

**Міністерство освіти і науки України
Луцький національний технічний університет**



ІоТ-ТЕХНОЛОГІЇ ДЛЯ КІБЕРФІЗИЧНИХ СИСТЕМ

Методичні вказівки до лабораторних занять
для здобувачів другого (магістерського) рівня вищої освіти
освітньої програми «Комп'ютерна інженерія»
галузь знань F Інформаційні технології
спеціальності F7 Комп'ютерна інженерія
денної та заочної форм навчання

УДК 004.2
І 75

Рекомендовано до видання вченою радою факультету КІТ ЛНТУ,
протокол № _____ від « ____ » _____ 20_26 року.

Голова вченої ради факультету КІТ _____ Інна КОНДІУС

Електронна копія друкованого видання передана для внесення в репозитарій ЛНТУ
Директор бібліотеки _____ Наталія ПОЛІЩУК

Розглянуто і схвалено на засіданні кафедри комп'ютерної інженерії та безпеки
ЛНТУ, протокол № _____ від « ____ » _____ 20_26 року.

Завідувач кафедри КІБ _____ Тарас ТЕРЛЕЦЬКИЙ

Укладачі: _____ Сергій ГРИНЮК, кандидат технічних наук,
доцент кафедри комп'ютерної інженерії та безпеки ЛНТУ

_____ Людмила КОНКЕВИЧ, асистент
кафедри комп'ютерної інженерії та безпеки ЛНТУ

Рецензент: _____ Андрій ЯЦУК, кандидат технічних наук,
доцент кафедри інженерії програмного забезпечення ЛНТУ

Відповідальний за випуск: _____ Тарас ТЕРЛЕЦЬКИЙ, кандидат
технічних наук, доцент кафедри комп'ютерної інженерії та безпеки ЛНТУ

І 75 **ІоТ-технології для кіберфізичних систем:** методичні вказівки до лабораторних занять для здобувачів другого (магістерського) рівня вищої освіти освітньої програми «Комп'ютерна інженерія» галузь знань F Інформаційні технології спеціальності F7 Комп'ютерна інженерія денної та заочної форм навчання / уклад. С.В. Гринюк, Л.М. Конкевич Луцьк: ЛНТУ, 2026. 48 с.

Методичні вказівки до лабораторних занять з дисципліни «ІоТ-технології для кіберфізичних систем» складені відповідно до діючої програми курсу. Видання містить тематику лабораторних занять, теоретичні відомості, практичні завдання, індивідуальні варіанти, контрольні питання та рекомендації щодо виконання й оформлення звітів. Призначене для здобувачів другого (магістерського) рівня вищої освіти спеціальності F7 Комп'ютерна інженерія освітньої програми «Комп'ютерна інженерія».

ЗМІСТ

Лабораторна робота №1. Модуль ESP32 в проєктах КФС/ІоТ	4
Лабораторне робота №2. Локальна мережа Wi-Fi на основі модуля ESP32	8
Лабораторне робота №3. Застосування стеку протоколів TCP/IP для передачі даних через Wi-Fi у вбудованих системах. lwIP. Технологія сокетів	11
Лабораторне робота №4. Застосування ESP32 для роботи з сенсорами у проєктах КФС/ІоТ. Розширювач GPIO PCF8574 для сенсорних мереж.....	15
Лабораторне робота №5. Застосування реконфігуровних середовищ у задачах синтезу проєктів КФС/ІоТ	18
Лабораторне робота №6. Методологія реалізації комплексних проєктів КФС/ІоТ	22
Лабораторне робота №7. Модель-орієнтоване проєктування КФС на основі використання профілю UML 2 та MARTE	25
Лабораторне робота №8. Модель-орієнтоване проєктування КФС на основі використання SysML	28
Лабораторне робота №9. Програмні засоби для інтелектуальних систем моделювання.....	31
Лабораторне робота №10. Моделювання ІоТ з використанням UML-діаграм на етапах архітектурного проєктування	34
Лабораторна робота №11. Дослідження поведінки систем Інтернету речей за допомогою Color Petri Net Tools.....	37
Лабораторна робота №12. Моделювання ІоТ з використанням темпоральної логіки на етапах їх архітектурного проєктування	40
Лабораторне заняття №13. Побудова інформаційних моделей на основі ІоТ-технологій пристроїв	43
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	46

Лабораторна робота №1

Тема: Модуль ESP32 в проектах КФС/ІоТ

Мета роботи: отримання навичок роботи в Espressif IoT Development Framework (Esp-IDF) з розробки програмного середовища (SDE), ознайомлення з ключовими особливостями модуля ESP32.

Теоретичні відомості

ESP32 – це мікроконтролер з вбудованими модулями Wi-Fi та Bluetooth, створений компанією Espressif. Відзначається високою продуктивністю, енергоефективністю та багатим набором периферійних інтерфейсів:

- Процесор: двоядерний Tensilica LX6 до 240 МГц;
 - Wi-Fi: 802.11 b/g/n;
 - Bluetooth: BLE 4.2;
 - Інтерфейси: ADC, DAC, I2C, SPI, UART, PWM, Touch, CAN.
- ESP32 у КФС/ІоТ:
- Збір і передача сенсорних даних у реальному часі.
 - Побудова розумних пристроїв з автономною логікою.
 - Комунікація з іншими модулями або хмарними платформами.

Практичне завдання

Приклад.

Опис проекту: «Міні-метеостанція»

ESP32 зчитує температуру та вологість з сенсора DHT22. Дані передаються у вебінтерфейс через вбудований вебсервер. Користувач може переглядати поточні показники через браузер.

1. Схема підключення

DHT22 → ESP32

VCC → 3.3V

DATA → GPIO 4

GND → GND

2. Програмування мікроконтролера:

```
#include <WiFi.h>
#include "DHT.h"
#define DHTPIN 4
#define DHTTYPE DHT22
const char* ssid = "IoT_Network";
const char* password = "12345678";
WiFiServer server(80);
DHT dht(DHTPIN, DHTTYPE);
void setup() {
  Serial.begin(115200);
  dht.begin();
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("WiFi connected!");
  server.begin();
}
void loop() {
  WiFiClient client = server.available();
```

```

if (client) {
  float t = dht.readTemperature();
  float h = dht.readHumidity();
  client.println("HTTP/1.1 200 OK");
  client.println("Content-type:text/html");
  client.println();
  client.print("<h1>Міні-метеостанція</h1>");
  client.print("<p>Температура: ");
  client.print(t);
  client.println(" &deg;C</p>");
  client.print("<p>Вологість: ");
  client.print(h);
  client.println(" %</p>");
  delay(100);
  client.stop();
}
}

```

3. Скріншот вікна веб-інтерфейсу:

Веб-інтерфейс ESP32

Міні-метеостанція

Температура: 22.4 °C

Вологість: 55 %

4. Висновок

ESP32 дозволяє швидко реалізувати IoT-проекти завдяки вбудованим комунікаційним модулям. У даній лабораторній було реалізовано вебінтерфейс для перегляду показників датчика DHT22, що демонструє можливість побудови простих кіберфізичних систем моніторингу.

Індивідуальні завдання

№ Варіанту	Індивідуальне завдання
1	Зчитування температури і вологості з DHT11 та передача через Wi-Fi
2	Вимірювання відстані за допомогою HC-SR04 та відображення на вебсторінці
3	Створення Bluetooth-сигналізатора на ESP32 з мобільним керуванням
4	Передача даних з фоторезистора на мобільний застосунок через Blynk
5	Реалізація IoT-сигналізації з сенсором руху PIR
6	Модуль управління LED-стрічкою через вебсервер ESP32
7	Система моніторингу температури в теплиці з графіком оновлень
8	Контроль вологості ґрунту з віддаленим увімкненням поливу
9	Моніторинг рівня води в баку з push-сповіщенням
10	IoT-кнопка «Паніка» з передачею тривоги через Wi-Fi
11	Передача даних з сенсора газу MQ-135 до хмарного сервера
12	IoT-світильник: керування яскравістю через PWM
13	Сенсорна система освітлення з керуванням через ESP-NOW
14	Індикатор освітлення на базі фоторезистора з вебінтерфейсом
15	Система контролю CO2 з ESP32 та OLED-дисплеєм
16	Вимірювання температури з BMP280 та публікація в MQTT-брокер
17	Керування вентилятором залежно від температури

18	Розумний годинник на ESP32 з дисплеєм і синхронізацією часу
19	Система контролю входу з RFID-модулем та ESP32
20	IoT-дзвінок: передача сигналу та зображення з камери
21	Мережевий датчик диму з push-повідомленням користувача
22	Зчитування пульсу з датчика MAX30100 та відображення в браузері
23	Автоматична система вентиляції теплиці з Wi-Fi моніторингом
24	Виявлення відкриття дверей та передача події у Telegram
25	Побудова локального сервера даних для IoT-сенсорів на ESP32
26	IoT-годинник із відображенням погоди через інтернет
27	Система поливу з датчиком дощу та ручним веб-керуванням
28	Керування розеткою з реле через вебінтерфейс
29	Збір даних з кількох сенсорів та збереження на SD-карті
30	Побудова інтелектуального термостата на базі ESP32
31	Підключення ESP32 до Google Firebase для зберігання даних
32	Система виявлення шуму з відображенням рівня на графіку
33	Керування серводвигуном через вебінтерфейс ESP32
34	Передача даних через LoRa між двома ESP32
35	Вебсервер з автентифікацією для доступу до IoT-пристрою
36	IoT-погодна станція з трьома сенсорами (температура, тиск, вологість)
37	Виявлення руху і активація відеоканери ESP32-CAM
38	Мережева кнопка ввімкнення/вимкнення побутового пристрою
39	Збір геолокаційних даних з GPS-модуля та передача по Wi-Fi
40	Побудова розумного дзеркала з ESP32 і LED-підсвіткою

Контрольні питання

1. Що таке модуль ESP32?
2. Яке призначення ESP32 у проектах КФС та IoT?
3. Що означає поняття Інтернет речей?
4. Що таке кіберфізична система?
5. Які основні відмінності між IoT-системою та кіберфізичною системою?
6. Які бездротові інтерфейси підтримує ESP32?
7. Для чого в ESP32 використовується Wi-Fi?
8. Для чого в ESP32 використовується Bluetooth?
9. Які переваги має ESP32 порівняно з Arduino Uno?
10. Які основні складові має плата ESP32?
11. Яку роль виконує мікроконтролер у вбудованій системі?
12. Які датчики можна підключати до ESP32?
13. Які виконавчі пристрої можна підключати до ESP32?
14. Що таке GPIO-виводи ESP32?
15. Для чого використовуються цифрові входи та виходи?
16. Для чого використовуються аналогові входи ESP32?
17. Що таке широтно-імпульсна модуляція?
18. Як ESP32 може використовуватися для збору даних із датчиків?
19. Як ESP32 може передавати дані в мережу?
20. Яке програмне середовище можна використовувати для програмування ESP32?
21. Для чого використовується Arduino IDE?
22. Що потрібно встановити в Arduino IDE для роботи з ESP32?
23. Що таке скетч у середовищі Arduino IDE?
24. Яке призначення функції setup()?
25. Яке призначення функції loop()?
26. Для чого використовується Serial Monitor?
27. Як перевірити, що ESP32 правильно підключено до комп'ютера?

28. Які типові помилки можуть виникати під час завантаження програми в ESP32?
29. Які переваги використання ESP32 у навчальних IoT-проектах?
30. Наведіть приклади практичного застосування ESP32 у кіберфізичних системах.

Лабораторне робота №2

Тема: Локальна мережа Wi-Fi на основі модуля ESP32

Мета роботи: вивчення принципів створення локальної бездротової мережі Wi-Fi на основі модуля ESP32, налаштування ESP32 у режимі точки доступу, підключення клієнтських пристроїв до створеної мережі та організація простого вебсерверу для обміну даними між ESP32 і користувачем.

Теоретичні відомості

ESP32 — це мікроконтролерний модуль, який має вбудовані засоби бездротового зв'язку Wi-Fi та Bluetooth. Завдяки цьому ESP32 широко використовується в системах Інтернету речей, локальних сенсорних мережах, системах моніторингу, керування пристроями та прототипах кіберфізичних систем.

У режимі Wi-Fi модуль ESP32 може працювати як клієнт бездротової мережі або як точка доступу. У режимі **STA** ESP32 підключається до вже наявної Wi-Fi-мережі, наприклад до домашнього маршрутизатора. У режимі **AP**, або **SoftAP**, сам ESP32 створює власну бездротову мережу, до якої можуть підключатися інші пристрої. Також ESP32 може працювати у комбінованому режимі **AP+STA**, тобто одночасно бути точкою доступу та клієнтом іншої мережі. Офіційна документація Arduino-ESP32 описує підтримку режимів Station, SoftAP, сканування мереж і базових режимів безпеки Wi-Fi.

Локальна мережа Wi-Fi на основі ESP32 може бути побудована без використання зовнішнього маршрутизатора. У такому випадку ESP32 створює мережу з власним іменем SSID та паролем. Користувач підключається до цієї мережі зі смартфона або ноутбука, після чого може відкрити вебсторінку, яка формується самим ESP32. У режимі SoftAP пристрої підключаються безпосередньо до ESP32 як до точки доступу.

Під час створення локальної Wi-Fi-мережі важливими параметрами є назва мережі, пароль, IP-адреса точки доступу, канал Wi-Fi та кількість підключених клієнтів. Типово ESP32 у режимі SoftAP має IP-адресу 192.168.4.1. Саме за цією адресою користувач може відкрити вебінтерфейс модуля у браузері.

Для реалізації вебсерверу в середовищі Arduino IDE можна використовувати бібліотеки WiFi.h та WebServer.h. Бібліотека WiFi.h відповідає за налаштування бездротового з'єднання, а WebServer.h дозволяє обробляти HTTP-запити від клієнтів. Наприклад, користувач може натиснути кнопку на вебсторінці, а ESP32 у відповідь увімкне або вимкне світлодіод.

Створення локальної мережі на основі ESP32 є зручним підходом для автономних пристроїв, які не потребують доступу до Інтернету. Такі рішення можуть використовуватися для локального керування освітленням, збору даних із датчиків, налаштування пристрою через вебінтерфейс, передавання параметрів або тестування IoT-систем.

Практичне завдання

Завдання. Створити локальну Wi-Fi-мережу на основі ESP32 у режимі точки доступу. Реалізувати простий вебсервер, який дозволяє користувачу через браузер керувати світлодіодом, підключеним до ESP32, та переглядати кількість підключених клієнтів.

Порядок виконання роботи

1. Підключити ESP32 до комп'ютера за допомогою USB-кабелю.
2. Відкрити Arduino IDE.
3. Вибрати плату ESP32 у меню **Tools / Board**.
4. Вибрати відповідний COM-порт.
5. Створити новий скетч.
6. Додати до програми бібліотеки:

```
#include <WiFi.h>
#include <WebServer.h>
```

7. Налаштувати назву та пароль Wi-Fi-мережі.
8. Запрограмувати ESP32 у режимі точки доступу.
9. Реалізувати вебсторінку з кнопками керування світлодіодом.
10. Завантажити програму в ESP32.
11. Підключитися зі смартфона або ноутбука до створеної мережі.
12. Відкрити у браузері адресу: 192.168.4.1

13. Перевірити роботу вебінтерфейсу та керування світлодіодом.
14. Після завантаження програми ESP32 створює локальну Wi-Fi-мережу з назвою ESP32_LAB_2. Користувач підключається до цієї мережі зі смартфона або ноутбука, відкриває у браузері адресу 192.168.4.1 і бачить вебсторінку для керування світлодіодом. Натискання кнопок «Увімкнути LED» та «Вимкнути LED» змінює стан світлодіода на платі ESP32.

Індивідуальні завдання

№ варіанта	Індивідуальне завдання
1	Створити Wi-Fi-мережу ESP32_NET_01, реалізувати керування одним світлодіодом через вебсторінку.
2	Змінити IP-адресу точки доступу на 192.168.10.1 та вивести її на вебсторінці.
3	Додати на вебсторінку відображення кількості підключених клієнтів.
4	Реалізувати керування двома світлодіодами через окремі кнопки.
5	Створити вебсторінку з кнопками «ON», «OFF» і «STATUS».
6	Додати автоматичне оновлення вебсторінки кожні 5 секунд.
7	Вивести на сторінці назву створеної Wi-Fi-мережі та IP-адресу ESP32.
8	Реалізувати зміну стану світлодіода однією кнопкою «Перемкнути».
9	Додати сторінку /info, на якій відображається інформація про ESP32.
10	Створити вебінтерфейс із простим CSS-оформленням.
11	Реалізувати керування світлодіодом, підключеним до GPIO 4.
12	Реалізувати керування світлодіодом, підключеним до GPIO 5.
13	Додати виведення повідомлення в Serial Monitor при кожному підключенні користувача до сторінки.
14	Реалізувати сторінку з трьома режимами: «LED увімкнено», «LED вимкнено», «миготіння».
15	Додати кнопку для перезапуску ESP32 через вебінтерфейс.
16	Реалізувати локальну мережу з паролем, який містить номер варіанта.
17	Додати на вебсторінку таблицю з параметрами мережі: SSID, IP, MAC-адреса, кількість клієнтів.
18	Реалізувати простий вебінтерфейс для керування реле замість світлодіода.
19	Підключити датчик освітленості та вивести його значення на вебсторінці.
20	Підключити датчик температури та вивести його значення у локальній Wi-Fi-мережі ESP32.

Контрольні питання

1. Що таке ESP32?
2. Які бездротові інтерфейси має ESP32?
3. Що таке локальна Wi-Fi-мережа?
4. У чому полягає режим Station для ESP32?
5. Що таке режим SoftAP?
6. Чим відрізняється режим AP від режиму STA?
7. Чи може ESP32 працювати без зовнішнього Wi-Fi-маршрутизатора?
8. Яке призначення SSID?
9. Для чого використовується пароль Wi-Fi-мережі?
10. Яка типова IP-адреса ESP32 у режимі SoftAP?
11. Для чого використовується бібліотека WiFi.h?
12. Для чого використовується бібліотека WebServer.h?
13. Що таке HTTP-запит?
14. Яке призначення функції server.on()?
15. Для чого використовується функція server.handleClient()?

16. Що означає порт 80 у вебсервері?
17. Як користувач підключається до локальної мережі ESP32?
18. Як можна керувати світлодіодом через вебсторінку?
19. Які переваги має локальна мережа на основі ESP32?
20. Де можуть використовуватися локальні Wi-Fi-мережі на базі ESP32?
21. Що таке IP-адреса пристрою?
22. Як перевірити, що ESP32 створив Wi-Fi-мережу?
23. Які дані можна передавати через вебінтерфейс ESP32?
24. Чому для автономних IoT-пристроїв зручно використовувати режим SoftAP?
25. Які недоліки має локальна мережа ESP32 без доступу до Інтернету?

Лабораторне робота №3

Тема : Застосування стеку протоколів TCP/IP для передачі даних через Wi-Fi у вбудованих системах. lwIP. Технологія сокетів

Мета роботи: вивчення принципів передавання даних через Wi-Fi у вбудованих системах на основі модуля ESP32, ознайомлення зі стеком протоколів TCP/IP, бібліотекою lwIP та технологією сокетів, а також набуття практичних навичок створення TCP-сервера на ESP32 для обміну повідомленнями з клієнтським пристроєм у локальній мережі.

Теоретичні відомості

1. Передавання даних у вбудованих Wi-Fi-системах

У сучасних кіберфізичних системах та IoT-проектах важливу роль відіграє передавання даних між вбудованими пристроями, серверами, мобільними застосунками та хмарними платформами. Модуль ESP32 має вбудований Wi-Fi-інтерфейс, що дозволяє створювати локальні мережі, підключатися до маршрутизатора або організувати обмін даними безпосередньо між пристроями.

Передавання даних через Wi-Fi зазвичай реалізується з використанням стеку протоколів TCP/IP. Саме TCP/IP забезпечує адресацію пристроїв, встановлення з'єднань, передавання пакетів, контроль доставки та взаємодію між мережевими вузлами.

У вбудованих системах стек TCP/IP повинен бути достатньо функціональним, але водночас економним щодо використання оперативної пам'яті та обчислювальних ресурсів. Для цього в ESP32 використовується легковаговий стек lwIP. ESP-IDF використовує відкритий стек lwIP, а Espressif підтримує роботу з BSD Sockets API для прикладних програм ESP32.

2. Стек протоколів TCP/IP

TCP/IP – це набір мережеских протоколів, які забезпечують обмін даними між пристроями в комп'ютерних мережах. У спрощеному вигляді стек TCP/IP можна подати через такі рівні:

Рівень	Призначення	Приклади протоколів
Прикладний	Взаємодія користувачьких програм	HTTP, MQTT, FTP, DNS
Транспортний	Передавання даних між процесами	TCP, UDP
Мережеский	IP-адресація та маршрутизація	IP, ICMP
Канальний	Передавання кадрів у фізичному середовищі	Wi-Fi, Ethernet

3. Протокол TCP

TCP, або Transmission Control Protocol, – це транспортний протокол, який забезпечує надійне передавання даних між двома пристроями. Перед початком обміну даними TCP встановлює логічне з'єднання між клієнтом і сервером.

Основні особливості TCP: TCP є протоколом зі встановленням з'єднання; забезпечує контроль доставки даних; гарантує правильну послідовність отримання повідомлень; дозволяє передавати потік байтів між клієнтом і сервером; використовується у вебсерверах, месенджерах, системах моніторингу, IoT-шлюзах та інших мережеских застосунках.

У контексті ESP32 TCP можна використати для передавання показників датчиків, приймання команд керування, обміну службовими повідомленнями або організації локального керування пристроєм без використання хмарного сервера.

4. Протокол UDP

UDP, або User Datagram Protocol, також належить до транспортного рівня, але працює без встановлення з'єднання. Він не гарантує доставку пакетів і не контролює їх порядок, однак має менші затримки та простішу структуру.

UDP доцільно використовувати для швидкого передавання невеликих повідомлень, телеметрії, локального виявлення пристроїв, потокових даних або випадків, коли втрата окремого пакета не є критичною.

5. Бібліотека lwIP

lwIP — це легковаговий стек TCP/IP, призначений для використання у вбудованих системах з обмеженими ресурсами. Його головна ідея полягає у зменшенні використання оперативної пам'яті при збереженні базової функціональності TCP/IP. Офіційний опис lwIP визначає його як невелику незалежну реалізацію TCP/IP, орієнтовану на зменшення використання RAM у вбудованих системах.

lwIP підтримує кілька способів взаємодії прикладних програм із TCP/IP-стеком. Зокрема, документація lwIP описує Raw API, Sequential API та BSD-style Socket API.

6. Технологія сокетів

Сокет — це програмний інтерфейс для обміну даними між пристроями через мережу. Сокет можна розглядати як кінцеву точку мережевого з'єднання. Кожен сокет пов'язаний з IP-адресою, портом і протоколом передавання даних. Для TCP-з'єднання зазвичай використовуються такі основні функції:

Функція	Призначення
socket()	створення сокета
bind()	прив'язка сокета до IP-адреси та порту
listen()	переведення сокета в режим очікування підключень
accept()	приймання підключення від клієнта
recv()	отримання даних
send()	надсилання даних
close()	закриття сокета

Серверна частина створює сокет, відкриває порт і очікує підключення. Клієнт підключається до IP-адреси сервера та його порту. Після встановлення TCP-з'єднання сторони можуть обмінюватися повідомленнями.

Практичне завдання

Завдання. Створити локальну Wi-Fi-мережу на основі ESP32 у режимі точки доступу. Реалізувати TCP-сервер з використанням технології сокетів. Забезпечити приймання текстових команд від клієнта та виконання дій на ESP32. Сервер повинен обробляти такі команди:

Команда	Дія
LED ON	увімкнути світлодіод
LED OFF	вимкнути світлодіод
STATUS	повернути поточний стан світлодіода
HELP	вивести список доступних команд

Порядок виконання роботи

1. Підключити ESP32 до комп'ютера через USB.
2. Відкрити Arduino IDE.
3. Перевірити, що встановлено пакет плат ESP32.
4. Вибрати відповідну плату ESP32 та COM-порт.
5. Створити новий скетч.
6. Додати бібліотеки для роботи з Wi-Fi та сокетами.
7. Налаштувати ESP32 у режимі точки доступу.
8. Створити TCP-сервер на порту 5000.
9. Реалізувати приймання команд від клієнта.
10. Завантажити програму в ESP32.
11. Підключити смартфон або ноутбук до Wi-Fi-мережі ESP32.
12. Підключитися TCP-клієнтом до адреси: 192.168.4.1
13. Вказати порт: 5000.
14. Надіслати команди LED_ON, LED_OFF, STATUS, HELP.
15. Перевірити відповіді сервера та роботу світлодіода.

Індивідуальні завдання

№ варіанта	Індивідуальне завдання
1	Реалізувати TCP-сервер на ESP32 з командами ON, OFF, STATUS для керування світлодіодом.

2	Змінити порт TCP-сервера на 6001 і перевірити підключення клієнта.
3	Додати команду BLINK, яка запускає миготіння світлодіода.
4	Реалізувати керування двома світлодіодами через команди LED1_ON, LED1_OFF, LED2_ON, LED2_OFF.
5	Додати команду CLIENTS, яка повертає кількість підключених Wi-Fi-клієнтів.
6	Реалізувати команду INFO, яка повертає SSID, IP-адресу та порт сервера.
7	Додати команду UPTIME, яка повертає час роботи ESP32 після запуску.
8	Реалізувати TCP-сервер, який повертає значення з аналогового входу ESP32.
9	Підключити датчик температури та передавати його значення через TCP.
10	Підключити датчик освітленості та передавати його значення клієнту.
11	Реалізувати команду RESTART, яка перезапускає ESP32.
12	Додати просту перевірку пароля перед виконанням команд керування.
13	Реалізувати TCP-сервер у режимі підключення ESP32 до зовнішнього Wi-Fi-роутера.
14	Реалізувати TCP-клієнт на ESP32, який підключається до сервера на комп'ютері.
15	Реалізувати передавання даних від ESP32 до Python-сервера на ПК.
16	Додати обробку невідомих команд із поверненням повідомлення про помилку.
17	Реалізувати журнал команд у Serial Monitor.
18	Реалізувати керування реле через TCP-команди.
19	Реалізувати UDP-обмін повідомленнями та порівняти його з TCP.
20	Розробити власний простий текстовий протокол обміну між TCP-клієнтом і ESP32.

Контрольні питання

1. Що таке стек протоколів TCP/IP?
2. Яке призначення протоколу IP?
3. Яке призначення транспортного рівня TCP/IP?
4. Що таке TCP?
5. Що таке UDP?
6. У чому полягає основна відмінність TCP від UDP?
7. Чому TCP називають протоколом зі встановленням з'єднання?
8. Що таке IP-адреса?
9. Що таке порт у мережевому з'єднанні?
10. Що таке сокет?
11. Які параметри визначають мережевий сокет?
12. Для чого використовується функція socket()?
13. Для чого використовується функція bind()?
14. Для чого використовується функція listen()?
15. Яке призначення функції accept()?
16. Для чого використовується функція recv()?
17. Для чого використовується функція send()?
18. Для чого використовується функція close()?
19. Що таке TCP-сервер?
20. Що таке TCP-клієнт?
21. Яку роль виконує ESP32 у режимі TCP-сервера?
22. Як клієнт підключається до TCP-сервера ESP32?
23. Який порт використано у практичному прикладі?
24. Яка типова IP-адреса ESP32 у режимі SoftAP?
25. Що таке lwIP?
26. Чому lwIP використовується у вбудованих системах?
27. Які API підтримує lwIP?
28. Що таке BSD Sockets API?
29. Які команди можна передавати на ESP32 через TCP?

30. Як можна використати TCP-з'єднання в IoT-системах?
31. Які переваги має TCP для передавання команд керування?
32. Які недоліки TCP у порівнянні з UDP?
33. Як перевірити працездатність TCP-сервера на ESP32?
34. Для чого використовується Serial Monitor під час тестування мережевого обміну?
35. Які помилки можуть виникати під час створення сокета?

Лабораторне робота №4

Тема: Застосування ESP32 для роботи з сенсорами у проектах КФС/ІоТ. Розширювач GPIO PCF8574 для сенсорних мереж

Мета роботи: вивчити особливості використання інтерфейсу I2C інтерфейсу в ESP32 для обміну даними з датчиками різних типів; отримати практичні навички програмування сенсорних мереж.

Теоретичні відомості

Мікроконтролер ESP32 з підтримкою Wi-Fi та Bluetooth, що має обмежену кількість GPIO (~30, але багато з них зайняті системними функціями). У складних ІоТ-проектах може виникнути потреба у додаткових цифрових лініях.

Розширювач GPIO PCF8574 /

Інтерфейс I²C розширює 2 лінії (SCL, SDA) до 8 цифрових ліній введення/виведення.

Можна підключити до 8 таких мікросхем (адресація 0x20...0x27), розширивши до 64 GPIO.

Практичне завдання

Приклад.

Побудувати прототип сенсорної мережі, де до розширювача PCF8574 підключено 8 кнопок. Замість використання 8 цифрових портів ESP32 використовується лише 2 лінії I²C (SDA, SCL), що значно зменшує навантаження на плату.

1. Схема підключення

Кнопки → P0...P7 на PCF8574

PCF8574 VCC → 3.3V ESP32

PCF8574 GND → GND ESP32

PCF8574 SDA → GPIO 21 ESP32

PCF8574 SCL → GPIO 22 ESP32

2. Програмування мікроконтролера:

```
#include <Wire.h>
#include <PCF8574.h>
PCF8574 expander(0x20); // Адреса модуля PCF8574
void setup() {
  Serial.begin(115200);
  Wire.begin(21, 22); // I2C пін ESP32
  expander.begin();
  for (int i = 0; i < 8; i++) {
    expander.pinMode(i, INPUT_PULLUP);
  }
  Serial.println("Початок моніторингу кнопок через PCF8574...");
}
void loop() {
  for (int i = 0; i < 8; i++) {
    int val = expander.digitalRead(i);
    Serial.print("Кнопка ");
    Serial.print(i);
    Serial.print(": ");
    Serial.println(val == LOW ? "Натиснута" : "Відпущена");
  }
  Serial.println("-----");
  delay(1000);
}
```

3. Скріншот з Arduino IDE (Серійний монітор):

Кнопка 0: Відпущена
 Кнопка 1: Натиснута
 Кнопка 2: Відпущена
 Кнопка 3: Відпущена
 Кнопка 4: Відпущена
 Кнопка 5: Відпущена
 Кнопка 6: Відпущена
 Кнопка 7: Відпущена

4. Висновок

Використання розширювача PCF8574 значно розширює можливості ESP32 у сфері IoT та КФС-проектів. Це дозволяє будувати сенсорні мережі з великою кількістю входів/виходів без перевантаження мікроконтролера.

Індивідуальні завдання

№ Варіанту	Індивідуальне завдання
1	Підключення 8 кнопок до ESP32 через PCF8574 і зчитування їх станів
2	Підключення 8 LED до ESP32 через PCF8574 і керування блиманням
3	Побудова клавіатури 4x2 на основі PCF8574 і ESP32
4	Індикація натискань кнопок через серіальний порт ESP32
5	Система охорони з підключенням сенсорів відкриття дверей через PCF8574
6	Увімкнення/вимкнення світлодіодів за станом кнопок через розширювач
7	Підключення температурних сенсорів через PCF8574 з опитуванням
8	Керування реле через PCF8574 для створення розумного дому
9	Реалізація виводу двійкового коду на 8-сегментний індикатор через PCF8574
10	Зчитування 8 фоторезисторів через цифрові порти PCF8574
11	Індикація стану сенсорів через LED-матрицю підключену через PCF8574
12	Використання кількох PCF8574 для керування світлодіодною панеллю
13	Моніторинг стану 8 дверей або вікон з виводом на вебінтерфейс
14	Керування яскравістю LED через PCF8574 за аналогією з PWM
15	Реалізація простого охоронного пульта через клавіатуру на PCF8574
16	Побудова міні-системи голосування з 8 кнопками
17	Контроль присутності на основі сенсорів, підключених через PCF8574
18	Контроль доступу до кімнат через кнопки і світлову індикацію
19	Створення звукового сигналу при натисканні будь-якої кнопки
20	Збереження стану кнопок у EEPROM після натискання
21	Зміна кольору RGB-світлодіода в залежності від кнопки (через PCF8574)
22	Вивід повідомлень на OLED при натисканні кнопок через PCF8574
23	Керування цифровими сервомоторами через розширений GPIO
24	Використання PCF8574 для дистанційного пульта ESP32
25	Керування 8-релейним модулем для живлення пристроїв
26	Індикатор зайнятості (наприклад, переговорна) через LED
27	Мікроігра «Вгадай кнопку» з LED-індикацією перемоги
28	Моніторинг натискань з передачею в MQTT-брокер
29	Побудова цифрового замка з кодовим введенням на кнопках
30	Таймер на базі ESP32, керований через кнопки на PCF8574
31	Вивід значень кнопок у вигляді логічних рівнів в браузер
32	Побудова інтерфейсу для контролю доступу через keypad + PCF8574

33	Індикація тривоги при натисканні визначеної кнопки
34	Контроль світлової сигналізації в класі/офісі
35	Керування кількома модулями LED через різні адреси PCF8574
36	Лічильник кількості натискань кнопки з відображенням на екрані
37	Кнопки запуску/зупинки пристроїв на виробництві
38	Автоматичне тестування контактів з виводом результатів
39	Контроль температури і вивід порогового стану через LED
40	Підключення сенсорної панелі до ESP32 через розширювач портів

Контрольні питання

1. Що таке ESP32?
2. Яке призначення ESP32 у проектах КФС/ІоТ?
3. Що таке сенсор у вбудованій системі?
4. Які типи сенсорів можуть використовуватися в ІоТ-проектах?
5. Яку роль виконують сенсори у кіберфізичних системах?
6. Що таке сенсорна мережа?
7. Які дані можуть збирати сенсорні вузли?
8. Які інтерфейси використовуються для підключення сенсорів до ESP32?
9. Що таке GPIO-виводи ESP32?
10. Для чого використовуються цифрові входи ESP32?
11. Для чого використовуються цифрові виходи ESP32?
12. Для чого використовуються аналогові входи ESP32?
13. Які обмеження має ESP32 щодо кількості доступних GPIO-выводів?
14. Що таке розширювач GPIO?
15. Для чого використовується мікросхема PCF8574?
16. Яке основне призначення PCF8574 у сенсорних мережах?
17. Скільки додаткових портів введення/виведення надає PCF8574?
18. Який інтерфейс використовується для підключення PCF8574 до ESP32?
19. Що таке інтерфейс I2C?
20. Які дві основні лінії використовуються в шині I2C?
21. Для чого використовується лінія SDA?
22. Для чого використовується лінія SCL?
23. Які типові GPIO ESP32 можуть використовуватися для I2C?
24. Що таке адреса пристрою на шині I2C?
25. Чому кожен пристрій I2C повинен мати унікальну адресу?

Лабораторне робота №5

Тема: Застосування реконфігурованих середовищ у задачах синтезу проєктів КФС/ІоТ

Мета роботи: ознайомлення з принципами використання реконфігурованих середовищ для моделювання, проєктування та синтезу кіберфізичних систем і ІоТ-рішень, формування практичних навичок створення віртуальної моделі ІоТ-пристрою на основі ESP32, сенсорів та виконавчих елементів, а також перевірка логіки роботи системи до її фізичної реалізації.

Теоретичні відомості

1. Поняття реконфігурованого середовища

Реконфігуроване середовище — це програмне або апаратно-програмне середовище, яке дозволяє змінювати структуру, параметри, логіку роботи або взаємозв'язки між компонентами системи без повного перепроектування всієї системи.

У задачах КФС/ІоТ такі середовища використовуються для швидкого створення, перевірки та модифікації прототипів. Завдяки цьому розробник може змінювати сенсори, алгоритми керування, способи передавання даних, логіку обробки подій і структуру взаємодії компонентів.

До реконфігурованих середовищ можна віднести:

- Wokwi;
- Tinkercad Circuits;
- Node-RED;
- Arduino IDE;
- ESP-IDF;
- MATLAB/Simulink;
- LabVIEW;
- середовища FPGA-проєктування;
- хмарні ІоТ-платформи.

2. Кіберфізичні системи та ІоТ-проєкти

Кіберфізична система — це система, у якій обчислювальні компоненти взаємодіють із фізичними процесами. Така система отримує дані від сенсорів, аналізує їх, приймає рішення та впливає на фізичне середовище через виконавчі пристрої.

ІоТ-проєкт є прикладом кіберфізичної системи, якщо він забезпечує зв'язок між фізичним об'єктом і цифровою частиною системи. Наприклад, ESP32 може вимірювати температуру, передавати дані через Wi-Fi, виводити інформацію на вебсторінку та вмикати вентилятор при перевищенні заданого порогу.

Типова структура КФС/ІоТ-проєкту містить:

Компонент	Призначення
Сенсори	Отримання даних із фізичного середовища
Мікроконтролер	Обробка даних і виконання алгоритму керування
Мережевий модуль	Передавання даних через Wi-Fi, Bluetooth або інші канали
Виконавчі пристрої	Вплив на фізичний процес
Програмне забезпечення	Логіка роботи системи
Інтерфейс користувача	Відображення даних і керування системою

3. Синтез проєкту КФС/ІоТ

Синтез проєкту – це процес побудови структури системи, вибору її компонентів, визначення алгоритмів роботи та налаштування взаємодії між апаратною і програмною частинами.

У контексті КФС/ІоТ синтез включає:

- вибір мікроконтролера;
- вибір сенсорів;
- вибір виконавчих пристроїв;
- побудову схеми підключення;
- створення алгоритму роботи;
- написання програмного коду;

- тестування системи;
- аналіз результатів;
- внесення змін до проєкту.

4. Переваги використання реконфігурованих середовищ

Використання реконфігурованих середовищ має низку переваг для розроблення КФС/ІоТ-проєктів.

По-перше, вони дозволяють зменшити витрати на фізичні компоненти, оскільки значну частину перевірки можна виконати у віртуальному середовищі.

По-друге, такі середовища дають можливість швидко змінювати структуру системи. Наприклад, можна замінити датчик температури на датчик освітленості або додати релейний модуль без повного переписування проєкту.

По-третє, реконфігуровані середовища зручні для навчання, оскільки здобувачі можуть бачити результат роботи програми одразу після запуску симуляції.

По-четверте, вони допомагають уникнути типових апаратних помилок, зокрема неправильного підключення живлення, перевантаження виводів або помилок у логіці керування.

Практичне завдання

Завдання. Створити у реконфігурованому середовищі віртуальну модель ІоТ-пристрою на основі ESP32. Система повинна отримувати дані від сенсора, аналізувати їх і керувати виконавчим елементом відповідно до заданого алгоритму.

Алгоритм роботи системи

1. ESP32 запускає програму.
2. Ініціалізується датчик температури та вологості.
3. Зчитуються поточні значення температури й вологості.
4. Значення виводяться у Serial Monitor.
5. Якщо температура перевищує заданий поріг, вмикається світлодіод.
6. Якщо температура нижча за поріг, світлодіод вимикається.
7. Через певний проміжок часу вимірювання повторюється.

Схема підключення:

Компонент	Вивід компонента	Вивід ESP32
DHT22	VCC	3.3V
DHT22	GND	GND
DHT22	DATA	GPIO 15
LED	Анод	GPIO 2 через резистор
LED	Катод	GND

Порядок виконання роботи

1. Відкрити реконфігуроване середовище для моделювання ІоТ-пристроїв.
2. Створити новий проєкт на основі ESP32.
3. Додати до схеми датчик DHT22.
4. Додати світлодіод або релейний модуль.
5. Виконати підключення компонентів відповідно до таблиці.
6. Додати бібліотеку для роботи з DHT22.
7. Написати програму для зчитування температури та вологості.
8. Реалізувати умову вмикання світлодіода при перевищенні температурного порогу.
9. Запустити симуляцію.
10. Перевірити результати роботи у Serial Monitor.
11. Змінити порогове значення температури.
12. Перевірити, як зміна параметрів впливає на роботу системи.
13. Виконати індивідуальне завдання згідно з варіантом.
14. Зробити скріншоти схеми, коду та результатів симуляції.
15. Сформулювати висновок.

Індивідуальні завдання

№ варіанта	Індивідуальне завдання
1	Створити систему контролю температури з порогом 25 °С. При перевищенні порогу вмикати LED.
2	Створити систему контролю температури з порогом 28 °С. Виводити температуру в Serial Monitor.
3	Реалізувати систему контролю вологості. При вологості нижче 40 % вмикати LED.
4	Створити систему контролю вологості. При вологості вище 70 % вмикати попереджувальний сигнал.
5	Підключити два світлодіоди: зелений для нормального стану, червоний для аварійного стану.
6	Реалізувати систему контролю температури з трьома станами: норма, попередження, аварія.
7	Створити модель системи автоматичного освітлення на основі фоторезистора.
8	Реалізувати систему, у якій LED вмикається при недостатньому рівні освітленості.
9	Створити модель охоронної системи з кнопкою або PIR-сенсором.
10	Реалізувати систему сигналізації: при спрацюванні сенсора вмикати світлодіод і buzzer.
11	Створити систему контролю відкриття дверей з використанням кнопки як сенсора.
12	Реалізувати систему керування реле залежно від значення сенсора температури.
13	Створити модель «розумної теплиці» з контролем температури та вологості.
14	Реалізувати систему автоматичного поливу з умовним датчиком вологості ґрунту.
15	Створити модель метеостанції з виведенням температури й вологості в Serial Monitor.
16	Реалізувати систему з двома сенсорами та одним виконавчим пристроєм.
17	Створити систему з одним сенсором і двома виконавчими пристроями.
18	Реалізувати зміну порогового значення безпосередньо у програмному коді та порівняти результати.
19	Додати до моделі LCD-дисплей для виведення значень сенсорів.
20	Розробити власну модель IoT-пристрою у реконфігурованому середовищі з описом алгоритму роботи.

Контрольні питання

1. Що таке реконфігуроване середовище?
2. Для чого використовуються реконфігуровані середовища у КФС/IoT?
3. Що таке синтез проекту КФС/IoT?
4. Які етапи включає синтез IoT-проекту?
5. Що таке кіберфізична система?
6. Чим IoT-система відрізняється від звичайної вбудованої системи?
7. Яку роль виконує ESP32 у КФС/IoT-проектах?
8. Які переваги має використання ESP32 у навчальних IoT-проектах?
9. Що таке сенсор у кіберфізичній системі?
10. Що таке виконавчий пристрій?
11. Яке призначення віртуального моделювання?
12. Які переваги має моделювання перед фізичним складанням пристрою?
13. Які реконфігуровані середовища можна використовувати для IoT-проектів?
14. Що таке Wokwi?
15. Які компоненти можна моделювати у Wokwi?

16. Для чого використовується Tinkercad Circuits?
17. Що таке Node-RED?
18. Як Node-RED може використовуватися в IoT-системах?
19. Що таке MATLAB/Simulink у контексті моделювання КФС?
20. Яке призначення Arduino IDE?
21. Чому реконфігуровні середовища зручні для навчання?
22. Які параметри системи можна змінювати у реконфігуровному середовищі?
23. Що таке алгоритм роботи IoT-пристрою?
24. Яке призначення Serial Monitor?
25. Для чого використовується датчик DHT22?

Лабораторне робота №6

Тема: Методологія реалізації комплексних проєктів КФС/ІоТ

Мета роботи: ознайомлення з методологією реалізації комплексних проєктів кіберфізичних систем та Інтернету речей, формування практичних навичок планування структури ІоТ-рішення, визначення апаратних і програмних компонентів, побудови алгоритму роботи системи та підготовки проєкту до подальшої реалізації.

Теоретичні відомості

Комплексний проєкт КФС/ІоТ — це система, яка поєднує фізичні об'єкти, сенсори, виконавчі пристрої, мікроконтролери, мережеві технології та програмне забезпечення. Такий проєкт не обмежується лише підключенням одного датчика до мікроконтролера, а передбачає взаємодію кількох апаратних і програмних складових.

Кіберфізична система працює на основі постійного обміну даними між фізичним середовищем і цифровою частиною. Сенсори отримують інформацію про стан об'єкта або середовища, мікроконтролер обробляє ці дані, а виконавчі пристрої виконують певні дії. У проєктах ІоТ до цієї структури додається мережевий рівень, який забезпечує передавання даних між пристроями, сервером, хмарною платформою або користувацьким інтерфейсом.

Методологія реалізації комплексного проєкту передбачає послідовне проходження основних етапів: аналіз задачі, визначення вимог, вибір компонентів, розроблення структури системи, створення алгоритму роботи, програмування, тестування та оцінювання результатів. Важливо, щоб кожен етап був логічно пов'язаний з попереднім, а обрані апаратні та програмні рішення відповідали поставленій меті.

Під час створення КФС/ІоТ-проєкту особливу увагу потрібно приділяти сумісності компонентів, стабільності живлення, надійності передавання даних, захисту інформації та можливості подальшого масштабування системи. Наприклад, якщо система моніторингу температури в майбутньому має бути розширена сенсорами вологості, освітленості або руху, це потрібно враховувати вже на етапі проєктування.

Для навчальних проєктів зручно використовувати ESP32, оскільки цей модуль має достатню кількість GPIO-виводів, підтримує Wi-Fi і Bluetooth, працює з різними сенсорами та може виконувати функції локального сервера, клієнта мережі або вузла збору даних. Це дозволяє створювати як прості автономні пристрої, так і складніші ІоТ-системи з вебінтерфейсом, передаванням даних і керуванням виконавчими пристроями.

Практичне завдання

Завдання. Розробити концепцію комплексного КФС/ІоТ-проєкту на основі ESP32. Проєкт повинен містити сенсорну частину, керувальний модуль, виконавчий пристрій, мережеву взаємодію та опис алгоритму роботи системи.

Базовий приклад: система моніторингу мікроклімату приміщення на основі ESP32, датчика температури та вологості DHT22, світлодіодної індикації та локального вебінтерфейсу для перегляду показників.

Порядок виконання роботи

1. Обрати тему комплексного КФС/ІоТ-проєкту.
2. Визначити призначення системи.
3. Описати об'єкт керування або моніторингу.
4. Визначити, які фізичні параметри буде вимірювати система.
5. Обрати сенсори для збору даних.
6. Обрати виконавчі пристрої.
7. Обрати мікроконтролер або платформу керування.
8. Визначити спосіб передавання даних.
9. Побудувати структурну схему системи.
10. Описати алгоритм роботи системи.
11. Розробити таблицю апаратних компонентів.
12. Розробити фрагмент програмного коду або псевдокод.

13. Виконати моделювання окремої частини системи.
14. Описати очікуваний результат роботи системи.
15. Зробити висновок щодо можливості практичної реалізації проєкту.

Приклад структури комплексного проєкту

Компонент системи	Приклад реалізації	Призначення
Мікроконтролер	ESP32	Обробка даних і керування системою
Сенсор температури	DHT22	Вимірювання температури
Сенсор вологості	DHT22	Вимірювання вологості
Виконавчий пристрій	LED або реле	Сигналізація або керування навантаженням
Мережевий модуль	Wi-Fi ESP32	Передавання даних
Інтерфейс користувача	Вебсторінка	Перегляд показників і стану системи
Джерело живлення	USB або блок живлення 5 В	Живлення пристрою

Індивідуальні завдання

№ варіанта	Тема комплексного КФС/ІоТ-проєкту
1	Система моніторингу температури в навчальній аудиторії
2	Система контролю вологості повітря у приміщенні
3	Система автоматичного освітлення на основі ESP32
4	Система моніторингу освітленості робочого місця
5	Система контролю відкриття дверей
6	Система охоронної сигналізації на основі датчика руху
7	Система моніторингу мікроклімату теплиці
8	Система автоматичного поливу рослин
9	Система контролю рівня води в резервуарі
10	Система виявлення витоку газу
11	Система моніторингу якості повітря
12	Система контролю доступу до приміщення
13	Система керування вентиляцією за температурою
14	Система локального вебмоніторингу сенсорних даних
15	Система дистанційного керування реле через Wi-Fi
16	Система збору даних із кількох сенсорів
17	Система моніторингу енергоспоживання
18	Система контролю стану обладнання
19	Система моніторингу параметрів довкілля
20	Власний комплексний КФС/ІоТ-проєкт на основі ESP32

Контрольні питання

1. Що таке комплексний КФС/ІоТ-проєкт?
2. Чим комплексний проєкт відрізняється від простого підключення одного сенсора?
3. Що таке кіберфізична система?
4. Що таке Інтернет речей?
5. Яку роль виконує ESP32 у КФС/ІоТ-проєктах?
6. Що таке сенсорна частина системи?
7. Для чого використовуються виконавчі пристрої?
8. Що таке об'єкт моніторингу?
9. Що таке об'єкт керування?
10. Які етапи включає реалізація комплексного ІоТ-проєкту?
11. Чому важливо визначити вимоги до системи перед її реалізацією?
12. Для чого потрібна структурна схема системи?

13. Що відображає алгоритм роботи системи?
14. Як вибираються сенсори для IoT-проекту?
15. Як вибираються виконавчі пристрої?
16. Які мережеві технології може використовувати ESP32?
17. Чому Wi-Fi зручний для IoT-проектів?
18. Які дані може передавати ESP32 у мережу?
19. Що таке локальний вебінтерфейс?
20. Як ESP32 може використовуватися як вебсервер?
21. Чому важливо враховувати живлення пристрою?
22. Які проблеми можуть виникати через нестабільне живлення?
23. Що таке сумісність компонентів?
24. Чому потрібно враховувати рівні логічних сигналів?
25. Що таке масштабування IoT-системи?

Лабораторне робота №7

Тема: Модель-орієнтоване проектування КФС на основі використання профілю UML 2 та MARTE

Мета роботи: ознайомлення з принципами модель-орієнтованого проектування кіберфізичних систем, вивчення можливостей UML 2 для опису структури та поведінки системи, а також застосування профілю MARTE для моделювання апаратних, програмних і часових характеристик КФС/ІоТ-проектів.

Теоретичні відомості

Модель-орієнтоване проектування — це підхід, за якого основна увага приділяється створенню формалізованої моделі системи до її безпосередньої програмної або апаратної реалізації. Такий підхід дозволяє описати структуру системи, логіку взаємодії компонентів, потоки даних, поведінку об'єктів і вимоги до часу виконання окремих процесів.

UML, або Unified Modeling Language, є стандартною мовою графічного моделювання, яка використовується для візуалізації, специфікації, конструювання та документування програмних і системних рішень. Офіційна специфікація OMG визначає UML як мову для опису артефактів розподілених об'єктних систем.

У проектах КФС/ІоТ UML 2 може застосовуватися для побудови діаграм варіантів використання, класів, компонентів, послідовності, активності та станів. Наприклад, для системи моніторингу мікроклімату UML дозволяє описати користувача, сенсори, ESP32, модуль Wi-Fi, вебінтерфейс і виконавчі пристрої.

MARTE — це профіль UML, призначений для моделювання та аналізу систем реального часу і вбудованих систем. Офіційна специфікація OMG зазначає, що MARTE розширює UML можливостями для модельно-орієнтованої розробки Real-Time and Embedded Systems.

Профіль MARTE дозволяє уточнювати модель за допомогою спеціальних стереотипів і властивостей. У контексті КФС/ІоТ це дає змогу показати, які елементи є апаратними ресурсами, які виконують обчислення, які відповідають за сенсорні дані, які виконують керування, а також які часові обмеження має система.

Наприклад, у моделі КФС на основі ESP32 мікроконтролер можна позначити як обчислювальний апаратний вузол, сенсор температури — як сенсорний пристрій, реле або світлодіод — як виконавчий пристрій, а процес зчитування даних — як періодичну дію з певним інтервалом виконання.

Практичне завдання

Завдання. Розробити модель кіберфізичної системи або ІоТ-проекту на основі UML 2 та елементів профілю MARTE. Модель повинна описувати структуру системи, взаємодію компонентів, логіку роботи та часові характеристики окремих процесів.

Базовий приклад: система моніторингу мікроклімату приміщення на основі ESP32, датчика DHT22, Wi-Fi-модуля, вебінтерфейсу та світлодіодного або релейного виконавчого пристрою.

Порядок виконання роботи

1. Обрати тему КФС/ІоТ-проекту.
2. Визначити об'єкт моніторингу або керування.
3. Визначити основні компоненти системи: сенсори, ESP32, мережевий модуль, виконавчі пристрої, користувацький інтерфейс.
4. Побудувати діаграму варіантів використання.
5. Побудувати діаграму компонентів системи.
6. Побудувати діаграму класів або структурну модель системи.
7. Побудувати діаграму послідовності для основного сценарію роботи.
8. Побудувати діаграму станів для пристрою або підсистеми керування.
9. Додати до моделі елементи профілю MARTE для позначення апаратних, програмних і часових характеристик.
10. Описати часові параметри роботи системи.
11. Зробити висновок щодо переваг модель-орієнтованого підходу.

Приклад опису системи. Проектується система моніторингу мікроклімату приміщення. Сенсор DHT22 періодично вимірює температуру та вологість. ESP32 обробляє отримані дані, порівнює температуру з пороговим значенням і передає результати користувачу через Wi-Fi. Якщо температура перевищує встановлений поріг, система активує виконавчий пристрій, наприклад світлодіод або релейний модуль.

Основні компоненти моделі

Компонент	Тип у системі	Приклад позначення MARTE
ESP32	Обчислювальний вузол	<<HwProcessor>>
DHT22	Сенсор температури та вологості	<<HwSensor>>
Wi-Fi-модуль	Комунікаційний ресурс	<<HwCommunicationResource>>
LED або реле	Виконавчий пристрій	<<HwActuator>>
Програма керування	Програмний компонент	<<SwSchedulableResource>>
Цикл вимірювання	Періодичний процес	period = 2000 ms
Вебінтерфейс	Інтерфейс користувача	<<ClientServerFeature>>

Приклад UML-діаграми варіантів використання

Основні актори системи:

Актор	Роль
Користувач	Переглядає показники та стан системи
Адміністратор	Налаштовує порогові значення
ESP32	Збирає, обробляє та передає дані
Сенсор	Надає фізичні параметри середовища

Індивідуальні завдання

№ варіанта	Тема КФС/ІоТ-проєкту для моделювання
1	Модель системи моніторингу температури приміщення
2	Модель системи контролю вологості повітря
3	Модель системи автоматичного освітлення
4	Модель системи контролю рівня освітленості
5	Модель системи виявлення руху
6	Модель системи охоронної сигналізації
7	Модель системи моніторингу мікроклімату теплиці
8	Модель системи автоматичного поливу
9	Модель системи контролю рівня води
10	Модель системи виявлення витoku газу
11	Модель системи моніторингу якості повітря
12	Модель системи контролю доступу
13	Модель системи керування вентиляцією
14	Модель системи локального вебмоніторингу
15	Модель системи дистанційного керування реле
16	Модель сенсорної мережі на основі ESP32
17	Модель системи моніторингу енергоспоживання
18	Модель системи контролю стану обладнання
19	Модель системи моніторингу параметрів довкілля
20	Власна модель КФС/ІоТ-проєкту з використанням ESP32

Контрольні питання

1. Що таке модель-орієнтоване проєктування?
2. Для чого використовується UML?
3. Що таке UML 2?
4. Яке призначення UML-діаграм у проєктуванні КФС?
5. Що таке профіль UML?

6. Що таке MARTE?
7. Для яких систем призначений профіль MARTE?
8. Чому MARTE доцільно використовувати для КФС/ІоТ-проектів?
9. Які характеристики системи можна уточнювати за допомогою MARTE?
10. Що таке стереотип у UML?
11. Для чого використовуються стереотипи MARTE?
12. Як у моделі можна позначити мікроконтролер ESP32?
13. Як у моделі можна позначити сенсор?
14. Як у моделі можна позначити виконавчий пристрій?
15. Що таке діаграма варіантів використання?
16. Для чого використовується діаграма компонентів?
17. Що показує діаграма класів?
18. Що показує діаграма послідовності?
19. Що показує діаграма станів?
20. Як UML допомагає описати поведінку системи?
21. Як UML допомагає описати структуру системи?
22. Що таке актор у діаграмі варіантів використання?
23. Що таке сценарій роботи системи?
24. Що таке часове обмеження в КФС?
25. Що означає періодичність виконання процесу?

Лабораторне робота №8

Тема: Модель-орієнтоване проєктування КФС на основі використання SysML

Мета роботи: ознайомлення з основами модель-орієнтованого проєктування кіберфізичних систем на основі мови SysML, формування практичних навичок побудови системної моделі КФС/ІоТ-проєкту, опису вимог, структури, поведінки та взаємозв'язків між апаратними й програмними компонентами системи.

Теоретичні відомості

SysML, або Systems Modeling Language, — це мова системного моделювання, яка використовується для опису складних технічних систем. Вона застосовується у модель-орієнтованому системному інжинірингу для специфікації, аналізу, проєктування та перевірки систем, що можуть включати апаратні засоби, програмне забезпечення, дані, персонал, процедури та інфраструктуру. Таке визначення подає Object Management Group, яка супроводжує специфікації SysML.

На відміну від UML, який переважно орієнтований на програмні системи, SysML краще підходить для опису комплексних кіберфізичних систем. У КФС потрібно моделювати не лише програмний код, а й фізичні об'єкти, сенсори, виконавчі пристрої, енергоживлення, потоки даних, обмеження, вимоги та взаємодію з користувачем.

SysML дозволяє створювати різні типи діаграм. Для КФС/ІоТ-проєктів найбільш важливими є діаграма вимог, діаграма блоків, діаграма внутрішньої структури блоків, діаграма активності, діаграма станів, діаграма послідовності та параметрична діаграма. Завдяки цим діаграмам можна послідовно описати, що система повинна виконувати, з яких частин вона складається, як ці частини взаємодіють між собою, які дані передаються між компонентами та які обмеження впливають на роботу системи.

У контексті КФС/ІоТ SysML зручно використовувати для проєктування систем на основі ESP32. Наприклад, система моніторингу мікроклімату може містити ESP32, датчик температури та вологості, Wi-Fi-модуль, вебінтерфейс, світлодіодну індикацію або реле. SysML-модель дозволяє відобразити не тільки ці компоненти, а й вимоги до точності вимірювання, періодичності зчитування даних, часу реакції системи та способу передавання інформації користувачу.

Актуальна специфікація SysML v2 розширює можливості моделювання вимог, поведінки, структури, аналізу та перевірки, а також підтримує узгодженість і простежуваність між різними частинами системної моделі. Водночас у навчальних роботах можна використовувати як підходи SysML v1, так і спрощені елементи SysML v2 залежно від доступного програмного середовища.

Практичне завдання

Завдання. Розробити SysML-модель кіберфізичної системи або ІоТ-проєкту. Модель повинна містити опис вимог, структуру системи, взаємодію компонентів, основний алгоритм роботи та обмеження, які впливають на функціонування системи.

Базовий приклад: система моніторингу мікроклімату приміщення на основі ESP32, датчика DHT22, Wi-Fi-з'єднання, локального вебінтерфейсу та світлодіодної індикації аварійного стану.

Порядок виконання роботи

1. Обрати тему КФС/ІоТ-проєкту.
2. Визначити призначення системи.
3. Сформулювати основні функціональні та нефункціональні вимоги.
4. Побудувати діаграму вимог SysML.
5. Побудувати блокову діаграму системи.
6. Побудувати діаграму внутрішньої структури блоків.
7. Побудувати діаграму активності для основного алгоритму роботи.
8. Побудувати діаграму станів для пристрою або підсистеми керування.
9. Побудувати параметричну діаграму з основними обмеженнями системи.
10. Описати взаємозв'язок між вимогами, компонентами та поведінкою системи.
11. Підготувати короткий висновок щодо доцільності використання SysML для

проектування КФС.

Приклад опису системи. Проектується система моніторингу мікроклімату приміщення. ESP32 отримує дані з датчика DHT22, обробляє значення температури й вологості, порівнює їх із заданими пороговими значеннями та передає інформацію користувачу через локальний вебінтерфейс. Якщо температура перевищує допустимий рівень, система активує світлодіодну індикацію або релейний модуль.

Приклад вимог до системи.

Позначення	Вимога	Тип вимоги
R1	Система повинна вимірювати температуру повітря	Функціональна
R2	Система повинна вимірювати вологість повітря	Функціональна
R3	Система повинна передавати дані користувачу через Wi-Fi	Функціональна
R4	Система повинна активувати індикацію при перевищенні температурного порогу	Функціональна
R5	Період оновлення даних не повинен перевищувати 2 с	Нефункціональна
R6	Час реакції на аварійний стан не повинен перевищувати 1 с	Нефункціональна

Приклад блокової структури системи

Блок SysML	Призначення
MonitoringSystem	Загальна система моніторингу
ESP32Controller	Обробка даних і керування компонентами
DHT22Sensor	Вимірювання температури та вологості
WiFiInterface	Передавання даних користувачу
WebInterface	Відображення показників
AlarmIndicator	Індикація аварійного стану
PowerSupply	Живлення системи

Приклад діаграми активності

Основний алгоритм роботи системи можна подати так:

Початок

Ініціалізація ESP32

Ініціалізація DHT22

Запуск Wi-Fi

Зчитування температури та вологості

Перевірка коректності даних

Порівняння температури з порогом

Якщо температура > 30 °C – увімкнути індикацію

Якщо температура <= 30 °C – вимкнути індикацію

Передати дані у вебінтерфейс

Пауза 2 с

Повторити цикл

Приклад параметричної діаграми

Параметр	Позначення	Значення
Поточна температура	T	вимірюється сенсором
Гранична температура	Tmax	30 °C
Період вимірювання	P	2 с
Час реакції	Rt	не більше 1 с
Стан сигналізації	A	0 або 1

Індивідуальні завдання

№ варіанта	Тема КФС/ІоТ-проєкту для SysML-моделювання
1	SysML-модель системи моніторингу температури в аудиторії
2	SysML-модель системи контролю вологості повітря
3	SysML-модель системи автоматичного освітлення
4	SysML-модель системи контролю рівня освітленості
5	SysML-модель системи виявлення руху
6	SysML-модель системи охоронної сигналізації
7	SysML-модель системи моніторингу мікроклімату теплиці
8	SysML-модель системи автоматичного поливу
9	SysML-модель системи контролю рівня води
10	SysML-модель системи виявлення витoku газу
11	SysML-модель системи моніторингу якості повітря
12	SysML-модель системи контролю доступу
13	SysML-модель системи керування вентиляцією
14	SysML-модель локальної системи вебмоніторингу
15	SysML-модель дистанційного керування реле через Wi-Fi
16	SysML-модель сенсорної мережі на основі ESP32
17	SysML-модель системи моніторингу енергоспоживання
18	SysML-модель системи контролю стану обладнання
19	SysML-модель системи моніторингу параметрів довкілля
20	Власна SysML-модель комплексного КФС/ІоТ-проєкту

Контрольні питання

1. Що таке SysML?
2. Для чого використовується SysML?
3. Чим SysML відрізняється від UML?
4. Що таке модель-орієнтоване проєктування?
5. Що таке системна модель?
6. Які системи доцільно описувати за допомогою SysML?
7. Чому SysML підходить для моделювання КФС?
8. Яке призначення діаграми вимог?
9. Що таке вимога до системи?
10. Чим функціональна вимога відрізняється від нефункціональної?
11. Що таке блок у SysML?
12. Яке призначення блокової діаграми?
13. Що показує діаграма внутрішньої структури блоків?
14. Що таке порт у SysML-моделі?
15. Що таке потік даних між блоками?
16. Яке призначення діаграми активності?
17. Що показує діаграма станів?
18. Що показує діаграма послідовності?
19. Для чого використовується параметрична діаграма?
20. Що таке обмеження в SysML?
21. Як у SysML можна описати сенсор?
22. Як у SysML можна описати виконавчий пристрій?
23. Як у SysML можна описати мікроконтролер ESP32?
24. Як у SysML можна описати Wi-Fi-модуль?
25. Як у SysML можна описати вебінтерфейс?

Лабораторне робота №9

Тема: Програмні засоби для інтелектуальних систем моделювання

Мета роботи: ознайомлення з програмними засобами, які використовуються для моделювання інтелектуальних систем у проєктах КФС/ІоТ, формування практичних навичок вибору середовища моделювання, побудови простої інтелектуальної моделі та аналізу результатів її роботи на основі сенсорних даних.

Теоретичні відомості

Інтелектуальна система — це система, яка здатна отримувати дані, аналізувати їх, приймати рішення та адаптувати свою поведінку залежно від зміни умов. У проєктах КФС/ІоТ такі системи використовуються для моніторингу стану об'єктів, прогнозування параметрів, виявлення аномалій, автоматичного керування та підтримки прийняття рішень.

Моделювання інтелектуальних систем дозволяє перевірити логіку роботи системи ще до її фізичної реалізації. Наприклад, можна змоделювати роботу системи «розумного будинку», яка аналізує температуру, вологість і освітленість, а потім приймає рішення про ввімкнення вентиляції, освітлення або сигналізації.

Програмні засоби для моделювання інтелектуальних систем поділяються на універсальні та спеціалізовані. Python використовується для аналізу даних, машинного навчання та побудови моделей прийняття рішень. MATLAB/Simulink зручний для математичного моделювання, імітації динамічних процесів та систем керування. Wokwi і Tinkercad Circuits застосовуються для моделювання мікроконтролерних пристроїв. Node-RED використовується для побудови логіки ІоТ-систем у вигляді потоків даних. AnyLogic, NetLogo та подібні середовища можуть застосовуватися для імітаційного та агентного моделювання.

У КФС/ІоТ інтелектуальна модель може виконувати різні функції: класифікацію стану системи, прогнозування показників сенсорів, виявлення аварійних ситуацій, оптимізацію режимів роботи або автоматичне керування виконавчими пристроями. Наприклад, система може аналізувати температуру й вологість повітря та визначати стан середовища як «норма», «попередження» або «аварія».

Практичне завдання

Завдання. Створити просту інтелектуальну модель для аналізу сенсорних даних у КФС/ІоТ-системі. Модель повинна отримувати значення температури, вологості та освітленості, визначати стан системи та формувати керувальне рішення.

Базовий приклад: модель системи мікроклімату приміщення, яка за значеннями температури та вологості визначає один із трьох станів: NORMAL, WARNING, ALARM.

Порядок виконання роботи

1. Обрати програмне середовище для моделювання.
2. Визначити об'єкт моделювання.
3. Сформуванати набір вхідних параметрів системи.
4. Визначити можливі стани системи.
5. Побудувати просту модель прийняття рішень.
6. Реалізувати модель у Python або іншому середовищі.
7. Виконати тестування моделі на різних вхідних даних.
8. Проаналізувати результати роботи моделі.
9. Зробити висновок щодо можливості використання моделі у КФС/ІоТ-проєкті.

Приклад вхідних параметрів

Параметр	Позначення	Опис
Температура	temperature	Значення температури повітря, °C
Вологість	humidity	Відносна вологість повітря, %
Освітленість	light	Умовний рівень освітленості
Стан системи	state	Норма, попередження або аварія

Приклад логіки прийняття рішень

Умова	Стан системи	Рішення
Температура до 28 °С і вологість до 70 %	NORMAL	Система працює у звичайному режимі
Температура від 28 до 32 °С або вологість від 70 до 80 %	WARNING	Вивести попередження
Температура понад 32 °С або вологість понад 80 %	ALARM	Увімкнути виконавчий пристрій

Індивідуальні завдання

№ варіанта	Індивідуальне завдання
1	Створити модель визначення стану температури в приміщенні.
2	Створити модель аналізу вологості повітря.
3	Реалізувати модель автоматичного керування освітленням.
4	Створити модель виявлення перегріву обладнання.
5	Реалізувати модель контролю рівня освітленості робочого місця.
6	Створити модель виявлення аварійного стану за двома сенсорами.
7	Реалізувати модель «розумної теплиці» з аналізом температури й вологості.
8	Створити модель автоматичного поливу за значенням вологості ґрунту.
9	Реалізувати модель виявлення витoku газу за умовним значенням сенсора.
10	Створити модель системи моніторингу якості повітря.
11	Реалізувати модель охоронної системи на основі датчика руху.
12	Створити модель контролю доступу за кількома умовами.
13	Реалізувати модель керування вентиляцією за температурою.
14	Створити модель визначення стану приміщення: норма, попередження, аварія.
15	Реалізувати модель прогнозування наступного значення температури.
16	Створити модель класифікації сенсорних даних за трьома класами.
17	Реалізувати модель виявлення аномальних значень сенсорів.
18	Створити модель керування реле на основі аналізу сенсорних параметрів.
19	Реалізувати модель прийняття рішення для системи енергомоніторингу.
20	Розробити власну інтелектуальну модель для КФС/ІoT-проєкту.

Контрольні питання

1. Що таке інтелектуальна система?
2. Для чого використовуються інтелектуальні системи у КФС/ІoT?
3. Що таке моделювання системи?
4. Яке призначення програмних засобів моделювання?
5. Чому моделювання важливе перед фізичною реалізацією системи?
6. Які програмні засоби можна використовувати для моделювання КФС/ІoT?
7. Для чого використовується Python у моделюванні інтелектуальних систем?
8. Для чого використовується MATLAB/Simulink?
9. Для чого може застосовуватися Node-RED?
10. Для чого використовується Wokwi?
11. Що таке сенсорні дані?
12. Які параметри можуть використовуватися як вхідні дані моделі?
13. Що таке вихідні параметри моделі?
14. Що таке модель прийняття рішень?
15. Що таке правило в інтелектуальній системі?
16. Що таке класифікація стану системи?
17. Які стани може мати інтелектуальна ІoT-система?
18. Що таке аварійний стан системи?
19. Як система може реагувати на аварійний стан?

20. Що таке порогове значення?
21. Як порогові значення використовуються в IoT-системах?
22. Що таке виявлення аномалій?
23. Які аномальні дані можуть виникати в сенсорній мережі?
24. Що таке прогнозування даних?
25. Як прогнозування може використовуватися в КФС?

Лабораторне робота №10

Тема: Моделювання IoT з використанням UML-діаграм на етапах архітектурного проектування
Мета роботи: ознайомлення з підходами до архітектурного проектування IoT-систем із використанням UML-діаграм, формування практичних навичок побудови моделей структури, поведінки та взаємодії компонентів IoT-рішення, а також підготовка архітектурної документації для подальшої реалізації проєкту на основі ESP32, сенсорів, мережевих технологій і програмного забезпечення.

Теоретичні відомості

Архітектурне проектування IoT-системи є одним із ключових етапів її розроблення. На цьому етапі визначається загальна структура майбутнього рішення, склад апаратних і програмних компонентів, способи їх взаємодії, канали передавання даних, логіка обробки інформації та роль користувача в системі. Для IoT-проєктів це особливо важливо, оскільки такі системи поєднують фізичні об'єкти, сенсори, мікроконтролери, мережеві протоколи, серверні компоненти, бази даних і користувацькі інтерфейси.

UML, або Unified Modeling Language, використовується для візуального опису системи на різних рівнях деталізації. За допомогою UML-діаграм можна показати, які функції виконує система, які компоненти входять до її складу, як між ними передаються дані, які стани має пристрій і як відбувається взаємодія користувача з IoT-рішенням.

У процесі архітектурного проектування IoT-систем доцільно використовувати кілька типів UML-діаграм. Діаграма варіантів використання дозволяє визначити основні функції системи та ролі користувачів. Діаграма компонентів показує загальну архітектуру рішення: сенсори, ESP32, мережевий модуль, сервер, базу даних і вебінтерфейс. Діаграма розгортання описує фізичне розміщення елементів системи та зв'язки між пристроями. Діаграма послідовності демонструє порядок обміну повідомленнями між компонентами. Діаграма активності описує алгоритм роботи системи, а діаграма станів дозволяє змоделювати зміну режимів роботи IoT-пристрою.

Типова IoT-архітектура складається з рівня сприйняття, мережевого рівня, рівня обробки даних і прикладного рівня. На рівні сприйняття працюють сенсори та виконавчі пристрої. Мережевий рівень відповідає за передавання даних через Wi-Fi, Bluetooth, Ethernet або інші технології. Рівень обробки даних забезпечує аналіз, збереження та прийняття рішень. Прикладний рівень надає користувачу доступ до інформації через вебінтерфейс, мобільний застосунок або панель моніторингу.

Використання UML на етапі архітектурного проектування дозволяє краще зрозуміти майбутню систему, виявити помилки ще до написання коду, узгодити взаємодію між апаратною і програмною частинами, а також підготувати основу для подальшого програмування, тестування та документування IoT-проєкту.

Практичне завдання

Завдання. Розробити архітектурну модель IoT-системи з використанням UML-діаграм. Система повинна містити сенсорний вузол на основі ESP32, один або кілька сенсорів, мережевий канал передавання даних, програмний компонент для обробки інформації та користувацький інтерфейс.

Базовий приклад: IoT-система моніторингу мікроклімату приміщення на основі ESP32, датчика DHT22, Wi-Fi-з'єднання, локального вебсервера та вебсторінки для перегляду температури, вологості й стану системи.

Порядок виконання роботи

1. Обрати тему IoT-проєкту.
2. Визначити мету створення системи.
3. Визначити основних користувачів системи.
4. Описати функціональні можливості IoT-рішення.
5. Визначити апаратні компоненти системи.
6. Визначити програмні компоненти системи.
7. Побудувати діаграму варіантів використання.

8. Побудувати діаграму компонентів.
9. Побудувати діаграму розгортання.
10. Побудувати діаграму послідовності для основного сценарію роботи.
11. Побудувати діаграму активності або діаграму станів.
12. Описати архітектуру системи за побудованими діаграмами.
13. Визначити потоки даних між компонентами.
14. Сформулювати висновок щодо доцільності використання UML у процесі архітектурного проектування.

Приклад опису IoT-системи. Проектується IoT-система моніторингу мікроклімату приміщення. Сенсор DHT22 вимірює температуру та вологість повітря. ESP32 зчитує ці дані, обробляє їх і передає користувачу через Wi-Fi. Користувач відкриває вебсторінку в браузері та переглядає поточні показники. Якщо температура перевищує встановлене порогове значення, система активує світлодіодну індикацію або релейний модуль.

Приклад архітектурних компонентів системи

Компонент	Призначення
ESP32	Центральний вузол обробки та передавання даних
DHT22	Вимірювання температури та вологості
LED або реле	Індикація або керування навантаженням
Wi-Fi	Передавання даних між ESP32 і користувачем
WebServer ESP32	Формування локальної вебсторінки
Користувач	Перегляд показників і контроль стану системи
Браузер	Засіб доступу до вебінтерфейсу

Приклад діаграми варіантів використання

Основні актори системи:

Актор	Роль
Користувач	Переглядає температуру, вологість і стан системи
Адміністратор	Налаштовує порогові значення
ESP32	Збирає, обробляє та передає дані
Сенсор	Надає виміряні значення фізичних параметрів

Індивідуальні завдання

№ варіанта	Тема IoT-системи для UML-моделювання
1	UML-модель IoT-системи моніторингу температури в аудиторії
2	UML-модель IoT-системи контролю вологості повітря
3	UML-модель системи автоматичного освітлення
4	UML-модель системи контролю рівня освітленості
5	UML-модель системи виявлення руху
6	UML-модель охоронної IoT-системи
7	UML-модель системи моніторингу мікроклімату теплиці
8	UML-модель системи автоматичного поливу
9	UML-модель системи контролю рівня води
10	UML-модель системи виявлення витoku газу
11	UML-модель системи моніторингу якості повітря
12	UML-модель системи контролю доступу
13	UML-модель системи керування вентиляцією
14	UML-модель локальної системи вебмоніторингу сенсорних даних
15	UML-модель дистанційного керування реле через Wi-Fi
16	UML-модель сенсорної мережі на основі ESP32

17	UML-модель системи моніторингу енергоспоживання
18	UML-модель системи контролю стану обладнання
19	UML-модель системи моніторингу параметрів довкілля
20	Власна UML-модель IoT-системи з описом архітектури

Контрольні питання

1. Що таке архітектурне проектування IoT-системи?
2. Для чого використовується UML?
3. Які UML-діаграми доцільно використовувати для IoT-проектів?
4. Що показує діаграма варіантів використання?
5. Що таке актор у UML?
6. Що таке варіант використання?
7. Що показує діаграма компонентів?
8. Що таке компонент у UML?
9. Для чого використовується діаграма розгортання?
10. Що таке вузол у діаграмі розгортання?
11. Що показує діаграма послідовності?
12. Що таке повідомлення на діаграмі послідовності?
13. Для чого використовується діаграма активності?
14. Що показує діаграма станів?
15. Які стани може мати IoT-пристрій?
16. Які основні рівні має типова IoT-архітектура?
17. Що таке сенсорний рівень IoT-системи?
18. Яке призначення мережевого рівня?
19. Що таке прикладний рівень IoT-системи?
20. Яку роль виконує ESP32 в IoT-архітектурі?
21. Які сенсори можна використовувати в IoT-проектах?
22. Для чого в IoT-системах використовуються виконавчі пристрої?
23. Як ESP32 може передавати дані користувачу?
24. Що таке локальний вебсервер на ESP32?
25. Які дані можуть передаватися між сенсором і мікроконтролером?

Лабораторна робота №11

Тема: Дослідження поведінки систем Інтернету речей за допомогою Color Petri Net Tools

Мета роботи: ознайомлення з принципами моделювання поведінки IoT-систем за допомогою кольорових мереж Петрі та середовища CPN Tools, формування практичних навичок побудови моделі сенсорного вузла, аналізу переходів між станами системи, виявлення можливих затримок, блокувань і помилок у процесах передавання та обробки даних.

Теоретичні відомості

Кольорові мережі Петрі, або Colored Petri Nets, є формальним засобом моделювання систем, у яких важливу роль відіграють паралельні процеси, обмін повідомленнями, синхронізація, черги, затримки та зміна станів. На відміну від класичних мереж Петрі, кольорові мережі дозволяють працювати не лише з однаковими маркерами, а й з маркерами, які мають певний тип, значення або структуру даних. Це дає змогу моделювати складні системи компактніше та ближче до реальної логіки їх роботи.

CPN Tools — це програмний засіб для редагування, моделювання та аналізу кольорових мереж Петрі. Середовище підтримує побудову моделі, перевірку синтаксису, симуляцію нетимованих і тимованих мереж, а також аналіз простору станів, зокрема властивостей обмеженості та живості моделі.

У проєктах Інтернету речей кольорові мережі Петрі можуть використовуватися для аналізу поведінки сенсорних вузлів, шлюзів, серверів, каналів зв'язку та виконавчих пристроїв. Наприклад, за допомогою CPN Tools можна змоделювати процес зчитування даних із сенсора, передавання повідомлення через Wi-Fi, обробку даних на контролері, прийняття рішення та активацію виконавчого пристрою.

Основними елементами мережі Петрі є позиції, переходи, дуги та маркери. Позиції відображають стани або умови системи. Переходи описують події, які можуть змінювати стан системи. Дуги визначають зв'язки між позиціями та переходами. Маркери показують поточний стан моделі. У кольорових мережах Петрі маркер може мати певний «колір», тобто тип або значення, наприклад тип сенсора, ідентифікатор вузла, числове значення температури або стан повідомлення.

Для IoT-систем важливими є такі властивості поведінки, як відсутність взаємних блокувань, своєчасна обробка повідомлень, правильна реакція на аварійні стани, стабільність передавання даних і можливість обслуговування кількох сенсорів одночасно. Кольорові мережі Петрі добре підходять для моделювання таких задач, оскільки вони орієнтовані на системи з паралельністю, комунікацією та синхронізацією.

Практичне завдання

Завдання. Побудувати модель поведінки IoT-системи в середовищі CPN Tools. Модель повинна описувати процес отримання даних із сенсорів, передавання їх до контролера, аналіз отриманих значень і формування керувальної дії.

Базовий приклад: IoT-система моніторингу мікроклімату на основі ESP32. Система отримує дані з сенсорів температури та вологості, передає їх через Wi-Fi до контролера, аналізує значення та у разі перевищення порогового рівня активує виконавчий пристрій.

Основні елементи моделі

Елемент моделі	Призначення
SensorReady	сенсор готовий до вимірювання
DataGenerated	дані сформовано сенсором
DataQueue	черга повідомлень для передавання
WiFiChannel	канал передавання даних
ControllerProcessing	обробка даних контролером
NormalState	параметри в межах норми
AlarmState	виявлено аварійний стан
ActuatorCommand	сформовано команду для виконавчого пристрою

Порядок виконання роботи

1. Відкрити CPN Tools або CPN IDE.
2. Створити нову модель кольорової мережі Петрі.
3. Визначити об'єкт моделювання — IoT-систему або сенсорний вузол.
4. Створити позиції, які описують основні стани системи.
5. Створити переходи, які описують події системи.
6. Додати дуги між позиціями та переходами.
7. Описати кольори маркерів і типи даних.
8. Задати початкове маркування моделі.
9. Додати умови спрацювання переходів.
10. Запустити симуляцію моделі.
11. Простежити зміну маркерів у позиціях.
12. Перевірити, чи не виникають блокування.
13. Проаналізувати, як система реагує на нормальні та аварійні значення.
14. За потреби додати затримку передавання або чергу повідомлень.
15. Підготувати висновок щодо поведінки змодельованої IoT-системи.

Індивідуальні завдання

№ варіанта	Індивідуальне завдання
1	Побудувати CPN-модель сенсора температури та контролера ESP32.
2	Змодельовати передавання даних від сенсора вологості до контролера.
3	Побудувати модель IoT-системи з двома сенсорами та одним контролером.
4	Змодельовати систему автоматичного освітлення з датчиком освітленості.
5	Побудувати модель виявлення перевищення температурного порогу.
6	Змодельовати чергу повідомлень від кількох сенсорів.
7	Побудувати модель Wi-Fi-каналу з можливою затримкою передавання.
8	Змодельовати втрату повідомлення під час передавання даних.
9	Побудувати модель повторного надсилання повідомлення після помилки.
10	Змодельовати систему «норма — попередження — аварія».
11	Побудувати модель активації реле при аварійному стані.
12	Змодельовати роботу сенсорної мережі з трьома вузлами.
13	Побудувати модель системи моніторингу якості повітря.
14	Змодельовати систему автоматичного поливу за значенням вологості ґрунту.
15	Побудувати модель охоронної системи з датчиком руху.
16	Змодельовати систему контролю доступу з двома станами: дозволено/заборонено.
17	Побудувати модель локального IoT-сервера, який приймає повідомлення від сенсорів.
18	Змодельовати перевантаження черги повідомлень у сенсорній мережі.
19	Побудувати CPN-модель системи керування вентиляцією.
20	Розробити власну CPN-модель IoT-системи з описом станів, переходів і маркерів.

Контрольні питання

1. Що таке мережа Петрі?
2. Що таке кольорова мережа Петрі?
3. Чим кольорова мережа Петрі відрізняється від класичної мережі Петрі?
4. Для чого використовується CPN Tools?
5. Які основні елементи має мережа Петрі?
6. Що таке позиція в мережі Петрі?
7. Що таке перехід у мережі Петрі?
8. Що таке дуга в мережі Петрі?

9. Що таке маркер?
10. Що означає колір маркера?
11. Для чого в CPN-моделі використовуються типи даних?
12. Що таке початкове маркування?
13. Що таке спрацювання переходу?
14. Яка умова потрібна для спрацювання переходу?
15. Як змінюється маркування після спрацювання переходу?
16. Для чого використовуються охоронні умови переходів?
17. Що таке симуляція моделі?
18. Що можна перевірити під час симуляції CPN-моделі?
19. Що таке простір станів?
20. Для чого виконується аналіз простору станів?
21. Що таке досяжний стан системи?
22. Що таке блокування в моделі?
23. Чому важливо перевіряти відсутність блокувань в IoT-системах?
24. Що таке живість моделі?
25. Що таке обмеженість моделі?

Лабораторна робота №12

Тема: Моделювання IoT з використанням темпоральної логіки на етапах їх архітектурного проектування

Мета роботи: вивчення можливостей структурно-функціонального, поведінкового, подієво-часового проектування системи IoT з використанням темпоральної логіки LTL та інструментального середовища SPIN / XSPIN.

Теоретичні відомості

Темпоральна логіка використовується для опису та перевірки поведінки систем у часі. Вона дозволяє визначати логічні формули, які описують, *що і коли* має відбуватися в системі.

Основні типи темпоральної логіки:

- **LTL (Linear Temporal Logic):** описує властивості вздовж окремих лінійних шляхів.
- **CTL (Computation Tree Logic):** враховує розгалуження у просторах станів.

Приклади LTL-формул:

- $G p$ – властивість p справедлива завжди (завжди в майбутньому)
- $F p$ – властивість p колись буде справедливою (в майбутньому)
- $p U q$ – p буде справжнім до тих пір, поки не настане q

Практичне завдання

Приклад.

1. Задано сценарій роботи системи розумного дому:

Якщо датчик руху зафіксував присутність у кімнаті, то через не більше ніж 3 секунди має ввімкнутись світло, і воно повинне залишатися увімкненим, поки присутність триває.

2. Сформулюємо модель у середовищі NuSMV або TLA+:

```
MODULE main
VAR
  motion : boolean;
  light  : boolean;
ASSIGN
  init(motion) := FALSE;
  init(light)  := FALSE;
  next(motion) := case
    motion : motion;
    TRUE   : {TRUE, FALSE};
  esac;
  next(light) := case
    motion : TRUE;
    !motion : FALSE;
  esac;
```

3. Формалізація властивостей у LTL:

- Світло вмикається, якщо є рух:
SPEC G (motion -> F[0:3] light)
- Світло не світить без присутності:
SPEC G (!motion -> !light)

4. Верифікація.

Індивідуальні завдання

1. Система розумного освітлення: Увімкнення світла при заході людини в кімнату, вимкнення — через 30 секунд після останнього руху.

2. Керування опаленням: Температура в кімнаті не повинна опускатися нижче 18°C. Якщо це сталося — вмикається обігрівач.

3. Розумний полив рослин: Полив активується при вологості ґрунту < 40%, не частіше ніж 1 раз на 2 години.

4. Система безпеки: У разі відкриття дверей вночі — спрацьовує сигналізація протягом 5 секунд.

5. Контроль доступу: Якщо ідентифікація не відбулась протягом 10 с після входу, двері блокуються.
6. Розумне вікно: Якщо температура зовні $> 30^{\circ}\text{C}$ і всередині $> 25^{\circ}\text{C}$ — вікно автоматично зачиняється.
7. Детектор диму: При виявленні диму має спрацювати сигнал і надіслатися повідомлення власнику.
8. Управління жалюзі: Жалюзі відкриваються на світанку, закриваються при заході сонця.
9. Контроль вологості в теплиці: Вентилятор вмикається, якщо вологість $> 80\%$.
10. Розумний холодильник: При тривалому відкритті дверей > 20 с — сигнал тривоги.
11. Система освітлення сходів: Освітлення активується при русі та згасає через 15 с без активності.
12. Автоматичне годування тварин: Годівниця активується рівно о 8:00 та 18:00.
13. Моніторинг вологи в погребі: При перевищенні вологості $> 70\%$ — включити осушувач повітря.
14. Пожежна сигналізація: При одночасному виявленні високої температури та диму — активація сирени.
15. Розумне освітлення класу: При освітленості < 300 лк — автоматичне увімкнення ламп.
16. Система поливу в теплиці: Враховується вологість, температура і час доби (не поливати вночі).
17. Система контролю CO_2 : Якщо рівень $\text{CO}_2 > 1000$ ppm — ввімкнути вентиляцію.
18. Розумне дзеркало: Вмикається при наближенні користувача, гасне через 10 с.+
19. Зовнішнє освітлення подвір'я: Активується тільки у вечірній час при виявленні руху.
20. Безконтактне відкриття дверей: У разі наявності авторизованого Bluetooth-сигналу — відкрити замок.
21. Сигналізація для вікна: При виявленні відкриття вікна у режимі охорони — активувати тривогу.
22. Розумна кавоварка: Готує каву щодня о 7:00, тільки якщо користувач знаходиться в будинку.
23. Вентиляція в серверній: Система активує охолодження при температурі $> 27^{\circ}\text{C}$.
24. Система паркінгу: Якщо вільних місць < 5 — показати повідомлення на дисплеї.
25. Контроль води у баку: При зниженні рівня води $< 10\%$ — ввімкнути насос.
26. Індивідуальний таймер освітлення: Світло вмикається після заходу та працює рівно 4 години.
27. Контроль температури в інкубаторі: Температура підтримується в межах $37\text{--}38^{\circ}\text{C}$.
28. Моніторинг забруднення повітря: Якщо $\text{AQI} > 150$ — надіслати push-сповіщення користувачу.
29. Розумна зарядка: Зарядка телефону припиняється при досягненні 100%.
30. Система нагадування про ліки: Через кожні 8 годин надсилати звукове сповіщення.
31. Виявлення падіння для літніх людей: У разі виявлення падіння — надіслати сигнал родичу.
32. Керування тепловою підлогою:
33. Вмикається при температурі нижче 22°C з 6:00 до 22:00.
34. Система збереження енергії: Вимикає неактивні пристрої після 5 хвилин бездіяльності.
35. Індикатор споживання електроенергії: При перевищенні 3 кВт — надіслати повідомлення.
36. Розумний душ: Температура регулюється автоматично залежно від уподобань користувача.
37. Автоматичне заповнення басейну: Якщо рівень води $< 80\%$ — відкрити клапан.
38. IoT-система вулика: При падінні температури $< 25^{\circ}\text{C}$ — вмикати нагрівач.
39. Розумний одяг: Повідомлення при перевищенні температури тіла $> 38^{\circ}\text{C}$.

40. Система керування жалюзі: Жалюзі автоматично орієнтуються залежно від сонячного світла.

41. Розумне виявлення шуму: Якщо шум > 80 дБ після 22:00 — повідомити власника.

Контрольні питання

1. Що таке темпоральна логіка?
2. Для чого використовується темпоральна логіка у проектуванні IoT-систем?
3. Чим темпоральна логіка відрізняється від класичної логіки висловлювань?
4. Що означає поняття «часова властивість системи»?
5. Які процеси IoT-системи доцільно описувати за допомогою темпоральної логіки?
6. Що таке архітектурне проектування IoT-системи?
7. Чому на етапі архітектурного проектування важливо враховувати поведінку системи в часі?
8. Які часові обмеження можуть виникати в IoT-системах?
9. Що таке стан системи у темпоральній логіці?
10. Що таке перехід між станами системи?
11. Як можна описати зміну станів IoT-пристрою в часі?
12. Що таке подія в IoT-системі?
13. Які події можуть виникати в системах Інтернету речей?
14. Що таке послідовність станів системи?
15. Як темпоральна логіка допомагає описати сценарії роботи IoT-системи?
16. Що таке лінійна темпоральна логіка?
17. Для чого використовується LTL?
18. Що таке обчислювальна деревоподібна логіка?
19. Для чого використовується STL?
20. Чим LTL відрізняється від STL?
21. Що означає оператор G у темпоральній логіці?
22. Що означає оператор F у темпоральній логіці?
23. Що означає оператор X у темпоральній логіці?
24. Що означає оператор U у темпоральній логіці?

Лабораторне заняття №13

Тема: Побудова інформаційних моделей на основі IoT-технологій пристроїв
Мета роботи: набуття навичок роботи з генератором коду для модифікації та реалізації інформаційних моделей пристрою на конкретних платформах

Теоретичні відомості

Інформаційна модель IoT-пристрою — це формалізований опис структури, функцій, даних та взаємодій пристрою в контексті IoT-архітектури. Такі моделі дозволяють:

- описати функціональну логіку системи;
- визначити вхідні/вихідні сигнали;
- змодельовати потоки даних;
- спроектувати взаємодію компонентів.

Основні типи моделей:

- UML-діаграми класів (опис структури);
- діаграми активностей (поведінка пристрою);
- діаграми послідовностей (передача повідомлень між об'єктами);
- DFD (Data Flow Diagrams) — моделі потоків даних;
- ER-моделі — у випадку з базами даних IoT-платформи.

Практичне завдання

Приклад.

1. Вибір IoT-пристрою для моделювання:

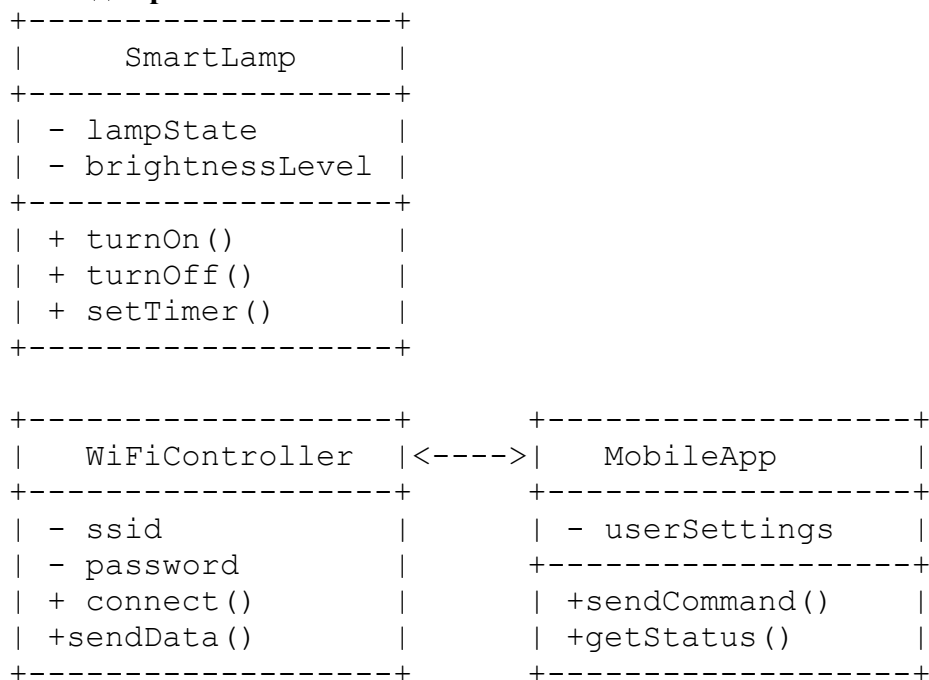
Розумна лампа з Wi-Fi керуванням

2. Опис функцій пристрою:

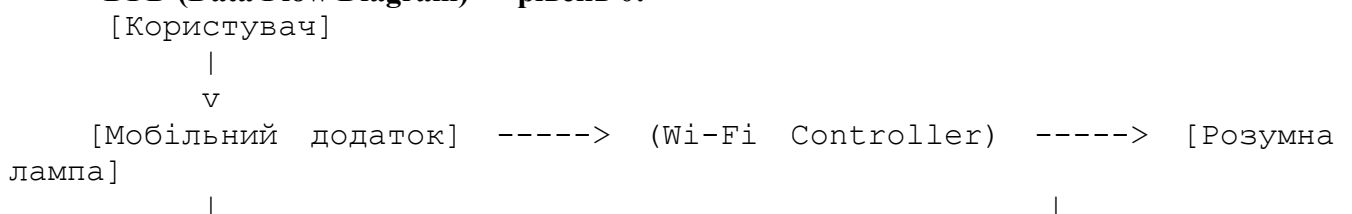
- Приймає команду з мобільного застосунку (ввімкнення / вимкнення)
- Передає статус (ввімкнена/вимкнена) до хмарного сервера.
- Може працювати за таймером або за розкладом.
- Підтримує протоколи MQTT або HTTP.

3. Побудова інформаційної моделі:

UML-діаграма класів:



DFD (Data Flow Diagram) — рівень 0:



Діаграма послідовностей:

Користувач → Мобільний застосунок: Увімкнути лампу
 Мобільний застосунок → Wi-Fi модуль: Надіслати команду MQTT
 Wi-Fi модуль → Лампа: Ввімкнути
 Лампа → Wi-Fi модуль: Статус = Ввімкнено
 Wi-Fi модуль → Хмара: Оновити статус

4. Аналіз моделі:

Поле даних	Тип	Призначення
command	string	"ON" або "OFF"
status	string	"Lamp is ON" або "Lamp is OFF"
timestamp	datetime	Час команди
device_id	UUID / string	Ідентифікатор пристрою

5. Висновок:

Було побудовано інформаційну модель IoT-пристрою — розумної лампи з Wi-Fi керуванням. Структура моделі охоплює її класи, потоки даних та логіку взаємодії з користувачем і хмарою. Це дозволяє ефективно проектувати подібні пристрої з урахуванням майбутнього програмного забезпечення та комунікаційних вимог.

Якщо датчик руху зафіксував присутність у кімнаті, то через не більше ніж 3 секунди має ввімкнутись світло, і воно повинне залишатися увімкненим, поки присутність триває.

Індивідуальні завдання

№ Варіанту	Індивідуальне завдання
1	Побудова інформаційної моделі розумної лампи з Wi-Fi керуванням
2	Моделювання IoT-системи поливу з сенсором вологості
3	Інформаційна модель розумного дверного замка з Bluetooth-модулем
4	Моделювання розумного термометра з передачею даних у хмару
5	Побудова моделі трекера для домашніх тварин із GPS та GSM
6	Інформаційна модель інтелектуального холодильника з моніторингом температури
7	Моделювання системи освітлення, яка активується за розкладом
8	Розробка моделі розумної пральної машини з підтримкою IoT
9	Модель IoT-пристрою керування жалюзі за рівнем освітленості
10	Побудова інформаційної моделі системи безпеки з датчиком руху
11	Моделювання системи моніторингу якості повітря (CO ₂ , пил, вологість)
12	Модель IoT-пристрою для контролю температури ґрунту
13	Побудова інформаційної моделі розумного дзеркала
14	Модель розумного вентилятора з автоматичним включенням
15	Моделювання системи визначення рівня води у баку
16	Інформаційна модель системи моніторингу пульсу та температури тіла
17	Модель IoT-пристрою для керування системою опалення
18	Моделювання розумної плити з контролем температури
19	Побудова інформаційної моделі смарт-годинника з функцією моніторингу активності
20	Інформаційна модель системи контролю відкриття вікон
21	Модель системи керування зарядкою електромобіля
22	Моделювання системи виявлення протікання води

23	Інформаційна модель системи поливу на основі погодних даних		
24	Модель автоматичної системи освітлення в офісі		
25	Побудова інформаційної моделі розумного кулера для води		
26	Моделювання системи вентиляції теплиці на основі датчиків температури		
27	Інформаційна модель пристрою контролю енергоспоживання		
28	Модель IoT-системи сигналізації для гаражу		
29	Моделювання IoT-пристрою для управління паркінгом		
30	Побудова моделі пристрою вимірювання рівня шуму		
31	Модель інтелектуальної кавоварки з керуванням через додаток		
32	Інформаційна модель IoT-брелока з функцією пошуку ключів		
33	Моделювання пристрою керування теплою підлогою		
34	Модель IoT-пристрою моніторингу стану кімнатних рослин		
35	Побудова інформаційної моделі смарт-перемикача з мобільним керуванням		
36	Моделювання системи контролю температури у серверній кімнаті		
37	Інформаційна модель автоматичного сортувальника сміття		
38	Модель пристрою для моніторингу якості води		
39	Моделювання IoT-системи керування освітленням у класі		
40	Інформаційна модель системи нагадування про прийом ліків		

Контрольні питання

1. Що таке інформаційна модель?
2. Для чого створюються інформаційні моделі в IoT-системах?
3. Що таке IoT-пристрій?
4. Які основні компоненти має типовий IoT-пристрій?
5. Яку роль у побудові інформаційної моделі виконує сенсор?
6. Яку роль виконує мікроконтролер в IoT-пристрої?
7. Що таке виконавчий пристрій в IoT-системі?
8. Які дані можуть збирати IoT-пристрої?
9. Що таке сенсорні дані?
10. Чим відрізняються аналогові та цифрові дані?
11. Як фізичний параметр перетворюється на цифрові дані?
12. Що таке об'єкт моделювання?
13. Що таке параметр інформаційної моделі?
14. Що таке атрибут IoT-пристрою?
15. Що таке стан IoT-пристрою?
16. Як можна описати стан сенсорного вузла?
17. Що таке подія в інформаційній моделі IoT-системи?
18. Які події можуть виникати в IoT-пристроях?
19. Що таке потік даних в IoT-системі?
20. Як описується передавання даних між IoT-пристроями?
21. Які рівні має типова архітектура IoT-системи?
22. Що таке сенсорний рівень IoT-системи?
23. Яке призначення мережевого рівня IoT-системи?
24. Що таке рівень обробки даних?
- 25.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ESP-IDF Programming Guide: Wi-Fi Driver. Espressif Systems. 2025. URL: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-guides/wifi.html> (дата звернення: 20.05.2026).
2. ESP-IDF Programming Guide: lwIP. Espressif Systems. 2024. URL: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-guides/lwip.html> (дата звернення: 20.05.2026).
3. Arduino ESP32 Documentation: Wi-Fi API. Espressif Systems. 2025. URL: <https://docs.espressif.com/projects/arduino-esp32/en/latest/api/wifi.html> (дата звернення: 20.05.2026).
4. Wi-Fi Overview. Arduino Documentation. 2024. URL: <https://docs.arduino.cc/language-reference/en/functions/wifi/overview/> (дата звернення: 20.05.2026).
5. Chui K. T., Gupta B. B., Liu J., Arya V., Nedjah N. A Survey of Internet of Things and Cyber-Physical Systems: Standards, Algorithms, Applications, Security, Challenges, and Future Directions. **Information**. 2023. Vol. 14, No. 7. URL: <https://www.mdpi.com/2078-2489/14/7/388> (дата звернення: 20.05.2026).
6. Mansour M., Gamal A., Ahmed A. I., Said L. A., Elbaz A., Herencsar N., Soltan A. Internet of Things: A Comprehensive Overview on Protocols, Architectures, Technologies, Simulation Tools and Future Directions. **Energies**. 2023. Vol. 16, No. 8. URL: <https://www.mdpi.com/1996-1073/16/8/3465> (дата звернення: 20.05.2026).
7. Krichen M., Mihoub A., Alzahrani M. Y., Adoni W. Y. H., Nahhal T. A Survey on Formal Verification and Validation Techniques for Internet of Things. **Applied Sciences**. 2023. Vol. 13, No. 14. URL: <https://www.mdpi.com/2076-3417/13/14/8122> (дата звернення: 20.05.2026).
8. Oks S. J., Jalowski M., Fritzsche A., Mösllein K. M. Cyber-Physical Systems in the Context of Industry 4.0. **Information Systems Frontiers**. 2022. URL: <https://link.springer.com/article/10.1007/s10796-022-10252-x> (дата звернення: 20.05.2026).
9. Jadidi N., Varmazyar M. A Survey of Cyber-Physical Systems Applications (2017–2022). **Springer Reference**. 2023. URL: https://link.springer.com/rwe/10.1007/978-3-030-97940-9_145 (дата звернення: 20.05.2026).
10. Abdelmalak M., Venkataramanan V., Macwan R. A Survey of Cyber-Physical Power System Modeling Methods for Future Energy Systems. **IEEE Access**. 2022. Vol. 10. P. 99875–99896. URL: <https://ieeexplore.ieee.org/document/9893997> (дата звернення: 20.05.2026).
11. Abdelmalak M., Bhusal N., Gautam M., Ben-Idris M. Cyber-Physical Power System Layers: Classification, Characterization, and Interactions. 2023. URL: <https://arxiv.org/abs/2302.06772> (дата звернення: 20.05.2026).
12. Sebestyen H. A Literature Review on Security in the Internet of Things. **Computers**. 2025. Vol. 14, No. 2. URL: <https://www.mdpi.com/2073-431X/14/2/61> (дата звернення: 20.05.2026).
13. Dritsas E. A Survey on Cybersecurity in IoT. **Future Internet**. 2025. Vol. 17, No.

1. URL: <https://www.mdpi.com/1999-5903/17/1/30> (дата звернення: 20.05.2026).
14. Mustafa R. та ін. A Cross-Layer Secure and Energy-Efficient Framework for Internet of Things Networks: A Comprehensive Survey. **Sensors**. 2024. Vol. 24, No. 22. URL: <https://www.mdpi.com/1424-8220/24/22/7209> (дата звернення: 20.05.2026).
15. Ahsan M. S. та ін. A Comprehensive Survey on the Requirements, Open Challenges and Future Directions of Access Control Models in IoT. 2025. URL: <https://www.mdpi.com/2624-831X/6/1/9> (дата звернення: 20.05.2026).
16. Al Asif M. R., Hasan K. F., Islam M. Z., Khondoker R. STRIDE-based Cyber Security Threat Modeling for IoT-enabled Precision Agriculture Systems. 2022. URL: <https://arxiv.org/abs/2201.09493> (дата звернення: 20.05.2026).
17. Wheaton J. S., Herber D. R. Digital Requirements Engineering with an INCOSE-derived SysML Meta-model. 2024. URL: <https://arxiv.org/abs/2401.16330> (дата звернення: 20.05.2026).
18. Systems Modeling Language Specification: SysML v2.0. Object Management Group. 2025. URL: <https://www.omg.org/spec/SysML/2.0/About-SysML> (дата звернення: 20.05.2026).
19. Berti A., van der Aalst W. M. P. CPN-Py: A Python-Based Tool for Modeling and Analyzing Colored Petri Nets. 2025. URL: <https://arxiv.org/abs/2506.12238> (дата звернення: 20.05.2026).
20. Steinsland V., Kristensen L. M., Zhang S. Modelling and Validation of Power Electronics Converter Systems using Coloured Petri Nets. 2022. URL: <https://arxiv.org/abs/2212.06781> (дата звернення: 20.05.2026).

I 75 **IoT-технології для кіберфізичних систем:** методичні вказівки до лабораторних занять для здобувачів другого (магістерського) рівня вищої освіти освітньої програми «Комп’ютерна інженерія» галузь знань F Інформаційні технології спеціальності F7 Комп’ютерна інженерія денної та заочної форм навчання / уклад. С.В. Гринюк, Л.М. Конкевич Луцьк: ЛНТУ, 2026. 48 с..

Комп’ютерний набір: С.В. Гринюк
Редактор: С.В. Гринюк

Підп. до друку _____ 2026 р.
Формат 60x84/16. Папір офс. Гарнітура Таймс.
Ум. друк. арк. ____ Тираж ____ прим. Зам. _____

Відділ іміджу та промоції
Луцького національного технічного університету
43018 м. Луцьк, вул. Львівська, 75