Лабораторна робота №2

Тема: Використання ESP32 в онлайн-симуляторі Wokwi з програмуванням на C++ та MicroPython

Мета роботи:

- Ознайомитися з підключенням ESP32 до Wi-Fi та отриманням часу через NTP;
- Навчитися працювати з LCD-дисплеєм через I2C;
- Порівняти реалізацію задачі мовами C++ (Arduino) та Python (MicroPython).

Необхідні інструменти:

- Онлайн-симулятор: https://wokwi.com
- Обліковий запис Google для збереження проектів.

Короткі теоретичні відомості:

Мікроконтролер ESP32 є потужним рішенням для IoT. Він підтримує Wi-Fi та Bluetooth, має значну обчислювальну потужність, і дозволяє програмування як через Arduino IDE (C++), так і через MicroPython. Wokwi — це онлайн-симулятор, який дозволяє моделювати роботу ESP32 без фізичного обладнання.

Хід виконання лабораторної роботи

Завдання 1: C++ (Arduino) реалізація

1. Створіть новий проект на Wokwi та оберіть мікроконтролер ESP32.

2. Додайте до схеми LCD дисплей з інтерфейсом I2C (адреса 0x27). Створіть схему представлену на рис. 2.1.

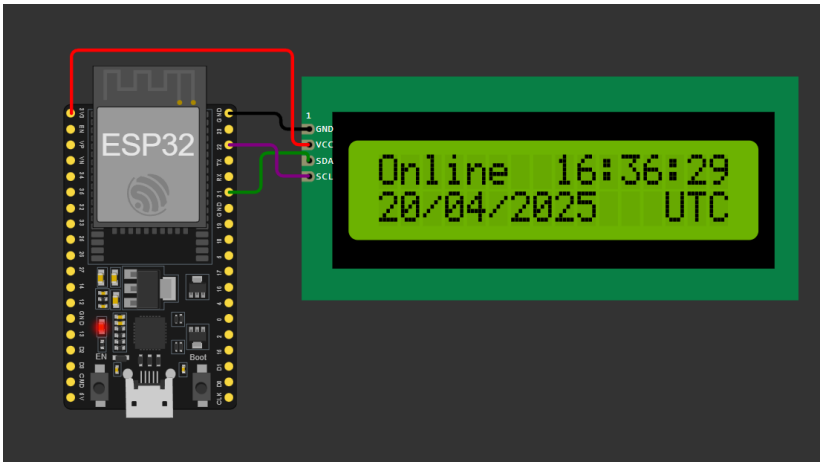

Рис. 2.1 Схема підключення екрана

3. Вставте в редактор коду наступний код C++:

```
#include <WiFi.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C LCD = LiquidCrystal_I2C(0x27, 16, 2);

#define NTP_SERVER     "pool.ntp.org"
#define UTC_OFFSET     0
#define UTC_OFFSET_DST 0

void spinner() {
  static int8_t counter = 0;
  const char* glyphs = "\xa1\xa5\xdb";
  LCD.setCursor(15, 1);
  LCD.print(glyphs[counter++]);
  if (counter == strlen(glyphs)) {
    counter = 0;
  }
}

void printLocalTime() {
  struct tm timeinfo;
  if (!getLocalTime(&timeinfo)) {
    LCD.setCursor(0, 1);
```

```
    LCD.println("Connection Err");
    return;
  }
  LCD.setCursor(8, 0);
  LCD.println(&timeinfo, "%H:%M:%S");
  LCD.setCursor(0, 1);
  LCD.println(&timeinfo, "%d/%m/%Y   %Z");
}

void setup() {
  Serial.begin(115200);
  LCD.init();
  LCD.backlight();
  LCD.setCursor(0, 0);
  LCD.print("Connecting to ");
  LCD.setCursor(0, 1);
  LCD.print("WiFi ");

  WiFi.begin("Wokwi-GUEST", "", 6);
  while (WiFi.status() != WL_CONNECTED) {
    delay(250);
    spinner();
  }

  Serial.println("");
  Serial.println("WiFi connected");
  Serial.print("IP address: ");
  Serial.println(WiFi.localIP());

  LCD.clear();
  LCD.setCursor(0, 0);
  LCD.println("Online");
  LCD.setCursor(0, 1);
  LCD.println("Updating time...");

  configTime(UTC_OFFSET, UTC_OFFSET_DST,
NTP_SERVER);
}

void loop() {
```

```
  printLocalTime();
  delay(250);
}
```

Зверніть увагу на табуляцію коду програми! Виправіть помилки, якщо вони виникнуть.

Завдання 2: MicroPython реалізація
1. Створіть окремий проект на Wokwi з MicroPython ESP32.

2. Використайте бібліотеку `ntptime` та `I2C_LCD` (можна підключити як окремі модулі).

3. Реалізуйте оновлення часу з NTP-серверу і вивід на LCD.

Необхідні бібліотеки
Необхідно додати в проєкт файли:

- - lcd_api.py
- - i2c_lcd.py

Їх можна завантажити з репозиторію:
https://github.com/dhylands/python_lcd або додати через вкладку "Libraries" у Wokwi.

Код основного файлу main.py

```
import network
import ntptime
import time
from machine import I2C, Pin
from lcd_api import LcdApi
from i2c_lcd import I2cLcd

# Налаштування I2C для LCD
i2c = I2C(0, scl=Pin(22), sda=Pin(21), freq=400000)
lcd = I2cLcd(i2c, 0x27, 2, 16)
```

```
# Підключення до WiFi
lcd.putstr("Connecting...\nWiFi")
wifi = network.WLAN(network.STA_IF)
wifi.active(True)
wifi.connect('Wokwi-GUEST', '')

while not wifi.isconnected():
    time.sleep(0.2)

lcd.clear()
lcd.putstr("WiFi Connected\nSync time...")

# Синхронізація часу з NTP
try:
    ntptime.settime()
    lcd.clear()
    lcd.putstr("Time updated!")
except:
    lcd.clear()
    lcd.putstr("Time update fail")

# Вивід поточного часу в циклі
while True:
    now = time.localtime()
    lcd.move_to(0, 0)
    lcd.putstr("{:02d}:{:02d}:{:02d}".format(now[3], now[4],
now[5]))
    lcd.move_to(0, 1)
    lcd.putstr("{:02d}/{:02d}/{}".format(now[2], now[1], now[0]))
    time.sleep(1)
```

Очікуваний результат

Після запуску коду на екрані LCD має з’явитися поточний час і дата, що оновлюються щосекунди. Якщо підключення до Wi-Fi або NTP не вдалося, на дисплеї з’явиться повідомлення про помилку.

Контрольні запитання:

1. Як ESP32 отримує час із мережі?
2. Що таке I2C та як підключається дисплей?
3. Які відмінності між C++ (Arduino) і MicroPython для ESP32?
4. Як змінити часовий пояс у коді?
5. Як передати ці дані в хмару або мобільний застосунок?