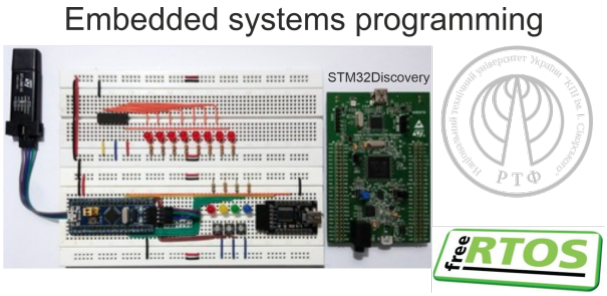
Національний технічний університет України
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Кафедра прикладної радіоелектроніки

[RE-90] ПРОГРАМУВАННЯ ВБУДОВАНИХ СИСТЕМ



Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	Другий (магістерський)
Галузь знань	-
Спеціальність	
Освітня програма	Всі ОП
Статус дисципліни	Вибіркова (Ф-каталог)
Форма здобуття вищої освіти	Очна
Рік підготовки, семестр	Доступно для вибору починаючи з 1-го курсу, весняний семестр
Обсяг дисципліни	5 кред. (Лекц. 18 год, Практ. год, Лаб. 36 год, СРС. год)
Семестровий контроль/контрольні заходи	Екзамен
Розклад занять	https://rozklad.kpi.ua
Мова викладання	Українська
Інформація про керівника курсу / викладачів	Лекц.: Навроцький Д. О., Лаб.: Навроцький Д. О.,
Розміщення курсу	

Програма навчальної дисципліни

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

Чому це цікаво/треба вивчати?

Ринок вбудованих систем (англ. Embedded) стрімко розширюється (кожен наступний рік на десятки мільярдів доларів), тобто збільшується кількість вакансій.

Людину оточує електроніка. Електроніка керується мікроконтролерами. Знаючи як програмувати мікроконтролери, можна створювати, вдосконалювати електронні

пристрої.

Слід зазначити, що порівняно з іншими сферами IT у Embedded відносно великий “поріг входу”. Embedded принципово складніше вивчити самостійно, ніж інші сфери IT, оскільки мікроконтролер це як “чорний ящик” для початківця і потрібен той, хто розповість, як з ним взаємодіяти.

Існує багато спеціалізованих мікроконтролерів, в даному курсі вивчається найпопулярніший на ринку STM32 (ARM Cortex-M архітектура)

Чому можна навчитися (результати навчання)?

- Використовувати STM32CubeIDE (офіційно безкоштовна IDE) для програмування STM32F4
- Основним протоколам обміну даними GPIO, UART, SPI, I2C, …
- Використовувати різну пам’ять STM32
- Створювати файли прошивки мікроконтролера, користуватись програматором для пошуку багів в коді і робити емуляцію в Proteus
- Програмувати STM32 використовуючи бібліотеку HAL
- Встановлювати на STM32 операційну систему FreeRTOS (писати задачі, використовуючи HAL або LL бібліотеку)
- Писати код під ядро ARM Cortex (на основі якого побудований STM32) використовуючи бібліотеку CMSIS (для тих, кому не вистачає можливостей HAL та LL бібліотек)
- Використовувати Git (контроль версій) для свого коду, зберігаючи свій код репозиторіях на GitHub або GitLab

Як можна користуватися набутими знаннями і уміннями (компетентності)?

- Зможете створювати, вдосконалювати пристрої які керуються STM32
- Зможете об’єднувати різними протоколами декілька мікроконтролерів у систему для вирішення задач
- Зможете користуватись Git для взаємодії з командою розробників

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

Бажане володіння мовою C

Але, можна навчитись програмуванню і під час проходження курсу

Буде корисним попередній досвід роботи з Arduino (AVR), ESP8266/ESP32, PIC або програмування STM32 (ARM) у Keil

3. Зміст навчальної дисципліни

Тема 1. Огляд існуючих апаратних і програмних рішень в Embedded

Тема 2. Можливості існуючих IDE. Налаштування STM32CubeIDE (альтернатива Keil)

Тема 3. Програматор ST-Link різних версій, режими bootloader, оновлення Firmware, SWV Trace (Debugging)

Тема 4. Попередні налаштування STM32 в STM32CubeMX (конструктор вбудований в STM32CubeIDE)

Тема 5. Основні бібліотеки. HAL, LL, CMSIS, регістри

Тема 6. GPIO менеджмент (робота з пінами)

Тема 7. Interrupts менеджмент (зовнішні і внутрішні переривання)

Тема 8. Протокол UART (режим опитування, переривання, циклічний буфер)

Тема 9. DMA менеджмент (периферія-пам’ять, пам’ять-пам’ять, пам’ять-периферія, нормальний і циклічний режими DMA)

Тема 10. Годинник реального часу в STM32, Осцилятор, RCC

Тема 11. Таймери (лічильник, переривання, ШІМ (PWM) сигнал)

Тема 12. АЦП (Analog-to-Digital Conversion)

Тема 13. ЦАП (Digital-to-Analog Conversion)

Тема 14. Протокол I^2C

Тема 15. Протокол SPI

Тема 16. Робота з пам’яттю мікроконтролера (SRAM, Flash, CCM Memory). Робота в STM32CubeProgrammer

Тема 17. Операційна системи FreeRTOS (налаштування, створення задач, “кучі” (heap), синхронізація (черги, семафори, м’ютекси)

Тема 18. Протокол USB (взаємодія периферії з комп’ютером)

Тема 19. Розробка IoT

4. Навчальні матеріали та ресурси

1. *Carmine Noviello*. Mastering STM32 - Second Edition [Електронний ресурс]. – Режим доступу: https://leanpub.com/mastering-stm32-2nd
2. STMicroelectronics [Електронний ресурс]. – Режим доступу: https://www.youtube.com/@stmicroelectronics
3. ControllersTech [Електронний ресурс]. – Режим доступу: https://www.youtube.com/@ControllersTech
4. *Scott Chacon* and *Ben Straub*. Pro Git 2nd ed. Edition [Електронний ресурс]. – Режим доступу: https://git-scm.com/book/uk/v2
5. STM32CubeIDE. The all-in-one STM32 development tool. For free [Електронний ресурс]. – Режим доступу: https://www.st.com/content/st_com/en/stm32cubeide.html

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

Лабораторні роботи:

1. Робота з пінами мікроконтролера. General-purpose input/output, GPIO
2. Зовнішні переривання. External interrupt, GPIO_EXTI
3. Універсальний асинхронний приймач/передавач. Universal Asynchronous Receiver-Transmitter, UART
4. Прямий доступ до пам'яті. Direct Memory Access, DMA
5. Осцилятор. Reset and Clock Control, RCC
6. Таймери (лічильник, переривання, ШІМ (PWM) сигнал
7. Аналого-цифровий перетворювач. Analog-to-digital converter, ADC
8. Цифро-аналоговий перетворювач. Digital-to-Analog Convertor, DAC
9. Шина I2C. Inter-Integrated Circuit, I2C or IIC bus
10. Шина SPI. Serial Peripheral Interface, SPI bus
11. Робота з пам’яттю мікроконтролера (SRAM, Flash, CCM Memory)
12. Операційна система реального часу. Real-Time Operating System, RTOS

13. Шина USB. Universal Serial Bus
14. Засоби передачі даних в мережі. Інтернет речей. Internet of Things, IoT. Internet of Everything, IoE

6. Самостійна робота студента

Домашня контрольна робота:

Розробити власний пристрій (реальний або намалювати схему в Proteus), який керується мікроконтролером STM32 (написати код в STM32CubeIDE). Викласти проект у своєму репозиторії на GitHub або GitLab (продемонструвати, що вмієте працювати з Git)

Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

Максимум практичних занять, мінімум теорії (тільки необхідне)

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

...

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

9. Додаткова інформація з дисципліни (освітнього компонента)

...

Опис матеріально-технічного та інформаційного забезпечення дисципліни

Програмне забезпечення: STM32CubeIDE, Visual Studio.

Обладнання: плати STM32Diskovery

Лабораторні роботи: 12 шт з використанням різної периферії, яка взаємодіє з мікроконтролером.

Робочу програму навчальної дисципліни (силабус):
Складено Навроцький Д. О.;
Ухвалено кафедрою ПРЕ (протокол № 06/2023 від 22.06.2023)
Погоджено методичною комісією факультету/ННІ (протокол № 06-2023 від 29.06.2023)