**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"**

**Інститут комп’ютерних технологій, автоматики та метрології**

**Кафедра "Комп'ютеризовані системи автоматики"**

# **Звіт до виконання лабораторної роботи №1 з дисципліни “Мікроконтролери, частина 1” за темою Реалізація алгоритму засвічування 3-LEDів**

******

**Виконав:**

студент групи ІР-21

Мамчур М. А.

**Прийняв:**

Ванюк В. І.

Львів 2025

**Мета роботи:** Метою лабораторної роботи є ознайомлення з основами програмування мікроконтролерів на базі ESP8266

**Завдання:**

1. Згідно варіанту завдання (таблиця 1) скласти у пакеті симуляції

схему на основі МК Esp8266. Написати програму мовою C++ в

Arduino IDE для реалізації вказаного алгоритму.

2. Підключити кнопку і діоди до контролера WemosD1 mini згідно із

схемою. (пін D4 підведений до builtin\_led, тому підключення

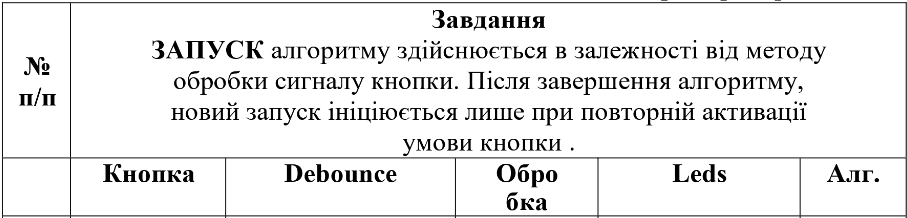
додаткового діоду до нього не обов’язкове).

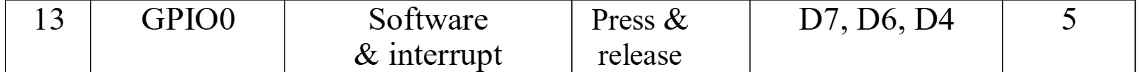
3. Залити програму в памʼять контролера.

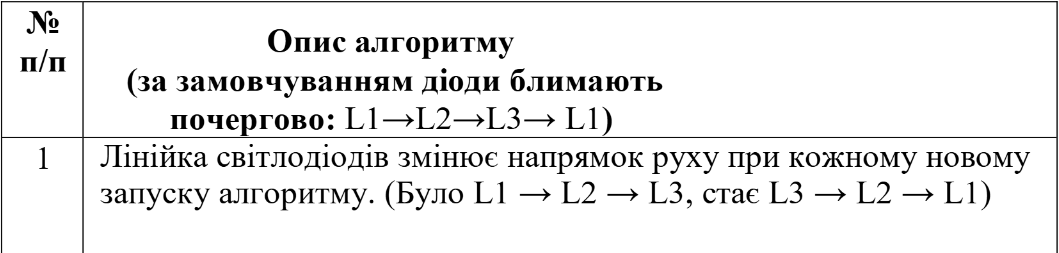
4. Перевірити на правильність виконання.

5. Реалізувати веб сторінку з кнопкою виклику алгоритма.

6. Отримані результати представити викладачу







**Код:**

**main.cpp:**

**#include <ESP8266WiFi.h>**

**#include <ESP8266WebServer.h>**

**#define LED1 13 // D7 (GPIO13)**

**#define LED2 12 // D6 (GPIO12)**

**#define LED3 2 // D4 (GPIO2)**

**#define BUTTON\_PIN 0 // GPIO0**

**bool reverseSequence = false;**

**uint32\_t lastInterruptTime = 0;**

**const uint8\_t debounceDelay = 50;**

**const char\* ssid = "Ronka 13";**

**const char\* password = "08042001";**

**ESP8266WebServer server(80);**

**unsigned long lastMillis = 0;**

**const uint16\_t delayTime = 500;**

**bool isLedOn = false;**

**unsigned long ledTimer = 0;**

**void ICACHE\_RAM\_ATTR handleButton() {**

**static bool buttonPressed = false;**

**uint32\_t interruptTime = millis();**

**if (interruptTime - lastInterruptTime > debounceDelay) {**

**if (!buttonPressed) {**

**buttonPressed = true;**

**} else {**

**reverseSequence = !reverseSequence;**

**buttonPressed = false;**

**}**

**lastInterruptTime = interruptTime;**

**}**

**}**

**void setup() {**

**Serial.begin(115200);**

**WiFi.begin(ssid, password);**

**uint32\_t maxTimer = millis();**

**while (WiFi.status() != WL\_CONNECTED) {**

**delay(1000);**

**Serial.println("Connecting to WiFi...");**

**if (millis() - maxTimer >= 60000) {**

**Serial.println("WiFi connection timeout. Restarting...");**

**ESP.restart();**

**}**

**}**

**Serial.println("Connected to WiFi");**

**Serial.println(WiFi.localIP());**

**pinMode(LED1, OUTPUT);**

**pinMode(LED2, OUTPUT);**

**pinMode(LED3, OUTPUT);**

**pinMode(BUTTON\_PIN, INPUT\_PULLUP);**

**attachInterrupt(digitalPinToInterrupt(BUTTON\_PIN), handleButton, CHANGE);**

**server.on("/", HTTP\_GET, []() {**

**String html = "<html><body><h1>ESP8266 KNOPKA</h1><br><br>";**

**html += "<button onclick=\"window.location.href='/toggle'\">";**

**html += reverseSequence ? "Reverse" : "Default";**

**html += "</button></body></html>";**

**server.send(200, "text/html", html);**

**});**

**server.on("/toggle", HTTP\_GET, []() {**

**reverseSequence = !reverseSequence;**

**server.sendHeader("Location", "/");**

**server.send(303);**

**});**

**server.begin();**

**}**

**void loop() {**

**server.handleClient();**

**if (millis() - lastMillis >= delayTime) {**

**lastMillis = millis();**

**lightSequence(reverseSequence); // Викликаємо правильну функцію**

**}**

**handleLedTiming();**

**}**

**void lightSequence(bool reverse) {**

**static uint8\_t currentStep = 0; // Використовуємо 1 байт пам’яті**

**const uint8\_t sequenceForward[] = { LED1, LED2, LED3 };**

**const uint8\_t sequenceReverse[] = { LED3, LED2, LED1, LED2 };**

**if (reverse) {**

**startLight(sequenceReverse[currentStep]);**

**currentStep = (currentStep + 1) % 4;**

**} else {**

**startLight(sequenceForward[currentStep]);**

**currentStep = (currentStep + 1) % 3;**

**}**

**}**

**void startLight(int led) {**

**digitalWrite(led, HIGH);**

**isLedOn = true;**

**ledTimer = millis();**

**}**

**void handleLedTiming() {**

**if (isLedOn && millis() - ledTimer >= delayTime / 2) {**

**digitalWrite(LED1, LOW);**

**digitalWrite(LED2, LOW);**

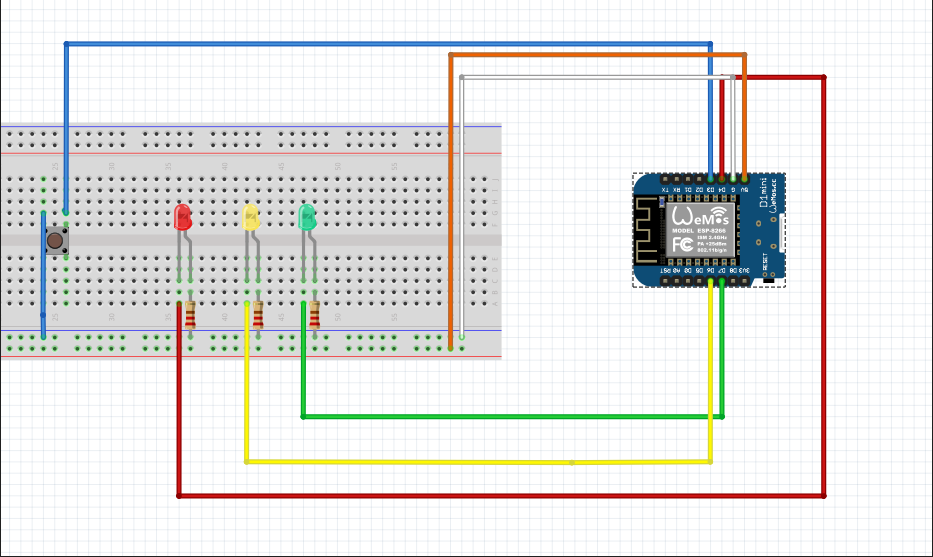
**digitalWrite(LED3, LOW);**

**isLedOn = false;**

**}**

**}**

**Схема:**

****

**Висновок:** У ході виконання лабораторної роботи було розглянуто основи роботи з мікроконтролером ESP8266. Було розроблено та запрограмовано алгоритм засвічування трьох світлодіодів відповідно до варіанту завдання, що дало змогу закріпити навички роботи з цифровими входами та виходами.