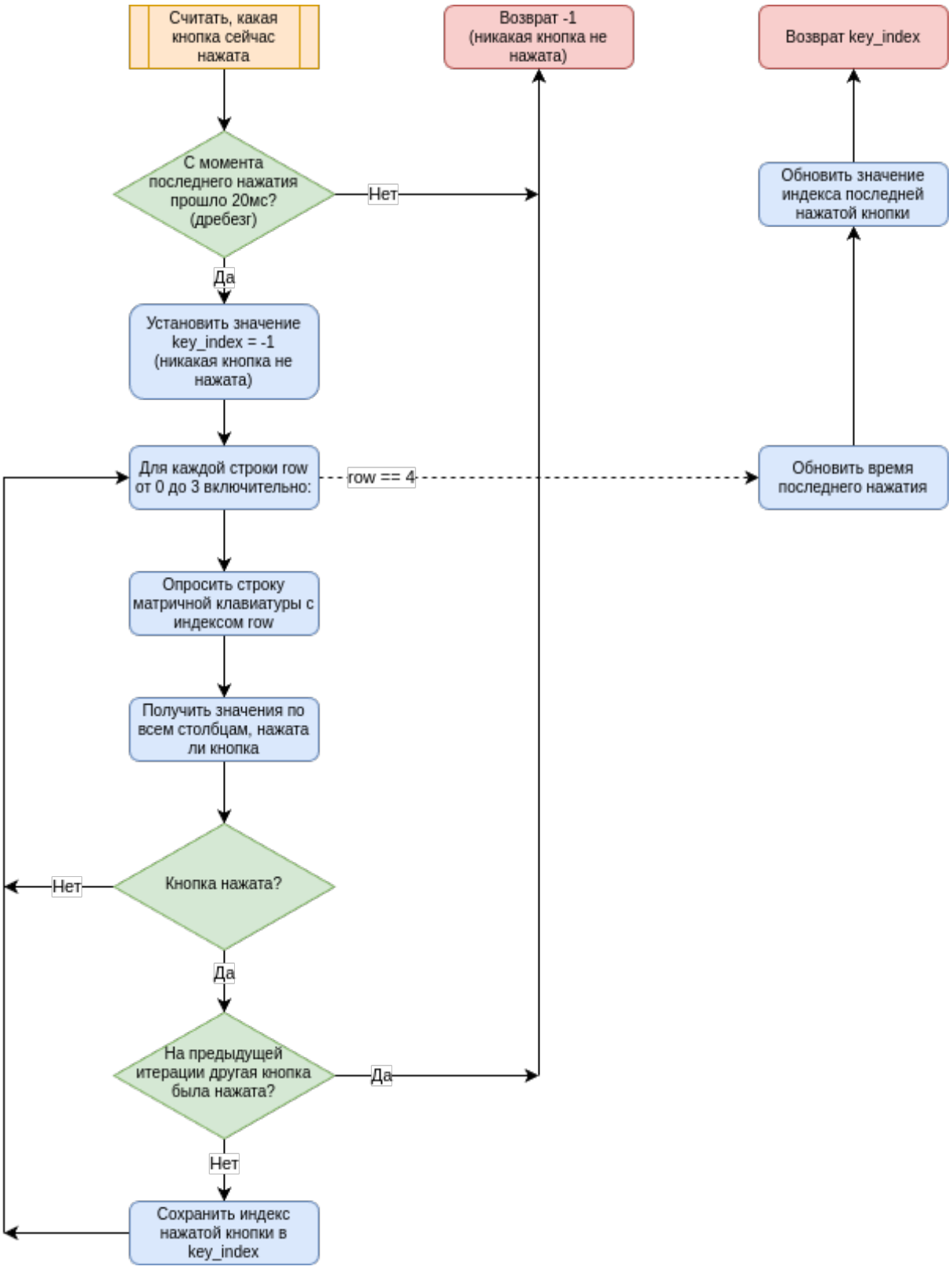
# 3 Описание программы

Пронумеруем все кнопки матричной клавиатуры от 1 до 12, начиная с верхнего левого угла. Тогда кнопки 1-9 отвечают за ввод чисел от 1 до 9, кнопка 10 – за переключение информационного и игрового (прикладного) режимов работы программы, кнопка 11 – за переключение сложности игры, кнопка 12 – за начало/конец игры.

Далее опустим объяснение всей логики, реализованной и описанной в предыдущей лабораторной работе.

Для выяснения того, какая кнопка нажата, мы будем поочередно опрашивать все строки матричной клавиатуры и на каждом опросе выяснять, нажата ли на текущей строке какая-либо кнопка. Если нажато несколько кнопок, то считаем, что не нажата ни одна. Если нажата только одна кнопка, то ее индекс мы и возвращаем.

# 4 Блок-схема алгоритма



# 5 Код программы

Вспомогательные определения:

```
#define INPUT_PORT_REG          (0x00)
#define OUTPUT_PORT_REG         (0x01)
#define POLARITY_INV_REG        (0x02)
#define CONFIG_REG              (0x03)
#define KEYPAD_ADDRESS          (0xE2)
#define KEYPAD_WRITE_ADDRESS    ((KEYPAD_ADDRESS) & ~1)
#define KEYPAD_READ_ADDRESS     ((KEYPAD_ADDRESS) | 1)
#define COLUMN_MASK             0x7
#define CONTACT_BOUNCE_MS       20

uint32_t last_pressed_time;
int last_pressed_key_index = -2;

char test_mode_msg[] = "\n\rCurrent mode: keypad test\n\r";
char main_mode_msg[] = "\n\rCurrent mode: main\n\r";
```

Функции-драйверы:

```
HAL_StatusTypeDef reset_keypad(void) {
    uint8_t buf = 0;
    HAL_StatusTypeDef status = HAL_I2C_Mem_Write(&hi2c1, KEYPAD_WRITE_ADDRESS,
    ↪  POLARITY_INV_REG, 1, &buf, 1, 100);
    if (status != HAL_OK)
        return status;
    status = HAL_I2C_Mem_Write(&hi2c1, KEYPAD_WRITE_ADDRESS, OUTPUT_PORT_REG, 1, &buf, 1,
    ↪  100);
    return status;
}

int keypad_read_key_index(void) {
    uint32_t cur_time = HAL_GetTick();
    if (cur_time - last_pressed_time < CONTACT_BOUNCE_MS) return -1;

    int key_index = -1;
    uint8_t buf;
    uint16_t pressed_column;

    for (int row = 0; row < 4; row++) {
        buf = ~((uint8_t) (1 << row));
        pressed_column = 0x00;

        reset_keypad();

        HAL_I2C_Mem_Write(&hi2c1, KEYPAD_WRITE_ADDRESS, CONFIG_REG, 1, &buf, 1, 100);
        HAL_Delay(10);
        HAL_I2C_Mem_Read(&hi2c1, KEYPAD_READ_ADDRESS, INPUT_PORT_REG, 1, &buf, 1, 100);

        pressed_column = (~(buf >> 4)) & COLUMN_MASK;
        switch (pressed_column) {
            case 0x1:
                if (key_index != -1) return -1;
```

```
                key_index = row * 3;
                break;
            case 0x2:
                if (key_index != -1) return -1;
                key_index = (row * 3) + 1;
                break;
            case 0x4:
                if (key_index != -1) return -1;
                key_index = (row * 3) + 2;
                break;
        }
    }

    if (key_index != -1) last_pressed_time = cur_time;
    if (key_index == last_pressed_key_index)
        return -1;
    last_pressed_key_index = key_index;
    return key_index;
}
```

Основной цикл программы:

```
last_pressed_time = 0;
bool button_pressed = false;
bool is_test_mode = true;

uart_write(is_test_mode ? test_mode_msg : main_mode_msg);

int key_index;
while (1)
{
    if (is_button_pressed() && button_pressed) continue;

    button_pressed = is_button_pressed();

    if (button_pressed) {
        is_test_mode = !is_test_mode;
        uart_write(is_test_mode ? test_mode_msg : main_mode_msg);
        continue;
    }

    key_index = keypad_read_key_index();

    if (key_index == -1)
        continue;

    if (is_test_mode) {
        char buf[4];
        sprintf(buf, "%d\n\r", key_index);
        uart_write(buf);
        continue;
    }

    if (key_index == 11) {
        switch_game_mode();
```

```
34          continue;
35      }
36
37      if (0 <= key_index && key_index <= 8) {
38          if (game_mode == INFO) {
39              uint8_t id = key_index;
40              play_note(&notes[id]);
41              uart_write_note_info(&notes[id]);
42
43              set_timer_ms(1000);
44          } else {
45              cur_guess = key_index + 1;
46              char buf[4];
47              sprintf(buf, "%d", cur_guess);
48              uart_write(buf);
49          }
50          continue;
51      }
52
53      if (key_index == 9 && game_mode == INFO) {
54          switch_note_mode();
55          continue;
56      }
57
58      if (key_index == 10 && game_mode == INFO) {
59          switch_difficulty();
60          continue;
61      }
62  }
```

# 6 Вывод

В результате выполнения лабораторной работы были получены базовые знания об интерфейсе $I^2C$ и особенностях передачи данных по данному интерфейсу, а также об устройстве и принципах работы контроллера интерфейса $I^2C$ в микроконтроллерах; были получены навыки его программирования.