



СИСТЕМНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ВБУДОВАНИХ СИСТЕМ

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Перший (бакалаврський)</i>
Галузь знань	<i>12 Інформаційні технології</i>
Спеціальність	<i>121 Інженерія програмного забезпечення</i>
Освітня програма	<i>Інженерія програмного забезпечення інтелектуальних кібер-фізичних систем в енергетиці</i>
Статус дисципліни	<i>Нормативна / Вибіркова</i>
Форма навчання	<i>очна(денна)/очна(вечірня)/заочна/дистанційна/змішана</i>
Рік підготовки, семестр	<i>3 курс, весняний семестр</i>
Обсяг дисципліни	<i>4 кредити, 120 годин, з яких 54 години аудиторних (36 год. лекції, 18 год. практичні), 66 годин становить самостійна робота.</i>
Семестровий контроль/ контрольні заходи	<i>залік</i>
Розклад занять	
Мова викладання	<i>Українська/Англійська</i>
Інформація про керівника курсу / викладачів	Лектор: асистент, аспірант, Голець Владислав Олександрович, +380993880280, vladyslav.holets@edo.kpi.ua Практичні / Семінарські: асистент, аспірант, Голець Владислав Олександрович, +380993880280, vladyslav.holets@edo.kpi.ua
Розміщення курсу	Посилання на дистанційний ресурс (Moodle, Google classroom, тощо)

Програма навчальної дисципліни

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

Силабус навчальної дисципліни «Системне програмне забезпечення вбудованих систем» складено відповідно до освітньої програми «Інженерія програмного забезпечення інтелектуальних кібер-фізичних систем в енергетиці» підготовки бакалаврів спеціальності 121 – Інженерія програмного забезпечення.

Метою навчальної дисципліни є формування та закріплення у студентів наступних здатностей: здатність до абстрактного мислення, аналізу та синтезу, здатність розробляти архітектури, модулі та компоненти програмних систем; здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення; здатність проектувати програмне забезпечення кібер-фізичних та енергетичних систем.

Предмет навчальної дисципліни – архітектурні моделі системного програмного забезпечення, методи синтезу драйверів, алгоритми багатозадачності та механізми прямої взаємодії з апаратними ресурсами мікроконтролерів, які використовуються для розробки кібер-фізичних систем та вузлів периферійних обчислень (Edge Computing) в енергетиці.

Програмні результати навчання, на формування та покращення яких спрямована дисципліна: уміння вибирати та використовувати відповідну задачі методологію створення програмного

забезпечення; застосовувати на практиці ефективні підходи щодо проектування програмного забезпечення; застосовувати на практиці інструментальні програмні засоби доменного аналізу, проектування, тестування, візуалізації, вимірювань та документування програмного забезпечення; вміти створювати програмне забезпечення для інтелектуальних кібер-фізичних систем, в тому числі з врахуванням специфіки предметної області енергетичної галузі; вміти проектувати програмне забезпечення кібер-фізичних та енергетичних систем.

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

Пререквізити:

- **«Основи програмування» (ПО 02) та «Алгоритми та структури даних» (ПО 01)** : впевнене володіння базовими конструкціями мов програмування, методологіями розробки та розуміння структур даних для оптимізації алгоритмів.
- **«Основи комп'ютерних систем і мереж» (ПО 04)** : базове розуміння архітектури обчислювальних систем та принципів мережевої взаємодії.
- **«Фізичні основи кібер-фізичних систем» (ПО 14)** : розуміння фізичних процесів та закономірностей, моніторинг і керування якими має здійснювати розроблене системне ПЗ.
- **«Архітектура системного програмного забезпечення» (ПО 17)** : теоретична концептуальна база, необхідна для переходу від високорівневих абстракцій до низькорівневої розробки (Firmware) під мікроконтролери.
- **«Архітектура та проектування вбудованих систем»**: базове розуміння апаратної складової, принципів організації пам'яті та загальних підходів до створення схемотехнічних рішень для мікроконтролерних пристроїв.
- **«Програмування мікроконтролерів»**: практичні навички розробки базового прикладного коду, первинна робота з портами вводу-виводу (GPIO) та розуміння механіки компіляції прошивок.

Постреквізити:

- **«Безпека програмного забезпечення» (ПО 10)** : розуміння низькорівневої архітектури є необхідним для імплементації методів кіберзахисту безпосередньо на рівні кінцевих пристроїв та сенсорних вузлів.
- **«Побудова масштабованих систем обробки даних у реальному часі» (ПО 26)** : використання створених вбудованих систем як надійних джерел телеметрії та засобів попередньої обробки даних (Edge Analytics) перед передачею у глобальні мережі.
- **«Переддипломна практика» (ПО 11) та «Дипломне проектування» (ПО 12)** : практична реалізація комплексних інженерних проєктів, пов'язаних із розробкою та інтеграцією реальних апаратно-програмних рішень для кібер-фізичних систем в енергетиці.дисципліни.

3. Зміст навчальної дисципліни

Розділ 1. Архітектура апаратного та системного рівнів мікроконтролерів

- **Тема 1.1.** Життєвий цикл розробки Firmware та архітектура мікроконтролерів.
- **Тема 1.2.** Процес завантаження (Boot Sequence) та організація пам'яті.
- **Тема 1.3.** Система тактування та енергозабезпечення ядра.
- **Тема 1.4.** Система переривань та виключень.

Розділ 2. Низькорівнева взаємодія з периферією та інтерфейси зв'язку

- **Тема 2.1.** Пряме керування регістрами (Bare-Metal) та рівні апаратних абстракцій (HAL).
- **Тема 2.2.** Архітектура послідовних інтерфейсів UART та SPI.

- **Тема 2.3.** Інтеграція зовнішніх пристроїв через шину I2C.
- **Тема 2.4.** Промислові інтерфейси в енергетиці: CAN-bus та RS-485.
- **Тема 2.5.** Аналого-цифрові перетворення (ADC).
- **Тема 2.6.** Прямий доступ до пам'яті (DMA).

Розділ 3. Операційні системи реального часу (RTOS)

- **Тема 3.1.** Фундаментальні принципи роботи RTOS.
- **Тема 3.2.** Життєвий цикл задачі в RTOS та керування пам'яттю.
- **Тема 3.3.** Міжпроцесна взаємодія (IPC) та черги повідомлень.
- **Тема 3.4.** Синхронізація доступу до ресурсів (м'ютекси та семафори).
- **Тема 3.5.** Програмні таймери та переривання в RTOS.

Розділ 4. Системна оптимізація та периферійні обчислення (Edge Analytics)

- **Тема 4.1.** Програмне керування енергоефективністю та сплячі режими.
- **Тема 4.2.** Периферійні обчислення (Edge Analytics) на кінцевому вузлі.
- **Тема 4.3.** Апаратне налагодження (JTAG/SWD) та профілювання коду.

4. Навчальні матеріали та ресурси

Базова література:

- Офіційна технічна документація (Reference Manuals, Datasheets, Errata sheets, Application Notes) від виробників мікроконтролерів (наприклад, STMicroelectronics, Espressif Systems).
- Barry, R. Mastering the FreeRTOS Real Time Kernel – A Hands-On Tutorial Guide. (Офіційний безкоштовний посібник від розробників FreeRTOS, доступний у форматі PDF на freertos.org).
- ARM® Cortex®-M Programming Guide (Офіційна відкрита документація архітектури ядра від ARM).
- Офіційна документація екосистеми Zephyr Project (docs.zephyrproject.org).

Додаткова література:

- Специфікації оригінальних розробників комунікаційних протоколів (I2C Specification від NXP, SPI Block Guide від Motorola).
- Офіційна документація інструментального ланцюга GNU Arm Embedded Toolchain (gcc.gnu.org).
- Відкриті гайди та матеріали спільноти PlatformIO (docs.platformio.org) щодо конфігурації систем збірки.
- Технічні блоги та референсні імплементації компаній-лідерів галузі (Nordic Semiconductor, Texas Instruments) щодо енергозбереження та периферійних обчислень (Edge Computing).
- Матеріали відкритих університетських курсів (наприклад, курси з Embedded Systems на edX або Coursera, доступні для безкоштовного аудиту).

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

Лекційні заняття

№ з/п	Назва теми лекції та перелік основних питань (перелік дидактичних засобів, посилання на інформаційні джерела)
1	Тема 1.1. Життєвий цикл розробки Firmware та архітектура МК Основні питання: класифікація системного ПЗ, специфіка розробки для кібер-фізичних систем, етапи компіляції та лінування, огляд архітектури сучасних 32-бітних мікроконтролерів.
2	Тема 1.2. Процес завантаження (Boot Sequence) та організація пам'яті

	<i>Основні питання: механізми Reset Vector, ініціалізація стеку, роль Startup-файлів, конфігурація розподілу пам'яті за допомогою Linker Scripts.</i>
3	Тема 1.3. Система тактування та енергозабезпечення ядра <i>Основні питання: ієрархія тактових частот (PLL, прескейлери), незалежні сторожові таймери (IWDG/WWDG), архітектура доменів живлення мікроконтролера.</i>
4	Тема 1.4. Система переривань та виключень <i>Основні питання: архітектура контролера переривань (NVIC), вектори переривань, пріоритезація подій, забезпечення детермінованості реакції системи.</i>
5	Тема 2.1. Пряме керування регістрами (Bare-Metal) та HAL <i>Основні питання: рівні апаратних абстракцій, прямий доступ до регістрів (LL), робота з GPIO, конфігурація портів вводу-виводу.</i>
6	Тема 2.2. Архітектура послідовних інтерфейсів UART та SPI <i>Основні питання: принципи передачі даних, розрахунок швидкості (Baud rate), регістри конфігурації, побудова кільцевих буферів (Ring Buffers).</i>
7	Тема 2.3. Інтеграція зовнішніх пристроїв через шини I2C <i>Основні питання: адресація на шині I2C, режими роботи (Master/Slave), особливості інтеграції цифрових сенсорів та пам'яті (EEPROM).</i>
8	Тема 2.4. Промислові інтерфейси в енергетиці: CAN-bus та RS-485 <i>Основні питання: диференційна передача сигналів, арбітраж на шині CAN, кадри даних, застосування в інтелектуальних енергомережах.</i>
9	Тема 2.5. Аналого-цифрові перетворення (ADC) <i>Основні питання: архітектура АЦП, регулярні та інжектовані канали, частота дискретизації, обробка аналогових сигналів у реальному часі.</i>
10	Тема 2.6. Прямий доступ до пам'яті (DMA) <i>Основні питання: архітектура контролера DMA, розвантаження процесорного ядра, режими потокової передачі (Memory-to-Peripheral, Peripheral-to-Memory).</i>
11	Тема 3.1. Фундаментальні принципи роботи RTOS <i>Основні питання: відмінності RTOS від архітектури Super-Loop, стани задач, кооперативне та превентивне (Preemptive) планування.</i>
12	Тема 3.2. Життєвий цикл задачі в RTOS та керування пам'яттю <i>Основні питання: створення, призупинення та видалення задач, динамічне та статичне виділення пам'яті під стек задачі, фонові задачі (Idle task).</i>
13	Тема 3.3. Міжпроцесна взаємодія (IPC) <i>Основні питання: механізми безпечної передачі даних між потоками, архітектура та використання черг повідомлень (Queues).</i>
14	Тема 3.4. Синхронізація доступу до ресурсів <i>Основні питання: синхронізація через події, семафори (бінарні та лічильні), м'ютекси, проблема інверсії пріоритетів та способи її вирішення.</i>
15	Тема 3.5. Програмні таймери та переривання в RTOS <i>Основні питання: архітектура програмних таймерів (Software Timers), відкладена обробка переривань (Deferred Interrupt Processing) за допомогою демона RTOS.</i>
16	Тема 4.1. Програмне керування енергоефективністю <i>Основні питання: стратегії зниження споживання (Tickless Idle в RTOS), конфігурація апаратних сплячих режимів (Sleep, Deep Sleep, Standby).</i>
17	Тема 4.2. Периферійні обчислення (Edge Analytics) <i>Основні питання: концепція попередньої обробки даних на кінцевому вузлі мережі, алгоритми стиснення та цифрової фільтрації телеметрії перед відправкою.</i>
18	Тема 4.3. Апаратне налагодження та профілювання

	Основні питання: інтерфейси JTAG/SWD, використання точок зупину (Breakpoints), трасування процесів, аналіз використання стека та пошук витоків пам'яті.
--	---

Практичні заняття

№ з/п	Назва теми заняття та перелік основних питань
1	Практичне заняття №1. Налаштування інструментарію та аналіз Boot Sequence. Основні питання заняття: конфігурація Toolchain та IDE, покрокове налагодження Startup-файлу, дослідження карти пам'яті (Map-file), базова ініціалізація системи тактування.
2	Практичне заняття №2. Розробка драйвера GPIO та обробка переривань. Основні питання заняття: написання драйвера портів вводу-виводу через прямий доступ до регістрів (LL), налаштування зовнішніх переривань (EXTI), програмне усунення брязкоту контактів.
3	Практичне заняття №3. Налаштування асинхронного обміну даними (UART). Основні питання заняття: ініціалізація інтерфейсу UART, реалізація асинхронного прийому та передачі телеметрії за допомогою переривань та побудова кільцевого буфера (Ring Buffer).
4	Практичне заняття №4. Інтеграція цифрових сенсорів через I2C/SPI. Основні питання заняття: зчитування ідентифікаторів та поточних значень зовнішніх цифрових сенсорів, парсинг отриманих пакетів даних, робота з логічним аналізатором.
5	Практичне заняття №5. Потіковий збір аналогових даних (ADC + DMA). Основні питання заняття: налаштування АЦП для зчитування показників енергомережі, конфігурація контролера DMA для фоновієї передачі масивів даних без залучення ресурсів центрального процесора.
6	Практичне заняття №6. Інтеграція ядра FreeRTOS та багатозадачність. Основні питання заняття: додавання вихідних файлів RTOS у проект, налаштування системного таймера, створення незалежних задач з різними пріоритетами, аналіз диспетчеризації.
7	Практичне заняття №7. Міжпроцесна взаємодія (IPC) та синхронізація. Основні питання заняття: передача структур з телеметрією між задачами через черги повідомлень, використання м'ютексів для ексклюзивного доступу до апаратного інтерфейсу (UART/SPI).
8	Практичне заняття №8. Керування енергоспоживанням. Основні питання заняття: переведення мікроконтролера в режим зниженого споживання під час виконання Idle Task, конфігурація пробудження системи за зовнішнім перериванням.
9	Практичне заняття №9. Edge Analytics та апаратне налагодження. Основні питання заняття: імплементація алгоритму попередньої обробки даних (наприклад, ковзного середнього) безпосередньо у прошивці, використання JTAG/SWD для профілювання завантаженості ядра мікроконтролера.

6. Самостійна робота здобувача вищої освіти

Самостійна робота становить більше половини від загального обсягу дисципліни (66 годин) і є ключовим етапом у формуванні професійних інженерних навичок. Робота з вбудованими системами вимагає індивідуального занурення в архітектурні деталі та самостійного вирішення низькорівневих апаратно-програмних конфліктів.

Види самостійної роботи та розподіл часу:

- *Опрацювання технічної документації та підготовка до аудиторних занять (18 годин):* Самостійне вивчення англомовних Datasheets, Reference Manuals та Application Notes від виробників мікроконтролерів перед практичними заняттями. Здобувач повинен заздалегідь орієнтуватися в мапі регістрів, розуміти призначення конфігураційних бітів та базову логіку роботи апаратних блоків.
- *Оптимізація та апаратне налагодження програмного коду (20 годин):* Завершення виконання та розширення функціоналу завдань, розпочатих на практичних заняттях. Включає рефакторинг коду, усунення прихованих логічних помилок, проведення самостійного трасування через інтерфейси JTAG/SWD на реальному обладнанні та підготовку аргументованих відповідей для захисту роботи.
- *Реалізація індивідуального завдання з проектування (18 годин):* Самостійне проектування та імплементація розширеного програмного модуля. Це може бути написання низькорівневого драйвера для специфічного промислового датчика або розробка алгоритму попередньої обробки даних (наприклад, цифрової фільтрації шумів) з його подальшою інтеграцією в загальний RTOS-проект.
- *Підготовка до контрольних заходів (10 годин):* Консолідація теоретичних знань та практичних навичок для написання модульної контрольної роботи (МКР) та підготовки до семестрового контролю (заліку).

Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

- **Правила відвідування занять:** Відвідування лекцій є наполегливо рекомендованим, оскільки на них закладається фундаментальна архітектурна база. Присутність на практичних заняттях є суворо обов'язковою, оскільки вони передбачають безпосередню роботу з апаратним забезпеченням (відлагоджувальними платами, мікроконтролерами) та лабораторним вимірювальним обладнанням (осцилографами, логічними аналізаторами).
- **Правила поведінки на заняттях:** Вітається активна участь в обговоренні архітектурних рішень та методів оптимізації. Під час виконання практичних робіт здобувачі зобов'язані неухильно дотримуватися правил техніки безпеки при роботі з електронним обладнанням.
- **Правила захисту практичних робіт:** Захист відбувається виключно шляхом демонстрації працездатного коду (Firmware) на цільовому пристрої. Здобувач повинен не лише продемонструвати результат, але й вміти пояснити логіку конфігурації регістрів, роботу алгоритму та причини вибору конкретних архітектурних підходів під час проектування.
- **Політика дедлайнів та перескладань:** Кожна практична робота має чітко визначений термін подачі. Роботи, подані на захист із порушенням дедлайну без поважних причин, оцінюються зі зниженням максимального балу на 20%. Перескладання МКР допускається за погодженням з викладачем у спеціально відведений час.
- **Політика щодо академічної доброчесності:** Інженерія вбудованих систем вимагає глибокого розуміння кожного написаного байта. Використання чужого коду (з відкритих репозиторіїв, форумів або згенерованого штучним інтелектом) допускається лише за умови повного розуміння механіки його роботи. Нездатність здобувача пояснити запозичений фрагмент розцінюється як академічна недоброчесність, а така робота анулюється до моменту самостійного переопрацювання.
- **Правила призначення заохочувальних та штрафних балів:** За виконання завдань підвищеної складності (наприклад, реалізація власного драйвера замість використання стандартного HAL) можуть призначатися заохочувальні бали (до +10 балів до семестрового рейтингу, але не більше 100 балів загалом). Штрафні бали застосовуються у разі порушення дедлайнів.

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

Поточний контроль:

- Виконання та захист 9 практичних робіт: кожна робота оцінюється максимум у 4 бали (3 бали за працездатність програмно-апаратного рішення; 4 бали за розуміння архітектури та відповіді на питання під час захисту). Максимальна сума — 63 балів.
- Штрафні бали: захист роботи з порушенням дедлайну без поважних причин призводить до зниження максимального балу за роботу на 2 бали.
- Модульна контрольна робота (МКР): комплексне завдання з проєктування структури системного ПЗ для заданого вузла. Оцінюється максимум у 37 балів (17 балів – теоретичний блок обґрунтування вибору архітектурних рішень, методів синхронізації чи інтерфейсів; 20 балів – практичний блок написання логіки ініціалізації периферії, конфігурації RTOS-задач, налаштування DMA або таймерів).

Календарний контроль: провадиться двічі на семестр як моніторинг поточного стану виконання вимог силабусу.

Семестровий контроль: **залік**

Умови допуску до семестрового контролю: **обов'язковою умовою є виконання та успішний захист усіх передбачених практичних робіт, написання МКР та отримання семестрового рейтингу не менше 60 балів..**

Таблиця відповідності рейтингових балів оцінкам за університетською шкалою:

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 асистент, аспірант, Голець Владислав Олександрович

Ухвалено кафедрою _____ (протокол № ____ від _____)

Погоджено Методичною комісією факультету¹ (протокол № ____ від _____)

¹ Методичною радою університету – для загальноуніверситетських дисциплін.