

Лабораторна робота №5

Мембранні клавіатури

Мета: дослідити способи взаємодії з мембранними клавіатурами.

Теоретичні відомості

Мембранна клавіатура — це тип клавіатурного пристрою введення, який використовує тонкі, гнучкі шари (мембрани) для розміщення контактів, що замикаються при натисканні. Такий тип клавіатур широко використовується в електроніці завдяки своїй простоті, низькій вартості виробництва, пилонепроникності та довговічності.

Мембранна клавіатура складається з кількох основних шарів:

1. **Верхній шар (графічна панель)** — це зовнішній шар, на якому надруковані символи клавіш. Він виготовляється з поліестеру або іншого гнучкого матеріалу і виконує декоративну та інформативну функцію.
2. **Верхня мембрана з провідними доріжками** — під графічною панеллю розташовується шар з нанесеними провідниками, що утворюють одну частину контактної пари. У місцях клавіш є гнучкі куполи або виїмки, які при натисканні замикають контакти.
3. **Проміжний ізоляційний шар** — тонка пластина з отворами у місцях клавіш. Вона виконує функцію розділення між провідними шарами, запобігаючи випадковому замиканню без натискання.
4. **Нижня мембрана з провідними доріжками** — цей шар містить другу частину контактної пари. Коли клавішу натискають, верхня провідна доріжка через отвір в ізоляційному шарі замикається з нижньою доріжкою, утворюючи електричний контакт.
5. **Основний несучий шар (зазвичай ПВХ або жорсткий поліестер)** — слугує жорсткою основою всієї клавіатури і може містити друковану плату або конектори для підключення.

Принцип дії мембранної клавіатури базується на механізмі замикання електричного кола. При натисканні клавіші, верхній провідний шар притискається до нижнього через отвір у ізоляційному шарі. У результаті відбувається електричне замикання, яке реєструється мікроконтролером або іншим пристроєм, що обробляє натискання.

Коли користувач відпускає клавішу, еластичність матеріалу мембран повертає її у вихідне положення, розмикаючи контакт.

Типи мембранних клавіатур

Плоскі (без зворотного зв'язку) — клавіші не мають тактильного відгуку. Такі клавіатури використовуються у приладах, де не потрібна часта взаємодія.

1. **З тактильним відгуком (із куполами)** — містять еластичні гумові куполи, які створюють відчутний "клік" при натисканні. Це поширений варіант для побутової електроніки та комп'ютерних клавіатур.
2. **З підсвічуванням** — мають вбудоване світлодіодне або EL-підсвічування для роботи в умовах низької освітленості.

До переваг належать

1. **Низька вартість** — виробництво мембранних клавіатур є дешевшим, ніж механічних.
2. **Тонкий профіль** — за рахунок гнучких матеріалів можливо виготовляти клавіатури товщиною менше 1 мм.

3. **Герметичність** — захищені від пилу, вологи та інших забруднень, що дозволяє використовувати їх у складних умовах.
4. **Низький рівень шуму** — клавіші працюють майже безшумно.
5. **Довговічність** — не мають металевих або механічних рухомих частин, які можуть зношуватись.

Недоліки

- **Обмежений тактильний відгук** — користувачу складно відчути момент активації клавіші, особливо у плоских варіантах.
- **Менша довговічність у порівнянні з механічними клавіатурами** — контактні доріжки можуть стиратись з часом.
- **Потреба в повному натисканні** — щоб спрацював контакт, потрібно натискати клавішу повністю донизу, що уповільнює набір.

Мембранні клавіатури широко застосовуються в наступних галузях:

- **Побутова техніка** — мікрохвильові печі, пральні машини, кондиціонери.
- **Медичне обладнання** — апарати, де важлива стерильність і легкість дезінфекції.
- **Промислова автоматика** — панелі керування, де потрібна пилонепроникність.
- **Комп'ютерна периферія** — дешеві моделі клавіатур для ПК.
- **Мікроконтролерні системи** — використання у проєктах Arduino, STM32, AVR тощо.

Мембранна клавіатура зазвичай реалізується у вигляді **матричної клавіатури**, де кнопки розташовані на перетині рядків і стовпців. Наприклад, у клавіатурі 4×3 використовується 4 вхідних і 3 вихідних лінії, що дозволяє зчитувати до 12 клавіш ($4 \times 3 = 12$) за допомогою опитування.

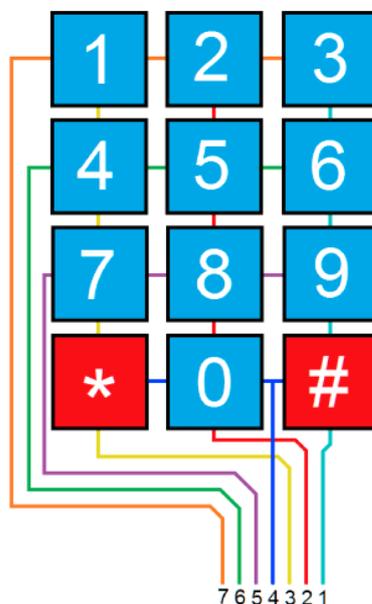


Рис.5.1 – Схема мембрани.

Опитування здійснюється шляхом поетапного встановлення низького рівня на одну колонку, після чого зчитується стан кожного рядка. При виявленні замкнутого контакту визначаються координати клавіші за номером рядка і колонки.

У мікроконтролерних системах зчитування натиснень реалізується через:

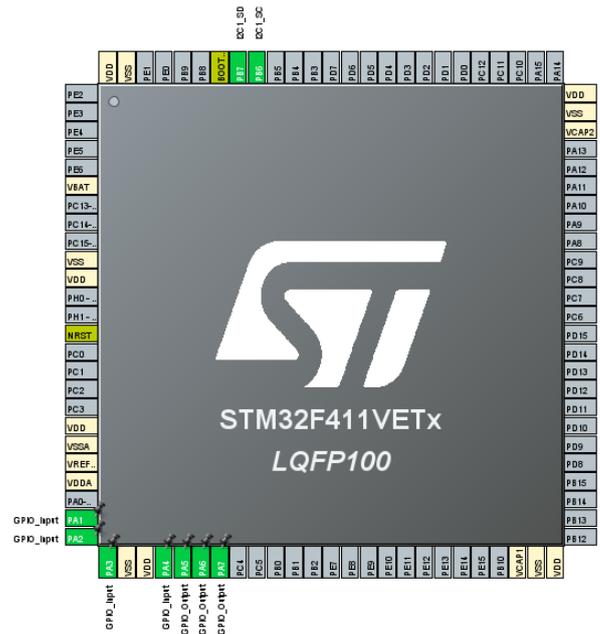
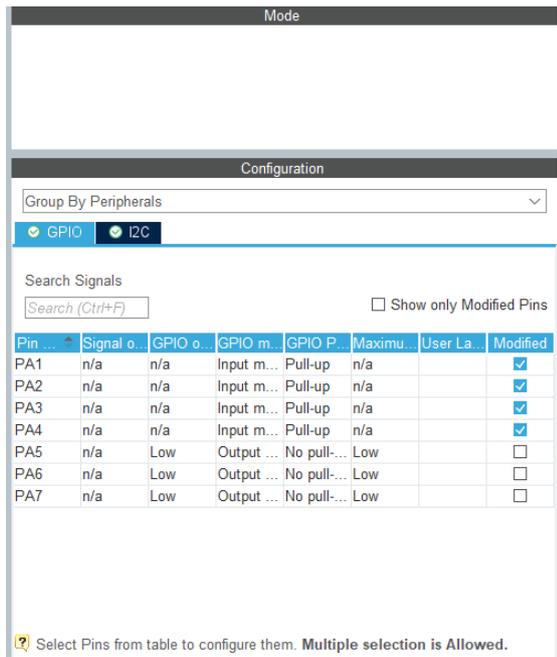
- Циклічне опитування рядків і колонок
- Програмне усунення дребізу контактів
- Затримку після натискання для стабільності
- Визначення символу по матриці клавіш

У багатьох мікроконтролерах реалізовані спеціальні бібліотеки або драйвери для полегшення роботи з мембранними клавіатурами.

Хід виконання роботи

5.1. Налаштувати файл конфігурації контролера:

5.1.1. Визначити доступні піни відповідно до документації.



5.1.2. Налаштувати ці піни PA1-4 режим input mode з підтягуванням Pull-up, а PA5-7 в режим output mode.

5.1.3. Здійснити підключення мембранної клавіатури до мікроконтролера.

5.1.4. Підключити LCD дисплей за протоколом I2C (як в попередній роботі)

5.2. Перевірити правильність підключення та налаштування за допомогою коду:

5.2.1. Ініціалізація масиву символів які є на клавіатурі.

```
/* USER CODE BEGIN 0 */
```

```
char keypad[4][3] = {
    {'1','2','3'},
    {'4','5','6'},
    {'7','8','9'},
    {'*','0','#'}
};
```

```
uint16_t row_pins[4] = { GPIO_PIN_1, GPIO_PIN_2, GPIO_PIN_3, GPIO_PIN_4 };
uint16_t col_pins[3] = { GPIO_PIN_5, GPIO_PIN_6, GPIO_PIN_7 };
```

```
/* USER CODE END 0 */
```

5.2.2. Імплементувати функцію читання з клавіатури.

```
char read_keypad() {
    for (int col = 0; col < 3; col++) {
        // Усі колонки високі
```

```

for (int i = 0; i < 3; i++) {
    HAL_GPIO_WritePin(GPIOA, col_pins[i], GPIO_PIN_SET);
}

// Активуємо одну колонку
HAL_GPIO_WritePin(GPIOA, col_pins[col], GPIO_PIN_RESET);

// Читаємо рядки
for (int row = 0; row < 4; row++) {
    if (HAL_GPIO_ReadPin(GPIOA, row_pins[row]) == GPIO_PIN_RESET) {

        return keypad[row][col];
    }
}
return 0;
}

```

5.2.3. В головній функції main реалізувати вивід на LCD дисплей відповідного символу після натискання:

```

int main(void)
{
    HAL_Init();
    SystemClock_Config();
    MX_GPIO_Init();
    MX_I2C1_Init();

    HD44780_Init(2);
    HD44780_Clear();
    HD44780_SetCursor(0, 0);
    HD44780_PrintStr("Press key:");

    while (1)
    {
        char key = read_keypad();
        if (key) {
            HD44780_Clear();
            HD44780_PrintStr("You pressed:");
            HD44780_SetCursor(0, 1);
            char key_str[2] = { key, '\0' };
            HD44780_PrintStr(key_str);
        }
    }
}

```

Індивідуальні завдання

1. Реалізувати систему «Цифровий замок». При першому запуску буде запропоновано ввести 4х значний код який буде кодом безпеки. При введенні код відображається. Збереження відбувається за допомогою клавіші «#». Після чого система переходить в режим блокування. І на екрані висвічується «Access denied». Знизу під статусом горять «_ _ _ _». Користувач може ввести з клавіатури цифри окрім «*», «#». Після введення цифр «_ _ _ _» на LCD дисплеї замінюється на «*». Як усі цифри введено користувач може натиснути «#» та його введений код буде перевірений на правильність. Якщо код правильний то на екрані буде «Access allowed», в іншому випадку надпис «Access denied» лишається і введений код скидується до «_ _ _ _». Натискання «*» видалить останню введену цифру. Код доступу можна змінити лише тоді, коли попередній був введений правильно, тобто на LCD дисплеї загорілось «Access allowed», після утримання «#» 5 секунд повинно висвітитись надпис

«New code» і користувач введе новий код. При натисканні «*» система повернеться в блокуючий режим.

Зміст звіту

1. Тема та мета лабораторної роботи.
2. Скріншот конфігурації мікропроцесора (файл .ios);
3. Код файлу main.c функції main для кожного пункту індивідуального плану;
4. Висновки.

Контрольні запитання

1. Які переваги використання мембранної клавіатури в мікроконтролерних системах?
2. Яка роль «рядків» і «стовпців» у клавіатурі?
3. Як визначити, яка саме кнопка натиснута?
4. Як зменшити кількість портів МК для підключення клавіатури?
5. Що таке мембранна клавіатура?

```

#include "main.h"
#include "liquidcrystal_i2c.h"
#include <string.h>
#include <stdbool.h>

char keypad[4][3] = {
    {'1', '2', '3'},
    {'4', '5', '6'},
    {'7', '8', '9'},
    {'*', '0', '#'}
};

uint16_t row_pins[4] = { GPIO_PIN_1, GPIO_PIN_2, GPIO_PIN_3, GPIO_PIN_4 };
uint16_t col_pins[3] = { GPIO_PIN_5, GPIO_PIN_6, GPIO_PIN_7 };

char secure_code[5] = "";
char entered_code[5] = "";
int input_index = 0;
bool access_granted = false;
bool code_set = false;

char read_keypad() {
    for (int col = 0; col < 3; col++) {
        for (int i = 0; i < 3; i++) HAL_GPIO_WritePin(GPIOA, col_pins[i],
GPIO_PIN_SET);
        HAL_GPIO_WritePin(GPIOA, col_pins[col], GPIO_PIN_RESET);

        for (int row = 0; row < 4; row++) {
            if (HAL_GPIO_ReadPin(GPIOA, row_pins[row]) == GPIO_PIN_RESET) {
                return keypad[row][col];
            }
        }
    }
    return 0;
}

void wait_keypad_release() {
    while (read_keypad() != 0) {
        HAL_Delay(10);
    }
}

uint32_t read_keypad_hold_time(char target) {
    uint32_t start = HAL_GetTick();
    while (read_keypad() == target) {
        HAL_Delay(10);
    }
    return HAL_GetTick() - start;
}

void display_masked_input() {
    --- реалізувати функцію введення символів
}

void display_input(const char *code, int length) {
    HD44780_SetCursor(0, 1);
    for (int i = 0; i < 4; i++) {
        if (i < length) {
            char c[2] = { code[i], '\0' };
            HD44780_PrintStr(c);
        } else {

```

```

        HD44780_PrintStr("_");
    }
}

void lock_mode() {
    --- реалізувати інформування невдалого введення паролю
}

void unlock_mode() {
    // --- реалізувати інформування успішного входу
}

void reset_input(char *buffer) {
    memset(buffer, 0, 5);
    input_index = 0;
}

int main(void) {
    HAL_Init();
    SystemClock_Config();
    MX_GPIO_Init();
    MX_I2C1_Init();
    HD44780_Init(2);

    // --- Встановлення початкового коду ---
    HD44780_Clear();
    HD44780_SetCursor(0, 0);
    HD44780_PrintStr("Set 4-digit code:");
    display_masked_input();

    while (!code_set) {
        char key = read_keypad();
        if (key && key != '#' && key != '*') {
            if (input_index < 4) {
                secure_code[input_index++] = key;
                display_input(secure_code, input_index);
            }
        } else if (key == '#') {
            if (input_index == 4) {
                secure_code[4] = '\\0';
                code_set = true;
                input_index = 0;
                memset(entered_code, 0, sizeof(entered_code));
                HAL_Delay(500);
                lock_mode();
            }
        }
        wait_keypad_release();
    }

    // --- Основний цикл ---
    while (1) {
        char key = read_keypad();
        if (!key) continue;

        if (key >= '0' && key <= '9' && input_index < 4) {
            entered_code[input_index++] = key;
            lock_mode();
        } else
            if (key == '*') {
                if(access_granted){
                    input_index = 0;
                    access_granted = false;
                }
            }
    }
}

```

