



Технології розробки вбудованих IoT систем

Сертифікатна програма

«Інженерія вбудованих систем та інтернет речей за участі компанії GlobalLogic»

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	Перший (бакалаврський)
Галузь знань	F (12) Інформаційні технології
Спеціальність	F2 (121) Інженерія програмного забезпечення; F7 (123) Комп'ютерна інженерія, F6 (126) Інформаційні системи та технології
Освітня програма	Комп'ютерні системи та мережі Інженерія програмного забезпечення комп'ютерних систем Інтегровані інформаційні системи Інформаційні управляючі системи та технології Інформаційне забезпечення робототехнічних систем Інженерія програмного забезпечення інформаційних систем Інженерія програмного забезпечення комп'ютерних систем
Статус дисципліни	Вибіркова
Форма навчання	очна (денна)
Рік підготовки, семестр	4 курс, весняний
Обсяг дисципліни	4 кредитів/120 год., лекцій 36 годин, лаб. роб. 18 год., СРС 66 год.
Семестровий контроль/ контрольні заходи	Залік, лабораторні роботи, МКР
Розклад занять	Згідно розкладу на осінній семестр поточного навчального року за адресою https://schedule.kpi.ua
Мова викладання	Українська
Інформація про керівника курсу / викладачів	Лектори: асистент кафедри ОТ Каплунов Артем Володимирович, kaplunov.artem@iit.kpi.ua Ментори: QA Engineer компанії Global Logic Таранюк Вікторія Анатоліївна Лабораторні: Каплунов Артем Володимирович
Розміщення курсу	Навчальний курс розміщений на платформі дистанційного навчання «Сікорський» в середовищі Google Workspace for Education https://classroom.google.com/c/NjU1NDI2MzI1NjIw?cjc=zotp7sj

1. Опис навчальної дисципліни, її мета, предмет вивчення та результати навчання

Вбудовані системи IoT поєднують у собі величезну кількість технологічних проблем, пов'язаних із вбудованим програмним забезпеченням, вбудованими обчисленнями, протоколами зв'язку з низьким енергоспоживанням, вбудованими протоколами безпеки, вбудованим збором енергії, обробкою даних датчиків поблизу, вбудованими методами штучного інтелекту та інтеграцією в хмарні та периферійні обчислювальні системи.

На сьогодні застосування вбудованих систем є повсюдним та все більше поширюється, вбудовані системи наразі це невід'ємна частина Інтернету речей (IoT) та Інтернету всього (IoE). Комерційна, урядова, військова, медична, освітня, культурна та інші інфраструктури сьогодні пов'язані в єдину мережу IoT і є конгломератом, як автономних та і взаємозв'язаних вбудованих систем.

Стрімкий розвиток сучасних технологій IoT потребує відповідної специфіки знань та вмінь від сучасних фахівців в області комп'ютерної інженерії, програмування та загалом інформаційних систем.

Цікаві проєкти інноваційних методів навчання вже реалізуються і плануються до використання в курсі «Технології розроблення вбудованих систем IoT», це – «Лабораторія по віддаленій прошивці плат». Для ініціативних студентів є купа цікавих задачок, де можна проявити себе та отримати сучасний досвід розробки та програмування для вбудованих систем. А компанія GL забезпечила сертифікатну програму базовими комплектами «заліза» для Ваших творчих досягнень.

Вивчення дисципліни «Технології розробки вбудованих IoT систем» підсилює наступні загальні та фахові компетенції:

Для спеціальності F2 (121) Інженерія програмного забезпечення

- Здатність до абстрактного мислення, аналізу та синтезу
- Здатність застосовувати знання у практичних ситуаціях
- Здатність вчитися і оволодівати сучасними знаннями
- Здатність працювати в команді
- Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення
- Здатність формулювати та забезпечувати вимоги щодо якості програмного забезпечення у відповідності з вимогами замовника, технічним завданням та стандартами
- Здатність дотримуватися специфікацій, стандартів, правил і рекомендацій в професійній галузі при реалізації процесів життєвого циклу
- Здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя
- Здатність до алгоритмічного та логічного мислення
- Здатність розробляти та використовувати мережні технології
- Здатність розробляти та використовувати програмне забезпечення для високопродуктивних комп'ютерних систем
- Здатність використовувати принципи побудови, функціонування та узагальнену структуру мікропроцесорних систем та особливості програмного забезпечення мікропроцесорних систем та мікроконтролерів

Для спеціальності F7 (123) «Комп'ютерна інженерія»

- Здатність до абстрактного мислення, аналізу і синтезу
- Здатність генерувати нові ідеї (креативність)
- Здатність виявляти, ставити та вирішувати проблеми
- Здатність приймати обґрунтовані рішення

- Здатність до визначення технічних характеристик, конструктивних особливостей, застосування і експлуатації програмних, програмно-технічних засобів, комп'ютерних систем та мереж різного призначення
- Здатність працювати в команді
- Здатність застосовувати законодавчу та нормативно правову базу, а також державні та міжнародні вимоги, практики і стандарти з метою здійснення професійної діяльності в галузі комп'ютерної інженерії.
- Здатність використовувати сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення.
- Здатність використовувати засоби і системи автоматизації проектування до розроблення компонентів комп'ютерних систем та мереж, Інтернет додатків, кіберфізичних систем тощо.
- Здатність проектувати системи та їхні компоненти з урахуванням усіх аспектів їх життєвого циклу та поставленої задачі, включаючи створення, налаштування, експлуатацію, технічне обслуговування та утилізацію.
- Здатність проектувати, впроваджувати та обслуговувати високопродуктивні паралельні та розподілені комп'ютерні системи та їх складові з використання ПЛІС модулів і систем автоматизованого проектування
- Здатність розробляти, адаптувати, використовувати програмно-технічні засоби для збирання, оброблення та інтелектуального аналізу даних в комп'ютерних системах: паралельних, розподілених, хмарних, безпечних, інтелектуальних, розумних, Інтернет речей, зокрема, з використанням штучного інтелекту.

Для спеціальності F6 (126) Інформаційні системи та технології

- Здатність до абстрактного мислення, аналізу та синтезу
- Здатність застосовувати знання у практичних ситуаціях
- Здатність розробляти та управляти проектами
- Здатність аналізувати об'єкт проектування або функціонування та його предметну область
- Здатність застосовувати стандарти в області інформаційних систем та технологій при розробці функціональних профілів, побудові та інтеграції систем, продуктів, сервісів і елементів інфраструктури організації
- Здатність до проектування, розробки, налагодження та вдосконалення системного, комунікаційного та програмно апаратного забезпечення інформаційних систем та технологій, Інтернету речей (IoT), комп'ютерно інтегрованих систем та системної мережної структури, управління ними
- Здатність управляти та користуватися сучасними інформаційно-комунікаційними системами та технологіями (у тому числі такими, що базуються на використанні Інтернет).
- Здатність розробляти структури робототехнічних систем та проектувати функціональні пристрої роботів та робототехнологічних комплексів
- Здатність використовувати знання основ електроніки і мікросхемотехніки, сучасної елементної бази при проектуванні мікропроцесорних пристроїв робототехнічних систем
- Здатність формувати нові конкурентоспроможні ідеї й реалізовувати їх у проектах (стартапах)

У відповідності до вищезазначеного, підсилені загальні та фахові компетенції нададуть наступні результати навчання:

- знання фундаментальних концепцій, парадигм і основних принципів функціонування вбудованих систем IoT;
- здатність до використання теоретичних, логічних та арифметичних основ для розробки драйверів та програмного забезпечення для вбудованих систем і вміння їх застосовувати при вирішенні професійних завдань;
- здатність розробляти драйвери для окремих компонентів вбудованих систем IoT, в тому числі, з використанням сучасних систем автоматизації проектування;
- здатність розробки та використання драйверів елементів архітектурозалежного рівня на основі знання загальних принципів організації та функціонування вбудованих систем;

- здатність брати участь у командній роботі по проектуванню апаратного та програмного забезпечення вбудованих систем IoT;
- здатність формувати та забезпечувати вимоги до вбудованих систем у відповідності з вимогами замовника, технічним завданням та стандартами.
- Практичні навички в області програмування для мікроконтролерів:
 - програмування на мові C, C++ для мікроконтролерів STM32, ESM32;
 - розробки програмного забезпечення для МК на платі Global Logic Startet Kit STM32;
- практичні навички в області Linux Kernel:
 - розвертання процесорних ядер за допомогою Toolchains.
 - розроблення програмного забезпечення в ОС Linux на платі та віртуальній машині для лінійки процесорів ARM орієнтованих на встановлення ОС; QA:
 - тестування розроблених модулів;
 - розроблення автотестів для вбудованих систем на мові Python;
 - використання системи контролю версій Git для роботи в команді; – генерації та аналізу мережного трафіку в інфраструктурі IoT;
 - оцінювання обчислювальних можливостей вбудованих систем IoT.
- незважаючи на поглиблену складову HW, студенти отримують основні знання та навички з різних рівнів інфраструктури IoT. Як то організація та розробка системи кінцевих датчиків та сенсорів, мережного оточення та інтернет інфраструктури для обміну даними між пристроями IoT та серверами обробки даних, архітектури Edge та хмарної інфраструктури аналітики даних – хмар IoT.

2. Пререквізити та постреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою)

Для успішного засвоєння дисципліни «Технології розробки вбудованих IoT систем» студентам доцільно мати базові теоретичні та лабораторні знання, необхідні для проектування, складання та налагодження апаратно-програмних електронних систем, які набуті під час вивчення базових та професійно орієнтованих дисциплін та охоплюють такі напрями:

- схемотехніка та базові знання з апаратних засобів вбудованих систем
- програмування та алгоритмічне мислення
- базові знання з архітектури комп'ютерів
- основи програмування мовою C для процесорного ядра
- операційні системи (Linux), системи контролю версій (Git)
- комп'ютерні мережі та принципи обміну даними
- основи системного програмування та розробки драйверів
- знання методів командної роботи, управління проєктами

Знання та навички, отримані під час вивчення дисципліни «Технології розробки вбудованих IoT систем», можуть бути застосовані під час вирішення навчальних та лабораторних завдань, що виникають під час:

- тестування та налагодження апаратно-програмних систем
- забезпечення надійності та функціональної безпеки вбудованих пристроїв
- автоматизації процесів тестування та використання відповідних інструментів
- розробки і верифікації прототипів інтелектуальних систем
- організації командної роботи у сфері вбудованого програмного забезпечення
- інтеграції компонентів у складних проєктах (апаратні модулі, прошивки, мережеві інтерфейси)
- розробки та супроводу IT-проєктів у галузях, пов'язаних з Інтернетом речей (IoT), комп'ютерними мережами, інфраструктурними рішеннями

3. Зміст навчальної дисципліни

Вступ

Розділ 1. Вступ до IoT та Smart City

Лекція 1. Основні поняття IoT та Smart City

Тема 1.1. Поняття Інтернету речей (IoT)

Тема 1.2. Концепція Smart City

Тема 1.3. Типові сфери застосування IoT у Smart City

Тема 1.4. Приклади простих Smart City рішень

Лекція 2. Архітектура IoT-систем

Тема 1.5. Загальна архітектура IoT-систем

Тема 1.6. Сенсорний рівень

Тема 1.7. Edge-рівень

Тема 1.8. Серверний та клієнтський рівні

Тема 1.9. Потік даних у Smart City системі

Лекція 3. Огляд технологій систем IoT

Тема 1.10. STM32 як IoT-пристрій

Тема 1.11. MQTT як протокол обміну

Тема 1.12. Edge-пристрої (Raspberry Pi)

Тема 1.13. Сервер, база даних, візуалізація

Розділ 2. Технології IoT систем

Лекція 4. Контейнеризація IoT-проєкту

Тема 4.1. Основи Docker

Тема 4.2. Docker Compose

Тема 4.3. Запуск IoT-проєкту в контейнерах

Лекція 5. MQTT у IoT

Тема 4.4. Принцип роботи MQTT

Тема 4.5. Broker, publisher, subscriber

Тема 4.6. Базові поняття технології QoS

Лекція 6. Взаємодія компонентів IoT системи

Потоки даних між модулями

Взаємодія STM32 – Edge – сервер

Розділ 3. Рівень збору даних: STM32 та сенсори

Лекція 7. STM32 як платформа IoT

Тема 3.1. Архітектура STM32

Тема 3.2. Плата STM32 Discovery

Роль мікроконтролера в IoT-системі

Лекція 8. Сенсори та інтерфейси

Тема 3.2. Типи сенсорів

Тема 3.3. ADC

Тема 3.4. I2C, SPI, UART

Тема 3.5. Підключення сенсорів до STM32

Лекція 6. Програмування STM32 мовою C

Тема 3.6. STM32CubeIDE

Тема 3.7. Структура embedded-проєкту

Тема 3.8. Робота з GPIO

Тема 3.9. Зчитування даних із сенсорів

Лекція 7. Передача даних з STM32

Тема 3.10. Формування пакетів даних

Тема 3.11. Передача даних на Edge-рівень

Тема 3.12. Обробка помилок передачі

Розділ 4. Edge-рівень обробки даних

Лекція 8. Edge Computing у IoT

Тема 4.1. Призначення Edge-рівня

Тема 4.2. Raspberry Pi як Edge-пристрій

Тема 4.3. Прийом даних від STM32

Лекція 9. Основи обробки даних на Edge

Тема 4.4. Попередня обробка даних

Тема 4.5. Агрегація

Тема 4.6. Перевірка коректності даних

Лекція 10. Збереження даних

Тема 4.7. Основи баз даних

Тема 4.8. PostgreSQL

Тема 4.9. Збереження сенсорних даних

Розділ 5. Візуалізація даних та інтеграція компонентів

Лекція 14. Візуалізація даних

Тема 5.1. Основи геовізуалізації в IoT-системах

Тема 5.2. Візуалізація даних Smart City на Google Maps

Тема 5.3. Оновлення та відображення даних у реальному часі

Лекція 15. Smart City проєкт

Тема 5.4. Інтеграція компонентів

Тема 5.5. Архітектура базового Smart City проєкту

Тема 5.6. Демонстрація системи

Розділ 5. Організація проєктної діяльності

Лекція 16. Організація командної роботи

Тема 5.7. Організація роботи над проєктом за технологією SCRUM

Тема 5.8. Розподіл ролей в команді

Тема 5.9. Планування роботи

Лекція 17. Тестування та налагодження

Тема 5.10. Збірка та налагодження проєкту

Тема 5.11. Тестування IoT-системи

Тема 5.12. Типові помилки

Лекція 18. Підсумкова лекція

Тема 5.13. Завершення Smart City проєкту

Тема 5.14. Подальші напрями навчання

4. Навчальні ресурси та матеріали

4.1. Базова література

1. Жураковський, Б. Ю. Технології інтернету речей. Навчальний посібник [Електронний ресурс] : навчальний посібник для здобувачів ступеня бакалавра за освітньою програмою «Інформаційне забезпечення робототехнічних систем» за спеціальністю 126 «Інформаційні системи та технології» / Б. Ю. Жураковський, І. О. Зенів ; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 5,1 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2021. – 271 с. – Назва з екрана. <https://ela.kpi.ua/handle/123456789/42078>

2. Тестування та контроль якості (QA) вбудованих систем: лабораторний практикум [Electronic resource]: Навчальний посібник для здобувачів ступеня бакалавра за спеціальністю 123 «Комп'ютерна інженерія» / уклад.: В. А. Таранюк, І. А. Клименко, В. В. Ткаченко, О. О. Писарчук. – Київ : КПІ ім. Ігоря Сікорського, 2022. – 75 с. Гриф надано Методичною радою КПІ ім. Ігоря Сікорського (протокол № 1 від 02.09.2022 р.). <https://ela.kpi.ua/handle/123456789/52162>.

3. Архітектура комп'ютерів 3. Мікропроцесорні засоби. Частина 2. Програмування для мікроконтролерів STM32. Теорія та лабораторний практикум. [Electronic resource]: Навчальний посібник для здобувачів ступеня бакалавра за спеціальністю 123 «Комп'ютерна інженерія» / уклад.: І. А. Клименко, В. А. Таранюк, В. В. Ткаченко, Каплунов А. В. – Київ : КПІ ім. Ігоря Сікорського, 2022. –

198 с. Гриф надано Методичною радою КПІ ім. Ігоря Сікорського (протокол № 1 від 02.09.2022 р.).
<https://ela.kpi.ua/handle/123456789/52015>.

4. Архітектура комп'ютерів 2. Процесори. Теорія та практикум [Електронний ресурс] : навчальний посібник для здобувачів ступеня бакалавра за спеціальністю 123 «Комп'ютерна інженерія» / КПІ ім. Ігоря Сікорського ; уклад.: І. А. Клименко, А. В. Каплунов, В. А. Таранюк, В. В. Ткаченко. – Електронні текстові дані (1 файл: 3.73 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2022. – 92 с. – Назва з екрана. URI <https://ela.kpi.ua/handle/123456789/52163>.

4.2. Додаткова література

1. **STMicroelectronics**. *STM32 Reference Manual: архітектура, периферійні модулі та регістри мікроконтролерів STM32*. STMicroelectronics, 2020. — Режим доступу: <https://www.st.com/en/microcontrollers-microprocessors/stm32-32-bit-arm-cortex-mcus.html>

2. **STMicroelectronics**. *STM32 Datasheet: технічні дані конкретної моделі плати STM32 Discovery*. STMicroelectronics, 2020. — Режим доступу: <https://www.st.com/en/microcontrollers-microprocessors/stm32-32-bit-arm-cortex-mcus.html>

3. **STMicroelectronics**. *Application Notes for STM32 (ANxxxx): методичні матеріали та приклади реалізації рішень*. STMicroelectronics, 2020. — Режим доступу: <https://www.st.com/en/application-notes.html>

4. STMicroelectronics. *Introduction to Embedded Systems for IoT*. ST Edu Platform, 2021. — Режим доступу: https://www.st.com/content/st_com/en/campaigns/educationalplatforms/intro-embedded-systems-for-iot-edu.html

5. Docker Inc. *Docker Documentation: containerization for IoT systems*. — Режим доступу: <https://docs.docker.com>.

6. PostgreSQL Global Development Group. *PostgreSQL Documentation: database design and management*. — Режим доступу: <https://www.postgresql.org/docs/>

7. Python Software Foundation. *Python 3.11 Documentation*. Python.org, 2023. — Режим доступу: <https://docs.python.org/3/>

8. Google LLC. *Google Maps Platform Documentation*. Офіційна документація щодо використання Maps API, інтеграції з веб/мобільними додатками. — Режим доступу: <https://developers.google.com/maps/documentation>

9. OASIS Standard. *Message Queuing Telemetry Transport (MQTT) Version 3.1.1 / 5.0 Specification*. OASIS, 2019. — Режим доступу: <https://oasis-open.github.io/mqtt-spec/>

10. *STM32 IoT Projects for Beginners: A Hands-On Guide to Connecting Sensors, Programming Embedded Systems, Build IoT Devices with STM32*. Independently published, 2023. ISBN: 979-8982958653.

11. Desmond A.B., *STM32 Handbook for Programmers: Optimizing Embedded Systems for IoT Development*. Independently published, 2021. ISBN: 979-8649321149.

4.3. Інформаційні ресурси

1. Курс відеолекцій – <https://bbb.comsys.kpi.ua/rooms/iry-u3o-khc-vmk/join>

2. Дистанційний курс – на платформі дистанційного навчання «Сікорський» в середовищі Google Workspace for Education: <https://classroom.google.com/c/NjU1NDI2MzI1NjIw?cjc=zotp7sj>. На ресурсі розміщено додаткові теоретичні матеріали та приклади виконання практичних задач, презентації та відеозаписи лекцій, перелік питань та завдання до самостійного опрацювання, матеріали для підготовки до заліку.

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компоненту)

Розподіл навчального часу за видами занять і завдань з дисципліни згідно з робочим навчальним планом. На кредитний модуль виділено 120 годин та 4 кредитів – 18 лекцій (36 годин), 9 лабораторних робіт (18 годин), самостійна робота – 66 годин.

Лекційні заняття

Для досягнення мети навчальної дисципліни слід зосередитись у лекційному матеріалі на особливостях проектування та розробки вбудованих IoT систем, їх архітектурі та методах інтеграції апаратних і програмних компонентів. Особливу увагу необхідно приділити засобам збору, обробки та передачі даних у реальному часі, використанню протоколів обміну повідомленнями (MQTT), побудові ефективних моделей обробки даних на граничному рівні (Edge Computing), а також методам візуалізації та аналізу даних. Крім того, слід розглянути питання продуктивності, масштабованості, надійності та відмовостійкості IoT систем, включно з управлінням мережею сенсорів, використанням контейнеризації (Docker) та інтеграцією різних платформ (Raspberry Pi, BeagleBone, STM StarterKit).

Лабораторні роботи та проєктна діяльність

Мета лабораторних робіт – придбання практичних вмінь застосування принципів проектування, моделювання та реалізації модулів для інтелектуальних IoT систем з урахуванням вимог до продуктивності та ефективності. Студенти набувають навичок роботи з контейнерними технологіями (Docker), системами обміну повідомленнями (MQTT), базами даних (PostgreSQL), інструментами обробки та візуалізації даних (Python, WebSocket, Grafana, GIS-системи). Для виконання лабораторних робіт та проєктної діяльності використовуються як **віртуальні середовища** (емулятори, віртуалізація, хмарні сервіси), так і **апаратні IoT-платформи** (Raspberry Pi, сенсори, edge-сервери).

Під час виконання лабораторних робіт та проєктної діяльності, студенти не лише набувають технічних навичок у сфері IoT та Smart City, а й отримують досвід **командної взаємодії та Agile-менеджменту**, що є важливим елементом підготовки сучасних фахівців з комп'ютерної інженерії.

Форма контролю / прийому лабораторних робіт:

MoM (Minutes of Meeting) – протокол командної зустрічі, у якому фіксуються:

- розподіл ролей та задач між членами команди;
- планування проєкту та етапів роботи;
- організаційні аспекти виконання лабораторної роботи;
- відзначення проміжних результатів і проблем, що виникли;

Використання GitLab / Jira – фіксація комітів, контроль виконання задач і спринтів команди.

Тематика лабораторних робіт:

Лабораторна робота 1. Розгортання програмного забезпечення (Docker та MQTT Explorer). Створення віртуального середовища. Розгортання та запуск проєкту «Програмна модуль системи моніторингу та аналізу стану дорожнього покриття для розумного міста»

Мета: Організовувати командну роботу за принципами SCRUM: планування, розподіл задач, контроль виконання та документування процесу.

Лабораторна робота 2. Реалізація модуля збирання даних з датчиків та первинна обробка

Мета: розробити програмний модуль для отримання даних від сенсорів дорожнього покриття. Студенти вчаться працювати з протоколами обміну повідомленнями (MQTT), форматами даних (JSON, CSV) та виконують первинну обробку, включаючи фільтрацію та виявлення аномалій.

Лабораторна робота 3. Збір та збереження даних у системі управління базами даних (PostgreSQL)

Мета: ознайомитися з методами зберігання IoT-даних у реляційних базах даних. Студенти проєктують структуру таблиць для універсальної моделі смарт-сіті, реалізують збереження даних у PostgreSQL та виконують прості запити для їх подальшої обробки.

Лабораторна робота 4. Реалізація модуля обробки даних на рівні Edge Server

Мета: навчитися застосовувати концепції Edge Computing для зменшення затримок та оптимізації трафіку. Студенти розробляють алгоритми попередньої агрегації даних, локальної аналітики та оптимізації перед відправкою до центральної системи.

Лабораторна робота 5. Реалізація модуля візуалізації даних

Мета: навчитися будувати інтерактивні засоби візуалізації для смарт-сіті. Студенти використовують Python, WebSocket та інтеграцію з картографічними сервісами (GPS/Map), створюють графіки та карти для моніторингу стану дорожнього покриття та роботи інфраструктури в реальному часі.

Хакатон: «Інтелектуальна IoT-система Smart City»

Хакатон спрямований на розширення базової моделі системи моніторингу стану дорожнього покриття шляхом інтеграції нових технологій і компонентів у концепцію «розумного міста». Студенти отримують універсальну структуру даних і базові модулі системи, після чого самостійно додають власні рішення.

Можливі напрями індивідуальних/командних проєктів:

- підключення додаткових об'єктів інфраструктури (система паркування, світлофор, моніторинг енергоспоживання тощо);
- інтеграція нових сенсорів (шум, освітленість, якість повітря, кліматичні параметри);
- удосконалення аналітики на граничному рівні (Edge AI, фільтрація, прогнозування);
- створення додаткових інструментів візуалізації та моніторингу (наприклад, дашборди у Grafana);
- розробка модулів підвищення завадостійкості та надійності системи.

Результатом хакатону є **презентація командного проєкту**, де студенти демонструють роботу власного розширеного модуля в системі Smart City та обґрунтовують його практичну користь.

Організація виконання лабораторних робіт і проєктної діяльності

Для виконання лабораторних робіт і участі в проєктній діяльності (хакатон) студенти об'єднуються у **команди по 5–7 осіб**, сформовані за принципами **SCRUM-методології**.

Основні вимоги до організації командної роботи:

Формат роботи:

- Робота над проєктом організована у вигляді *спринта* (2 тижні).
- На початку спринту проводиться *планування* (визначення цілей і завдань – *MoM (Minutes of Meeting)*).
- Наприкінці спринту – *демонстрація результатів* та *ретроспектива* (аналіз досягнень і помилок).

Документація та артефакти:

- Backlog команди з описом завдань та пріоритетів (MoM).
- Task board (Jira, Git).
- Щотижневі короткі звіти про прогрес (MoM).

Презентація результатів:

- Усі команди демонструють роботу власного модуля в рамках системи Smart City.
- Презентація включає як технічний аспект (реалізація, інтеграція, тестування), так і аргументацію практичної значущості рішення.

Ролі в команді:

- *Product Owner* – відповідає за бачення проєкту та формулювання вимог (викладач-практик).
- *Scrum Master* – слідкує за процесом, допомагає усувати блокери, контролює комунікації (один із студентів у контакті з викладачем-практиком, можлива ротація по спринтах (**під час виконання лабораторних робіт**), щоб кілька студентів спробували себе в організаційній функції).
- *Development Team* – студенти виконують роль **Development Team**, де всередині розподіляються підролі.

Конкретні підролі в студентській команді (Development Team)

- *Team Lead (Coordinator)* – розподіляє задачі, контролює прогрес, веде Jira.
- *Git Maintainer* – відповідає за наявність комітів, pull requests, підтримку структури репозиторію.
- *Documentation Manager* – веде протоколи MoM, формує звітну документацію.
- *Presentation Designer* – збирає презентацію, допомагає іншим оформити слайди.
- *Developers (3–4 особи)* – реалізують кодові та технічні завдання (збір даних, обробка, візуалізація, edge computing тощо).

У презентації **кожен учасник виступає**, демонструючи свій внесок (це важливо для оцінювання).

6. Самостійна робота здобувача вищої освіти денної форми навчання

Види самостійної роботи (66 годин):

- підготовка до аудиторних занять, виконання поточних домашніх завдань та опрацювання матеріалів лекцій (**0,5 годин x 36 годин = 18 годин**);
- **Виконання індивідуальних/командних завдань у складі лабораторних робіт:** аналіз поставленої задачі, моделювання та реалізація окремих модулів системи (збір даних, обробка, зберігання, візуалізація), проведення обчислювальних експериментів та вимірювання показників продуктивності, оформлення протоколу та підготовка звіту до практичної роботи (1 година x 10 годин лабораторних занять (5 лабораторних робіт) = **10 годин**);
- **Проектна діяльність (хакатон):** участь у командній розробці (хакатон) з використанням принципів SCRUM; підготовка індивідуальних презентацій за результатами виконаних завдань (1 година x 8 годин лабораторних занять (1 проєкт) = **8 годин**);
- підготовка до МКР – підготовка до доповіді (**2 години**);
- підготовка до заліку (**6 годин**);
- **Самостійна робота студентів:** опрацювання основного та додаткового теоретичного матеріалу з дисципліни; розгортання та налаштування програмного середовища (Docker, MQTT Explorer, PostgreSQL, Python), завантаження та використання вихідних кодів програмного забезпечення; робота з системою контролю версій Git та системами управління завданнями (Jira); (**22 години**).

Теми на самостійне опрацювання

Розділ 1

Вступ до IoT та Smart City

1. Ознайомитися з прикладами IoT-проєктів у сферах транспорту, екології, енергетики, міської інфраструктури.
2. Проаналізувати, які об'єкти можуть виступати джерелами даних у Smart City (сенсори, лічильники, транспорт, камери тощо).
3. Порівняти IoT-пристрої та класичні embedded-системи за призначенням, архітектурою та вимогами до масштабування.

2. Архітектура IoT-систем

4. Вивчити типову багаторівневу IoT-архітектуру (sensor → edge → server → client).
5. Побудувати спрощену структурну схему Smart City системи для одного вибраного сценарію.
6. Проаналізувати потоки даних та точки обробки (де і чому відбувається обчислення).

Розділ 2

Технології систем IoT

7. Ознайомитися з призначенням та базовими принципами роботи STM32, MQTT, Docker, Edge-пристроїв.
8. Проаналізувати роль кожної технології в IoT-проєкті та їхню взаємодію.
9. Ознайомитися з офіційною документацією відповідних технологій (Datasheet, Reference Manual, Application Notes).

DevOps-аспекти розробки IoT-систем

10. Вивчити роль DevOps у командній розробці IoT-проєктів.

11. Ознайомитися з концепцією контейнеризації та перевагами використання Docker у IoT.
12. Проаналізувати процес збірки, запуску та оновлення IoT-системи в контейнеризованому середовищі.
13. Розглянути базові підходи до керування конфігураціями та середовищами.

Робота з даними та датасетами

14. Ознайомитися з використанням синтетичних датасетів у навчальних IoT-проектах.
15. Знайти та проаналізувати відкриті датасети Smart City (транспорт, екологія, енергоспоживання).
16. Порівняти синтетичні та реальні дані з точки зору структури, обсягу та якості.
17. Проаналізувати можливості використання різних датасетів у межах одного Smart City проекту.

Розділ 3

Організація рівня збирання даних

18. Організувати збір даних на STM32 через ADC, I2C, SPI, UART.
19. Підключити STM32 до проекту через COM-порт та передати реальні дані на Edge-рівень, виконати тестування отриманих даних.
20. За потреби на основі зібраних реальних даних створити або підготувати синтетичний датасет/CSV-файл для передачі в проєкт і тестування.
21. Якщо реальні дані недоступні, використати інші датасети (в тому числі синтетичні) або CSV для тестування.

Розділ 4

Edge-обробка та базовий cloud-огляд

22. Ознайомитися з принципами попередньої обробки даних на Edge-рівні.
23. Проаналізувати переваги Edge-обробки порівняно з повною обробкою в хмарі.
24. Ознайомитися з базовими поняттями cloud-обробки IoT-даних (без глибокої реалізації).
25. Розглянути поняття IoT Hub та його роль у великих IoT-екосистемах (на рівні огляду).

Розділ 5

Командна робота та проєктна діяльність

26. Ознайомитися з принципами SCRUM у контексті IoT-проектів.
27. Проаналізувати ролі учасників команди (embedded-розробник, backend, DevOps, тестувальник).
28. Опрацювати сценарій командної збірки та запуску Smart City проєкту.
29. Проаналізувати типові проблеми командної розробки та способи їх вирішення.

Підсумковий аналіз

30. Проаналізувати повний цикл створення IoT-проекту: від сенсора до візуалізації.
31. Оцінити власні знання та навички, отримані під час курсу.
32. Визначити напрями подальшого поглиблення знань у сфері Embedded, IoT та DevOps.

Додаткові матеріали для самостійного опрацювання, питання для самоконтролю та контрольні запитання знаходяться на платформі дистанційного навчання «Сікорський» (Classroom) <https://classroom.google.com/c/NjU1NDI2MzI1NjIw?cjc=zotp7sj>.

3. Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

Дедлайни. Для виконання лабораторних робіт встановлюються дедлайни. Дедлайни уточнюються і оприлюднюються викладачем на початку семестру.

Заохочувальні бали (бонуси) виставляються за: активну участь на лекціях та семінарах, наприклад участь в дискусіях; виконання домашніх завдань, ведення конспекту, підготовка повідомлення з презентацією по одній із тем СРС дисципліни тощо. Кількість заохочуваних балів не більше 10.

Заохочувальні бали також можуть нараховуватися за прояви **soft skills** під час виконання контрольних заходів дисципліни (лабораторних робіт, поточних контрольних робіт, модульних

контрольних робіт). До таких проявів належать, зокрема, старанність під час виконання завдань, дотримання встановлених термінів подання робіт, відповідальне ставлення до навчального процесу, самостійність та якість оформлення результатів. Бали нараховуються в розмірі **до 1 балу за одне завдання** за рішенням викладача.

Лабораторні роботи. Виконанню кожної лабораторної роботи передую виконання індивідуального завдання і оформлення його у вигляді протоколу. Протокол виконання лабораторного завдання має бути завантажено в Classroom не пізніше, ніж за добу до лабораторного заняття. До протоколу додаються посилання на Git з кодом програм (за необхідності) і посилання на всі необхідні супровідні матеріали для захисту, зокрема на відеозапис виконання лабораторного завдання (за необхідності - в дистанційному режимі, за використання віддаленого устаткування, тощо). Студент, який прийшов на заняття без підготовленого протоколу до захисту лабораторної роботи не допускається. Першим етапом студент захищає результати виконання індивідуального завдання до лабораторної роботи, на другому етапі – захищає результати виконання завдання шляхом усного опитування або тестування. У дистанційному режимі більшість лабораторних робіт супроводжуються тестами для оцінки вивченого теоретичного та практичного матеріалу замість усного опитування. Викладач залишає за собою право провести усне опитування у разі, якщо результати виконання лабораторної роботи є сумнівними або потребують уточнення. Бали отримані за виконання лабораторної роботи, оформлення протоколу та захист результатів складають оцінку за лабораторну роботу. Тестування проводиться на лабораторному занятті після перевірки результатів виконання індивідуальної роботи. Студент, який не виконав індивідуальне завдання до лабораторної роботи до тесту (захисту результатів) не допускається.

Бали, які студент може отримати за виконання кожної лабораторної роботи наведено в таблиці 1 оцінювання семестрових робіт, розділ 8 силабуса.

Експрес-тести. Окремі теми лекційних занять супроводжуються короткими експрес-тестами (на 15 хвилин), які включають матеріал вивченої теми та питання, які задані на самостійне вивчення. Бали отримані за тест входять в семестрову рейтингову оцінку. Поточні тести не перескладаються і не є обов'язковими для допуску до семестрового контролю.

Модульна контрольна робота. Модульна контрольна робота проводиться у формі виконання практичного проєкту та презентації його розробки. Детальна методологія організації проєктної діяльності викладена в розділі 5 силабуса. Модульна контрольна не виконується поза визначений термін і не переписується. Бали за залікову роботу додаються до балів за лабораторні роботи та модульну контрольну роботи і складають семестрову рейтингову оцінку.

Семестровий контроль. Загальною умовою допуску до семестрового контролю (заліку) є зарахування всіх лабораторних робіт, МКР та рейтинг не менше 60 балів. Таким чином мінімальна оцінка, яку повинен отримати студент для допуску до семестрового контролю (заліку) дорівнює 60 балів за виконання всіх поточних робіт за семестр.

Рейтингова (семестрова) оцінка визначається як сума балів за модульну контрольну роботу, балів за індивідуальну семестрову роботу (лабораторні роботи) та експрес тести. Кількість балів за індивідуальну роботу за семестр дорівнює від 38 до 60 балів. Максимальна оцінка за МКР дорівнює 30 балів.

Відповідно до Положення про систему оцінювання результатів навчання в КПІ ім. Ігоря Сікорського https://osvita.kpi.ua/sites/default/files/downloads/Pologennia_RSO_2022.pdf пп. 3.7 – 3.11 передбачено наступне:

Здобувачі, які виконали всі умови допуску до заліку та мають рейтингову оцінку менше 60 балів, а також здобувачі, які бажають підвищити свою рейтингову оцінку, на останньому за розкладом занятті складають залік у вигляді виконання залікового завдання або співбесіди (за вибором викладача).

У разі виконання залікової контрольної роботи або співбесіди, рейтингова (семестрова) оцінка, визначається, як сума балів за індивідуальну семестрову роботу (що відповідають балам за лабораторні роботи протягом семестру, див. табл. 1) та оцінка за залікову контрольну роботу (співбесіду). Всі інші бали, які набрані протягом семестру (окрім балів за лабораторні роботи) анулюються.

Правила відвідування занять. Відвідування занять є вільним, бали за присутність на лекціях та лабораторних заняттях не додаються, відповідно штрафні бали не передбачаються. Втім, вагома частина рейтингу формується через активну участь у заходах на лабораторних заняттях.

Дистанційний режим навчання. У разі запровадження обмежень на відвідування університету, пов'язаних з введенням карантину або режиму воєнного стану в державі, освітній процес здійснюється у дистанційному режимі відповідно до Положення про дистанційне навчання в КПІ ім. Ігоря Сікорського (<https://osvita.kpi.ua/index.php/node/188>), Регламенту організації освітнього процесу в дистанційному режимі (<https://profkom.kpi.ua/reglament-organizatsiyi-osvitnogo-protsesu-v-distantsiynomu-rezhimi>) та Регламенту проведення семестрового контролю в дистанційному режимі (<https://osvita.kpi.ua/node/148>). У режимі дистанційного навчання заняття відбуваються у вигляді онлайн-конференції на платформах BigBlueButton, Google Meet, Zoom. Посилання на конференцію видається на початку семестру і розміщується в АС «Електронний кампус». З метою забезпечення якісної підготовки здобувачів, дистанційний курс дисципліни розміщено на Платформі дистанційного навчання «Сікорський» (<https://www.sikorsky-distance.org>). Результати оцінювання висвітлюють у АС «Електронний кампус» на особистій сторінці здобувача (<https://ecampus.kpi.ua>).

Правила поведінки на заняттях. На заняттях слід дотримуватись норм етичної поведінки визначених у Кодексі честі Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» (<https://kpi.ua/code>). На території університету здобувачі мають поводити себе відповідно до Правил внутрішнього розпорядку (<https://kpi.ua/admin-rule>).

Визнання результатів навчання, набутих у неформальній/інформальній освіті. Положення про визнання результатів навчання, набутих у неформальній / інформальній освіті (<https://osvita.kpi.ua/index.php/node/179>) регламентує визнання результатів навчання, набутих у неформальній/інформальній освіті:

- у разі навчання на неформальній освіті, яку студент обрав самостійно і хоче зарахувати результати навчання в рамках дисципліни, проходиться процедура валідації, що передбачає подання здобувачем заяви на ім'я декана, декларації підтверджувальних документів; рішення про визнання чи не визнання приймається комісією у складі завідувача кафедри, викладача, гаранта освітньо-професійної програми.

- за рішенням викладача окремі змістовні модулі або теми дисципліни можуть бути зараховані на підставі результатів проходження здобувачем онлайн-курсу або іншого елементу неформальної освіти; перелік завдань та елементів неформальної освіти визначається викладачем заздалегідь і розміщується на платформі дистанційного навчання «Сікорський», про що студенти повідомляються на початку семестру; у цьому випадку здобувач звільняється від виконання відповідних завдань і отримує максимальний бал згідно з рейтинговою системою оцінювання.

Політика використання штучного інтелекту. Використання штучного інтелекту регламентується «Політикою використання штучного інтелекту для академічної діяльності в КПІ ім. Ігоря Сікорського» (<https://osvita.kpi.ua/node/1225>). Усі завдання, як під час виконання навчальних завдань з дисципліни, так і індивідуальні завдання, мають бути результатом власної оригінальної роботи здобувача. Використання ШІ для автоматичної генерації відповідей без подальшого їх аналізу та доопрацювання заборонено.

Порушення академічної доброчесності: використання плагіату. У разі виявлення фактів використання плагіату під час виконання навчальних завдань, студент втрачає право на подальше виконання поточних завдань у межах навчальної дисципліни до з'ясування обставин порушення.

Студент зобов'язаний аргументовано пояснити причини використання плагіату, а також засвідчити готовність дотримуватись принципів академічної доброчесності в подальшому навчанні (**письмово у вигляді пояснювальної записки**); повторно виконати завдання або, за потреби, виконати додаткове завдання відповідно до бачення викладача з урахуванням вимог щодо самостійності.

Зазначені дії регламентуються Кодексом честі Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» (<https://kpi.ua/code>).

8. Види контролю та рейтингова система оцінювання результатів навчання (PCO)

Семестровий рейтинг студента з кредитного модуля розраховується, виходячи із 100-бальної шкали. Рейтинг складається з балів, що студент отримує за виконання 5 лабораторних робіт R_L та модульної контрольної роботи $R_{МКР}$ (проектна діяльність) та експерт-тестів $R_{ЕТ}$.

Максимальна кількість балів за лабораторні роботи складає 50 балів (5 лабораторних робіт по 10 балів), тобто $R_L = 50$.

Критерії оцінювання лабораторних робіт наступні:

– *Протокол* (2 б): Відповідність протоколу лабораторній роботі та встановленим вимогам до оформлення, коректність розміщення протоколу в Classroom/Git, повнота та логічність викладення матеріалу:

0 балів – протокол відсутній або не відповідає лабораторній роботі та встановленим вимогам, зокрема містить скриншоти та інші ілюстративні матеріали, які не дозволяють однозначно ідентифікувати середовище та користувача виконання роботи.

1 бал – протокол частково відповідає лабораторній роботі та вимогам до оформлення; матеріал подано неповно або з порушенням логіки викладення, результати виконання неповні, відсутні висновки.

2 бали – протокол повністю відповідає лабораторній роботі та встановленим вимогам до оформлення; коректно розміщений у Classroom/Git; матеріал викладено повно й логічно, наявні змістовні висновки; надані ілюстративні матеріали (скриншоти, фрагменти консолі тощо) є **автентичними, належать автору роботи та містять ідентифікаційні ознаки користувача середовища виконання**.

– *Виконання завдання* (4 б): розроблені моделі працюють правильно у програмному або апаратному середовищі, відповідають вимогам завдання (2 бали), студент показує власний репозиторій з усіма матеріалами лабораторної роботи; репозиторій містить логічну структуру файлів та історію комітів, що демонструє послідовність роботи (1 бал), студент пояснює та аргументує отримані результати, демонструє розуміння виконаного завдання (1 бал);

– *Опитування/тестування* (4 б) за тематикою лабораторної роботи для зарахування теоретичної частини роботи, відповіді на додаткові теоретичні запитання викладача: 0 – 4 бали.

Усний захист лабораторних робіт за баченням викладача може бути замінений на тестування, зокрема у вигляді автоматизованого тестування на платформі Google Workspace for Education / Moodle. Оцінка за поточний тест – 4 бали.

Детальний підхід до оцінювання кожної лабораторної роботи наведений в таблиці 1.

Таблиця 1. Деталізація оцінювання кожної лабораторної роботи

Назва заняття	Форма контролю	Кількість балів	Допуск до семестрового контролю	Всього балів
Лабораторна робота 1	Розвертання програмного забезпечення. МоМ – протокол обговорень та рішень	10	6	10
Лабораторна робота 2	Виконання завдання в команді/відео	2	6	10
	Протокол МоМ	3		
	GitLab/Jira контроль комітів та завдань	2		
Лабораторна робота 3	Виконання завдання в команді/відео	4	6	10
	Протокол МоМ – звіт про організаційні моменти	3		

	GitLab/Jira контроль комітів та завдань	3		
Лабораторна робота 4	Виконання завдання в команді/відео	4	6	10
	Протокол МоМ	3		
	GitLab/Jira контроль комітів та завдань	3		
Лабораторна робота 5	Виконання завдання в команді/Відео	4	6	10
	Протокол МоМ	3		
	Документація в Classroom – фінальний звіт, документація, GitLab/Jira	3		
Експрес-тести на лекціях			6	10
Модульна контрольна робота	Проектна діяльність. Хакатон		24	40
Всього балів			60	100

Максимальна кількість балів за МКР контрольну роботу дорівнює $R_3 = 40$ балів. Критерії оцінювання та вимоги до змісту проектної документації доступні на платформі дистанційного навчання «Сікорський» <https://classroom.google.com/c/NjU1NDI2MzI1NjIw?cjc=zotp7sj>.

Календарна атестація студентів (на 8 та 14 тижнях семестрів) з дисципліни проводиться за значенням поточного рейтингу студента на час атестації. Якщо значення цього рейтингу не менше 50 % від максимально можливого на час атестації, студент вважається атестованим. В іншому випадку в атестаційній відомості виставляється «неатестовано».

Кількість балів, що отримує студент за семестр визначається за формулою:

$$R = R_{\text{ПР}} + R_{\text{ЕТ}} + R_{\text{МКР}}$$

Якщо студент пише залікову контрольну роботу (проходить співбесіду) для підвищення балів його рейтингова (семестрова) оцінка визначається за формулою:

$$R = R_{\text{ПР}} + R_3$$

Максимальна кількість балів за семестр не перевищує **$R = 100$** .

З урахуванням одержаної суми балів кінцева оцінка визначається за таблицею 2.

Таблиця 2. Визначення семестрової оцінки

Кількість балів	Оцінка
100-95	Відмінно
94-85	Дуже добре
84-75	Добре
74-65	Задовільно
64-60	Достатньо
Менше 60	Незадовільно
Не виконані умови допуску	Не допущено

Робочу програму навчальної дисципліни (силабус):

Складено, д.т.н, проф., професор кафедри ОТ Клименко Ірина Анатоліївна.

Увалено кафедрою ОТ (протокол № 18 від 24.06.2026)

Погоджено методичною комісією ФІОТ (протокол № 12 від 26.06.2026)