****

Міністерство освіти і науки України

Національний технічний університет України

“Київський політехнічний інститут імені Ігоря Сікорського”

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

Лабораторна робота №4  
Проектування вбудованих систем

«FreeRTOS & QP»

Варіант 17

Виконав Перевірив:

студент групи ІА–13: Гордієнко Н.Ю.

Окаянченко Д.О.

Київ 2025

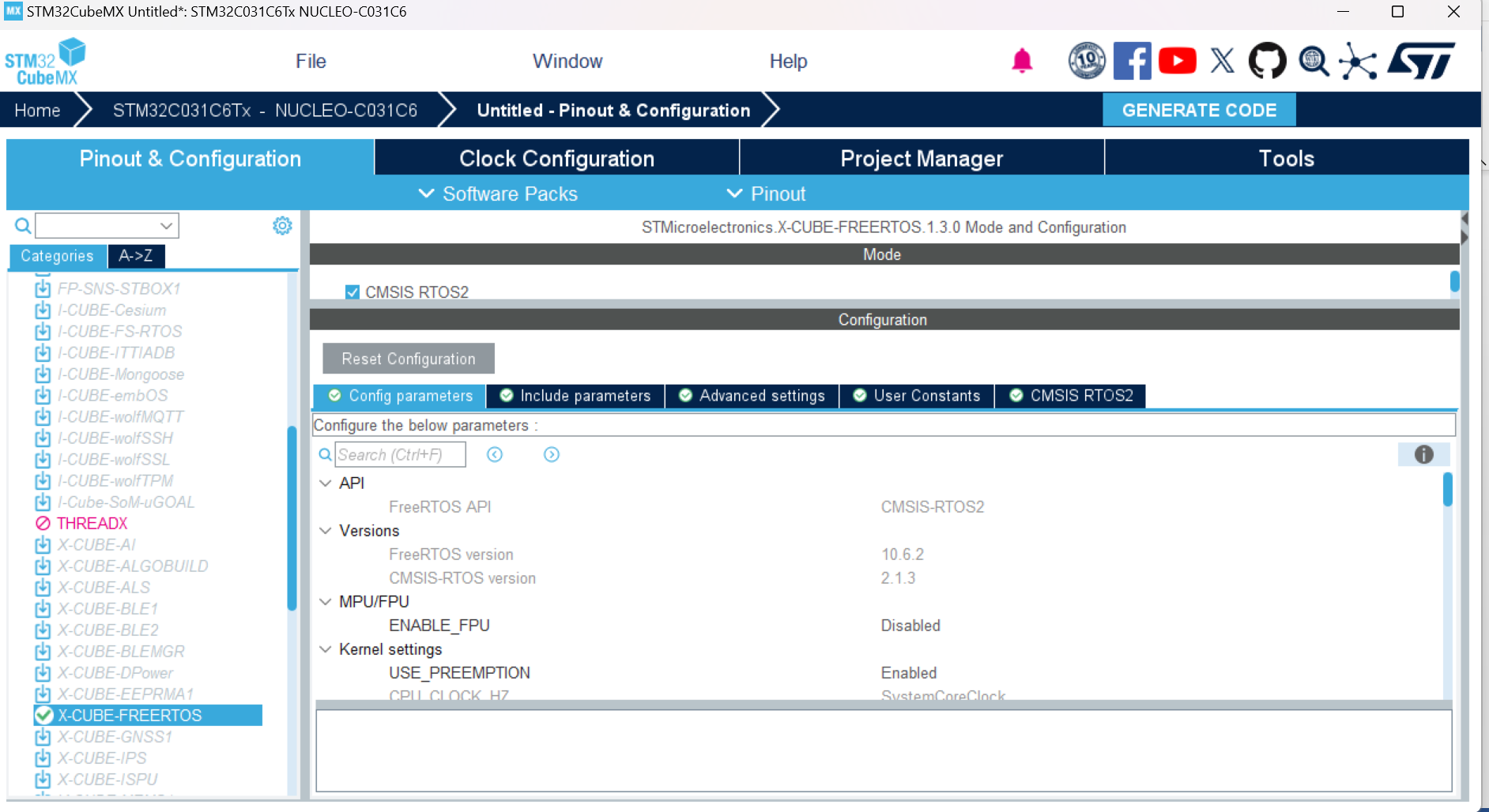
**Мета роботи:** Навчитись працювати з операційними системами реального часу та іншими фреймворками. Ознайомитись з механізмами переривань.

**Посилання на гітлаб-репозиторій:** <https://gitlab.com/david.okayach/designing-embedded-systems>.

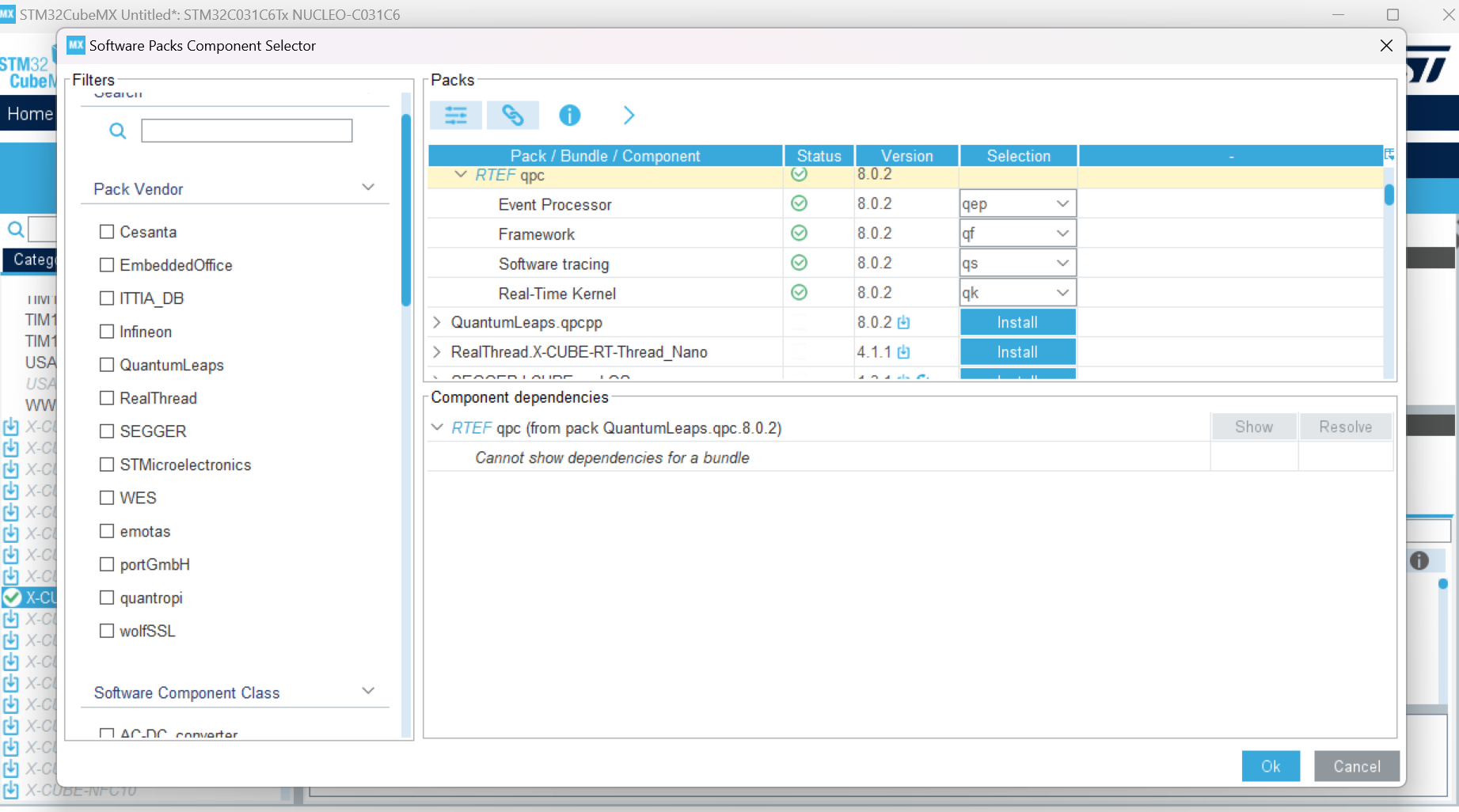
## Хід роботи

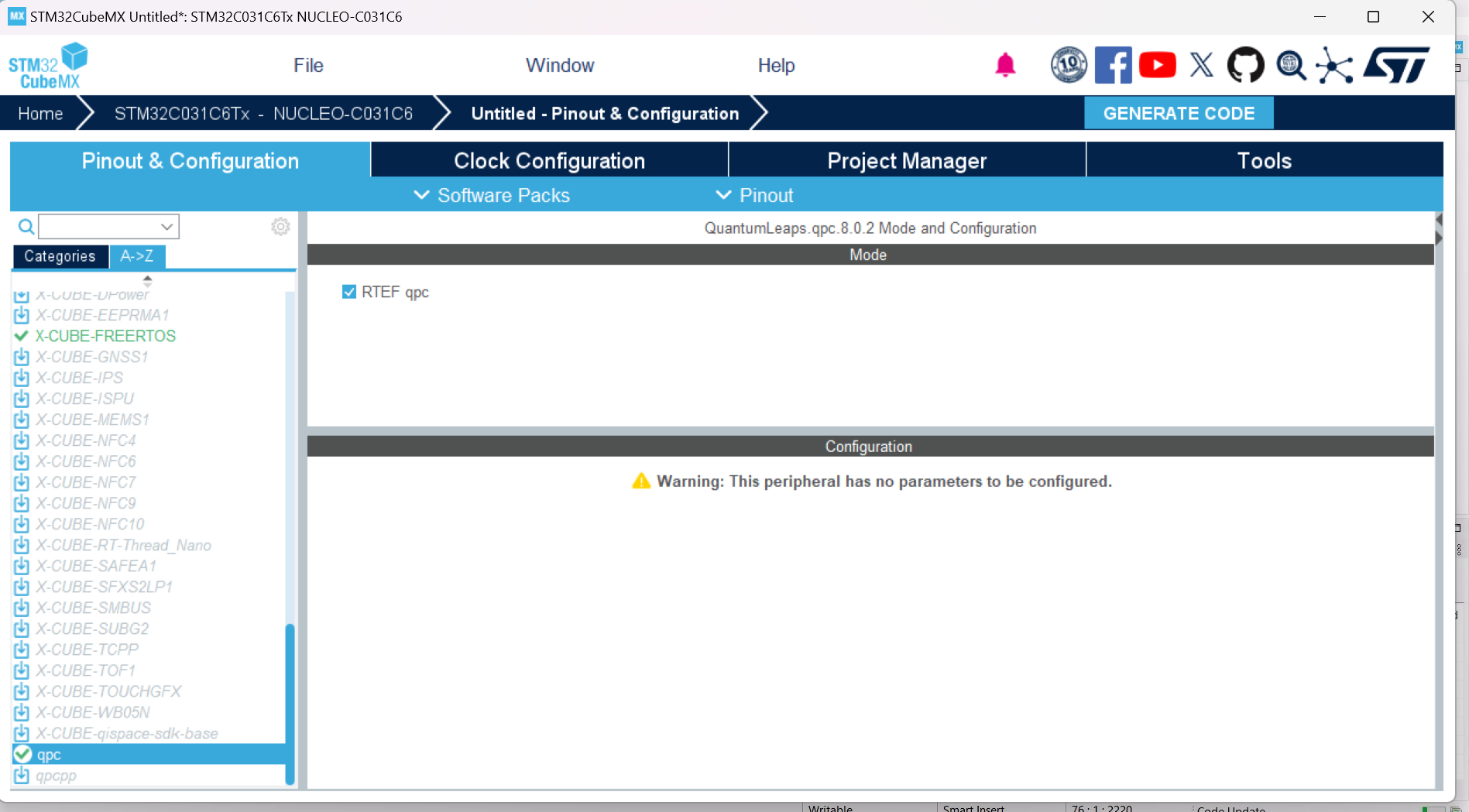
1. Створимо новий проєкт, додамо FreeRTOS і QP: (<https://www.youtube.com/watch?v=M05jMX3_wJk> <https://www.youtube.com/watch?v=E96nvsnY6yM>)

FreeRTOS:

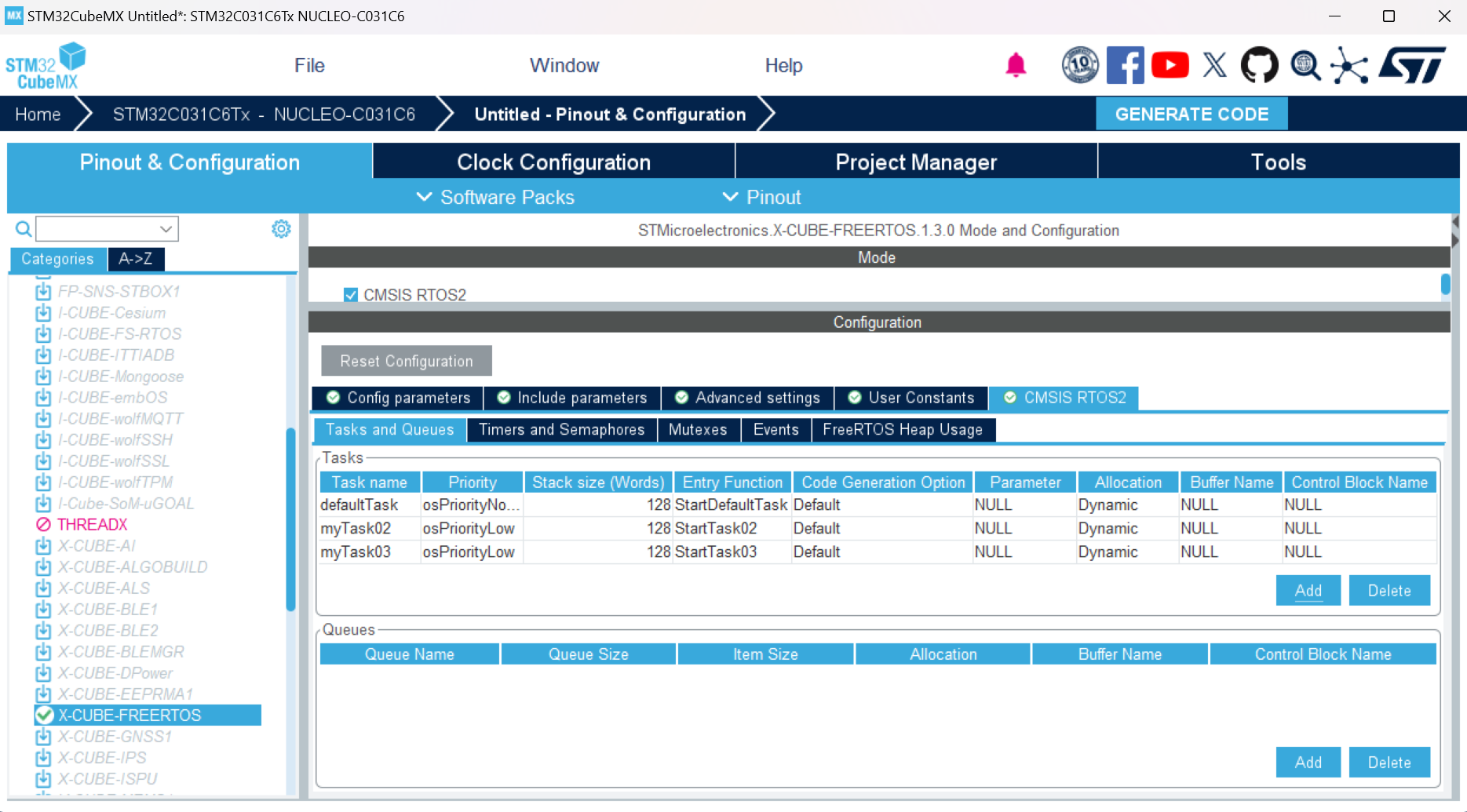


QP:

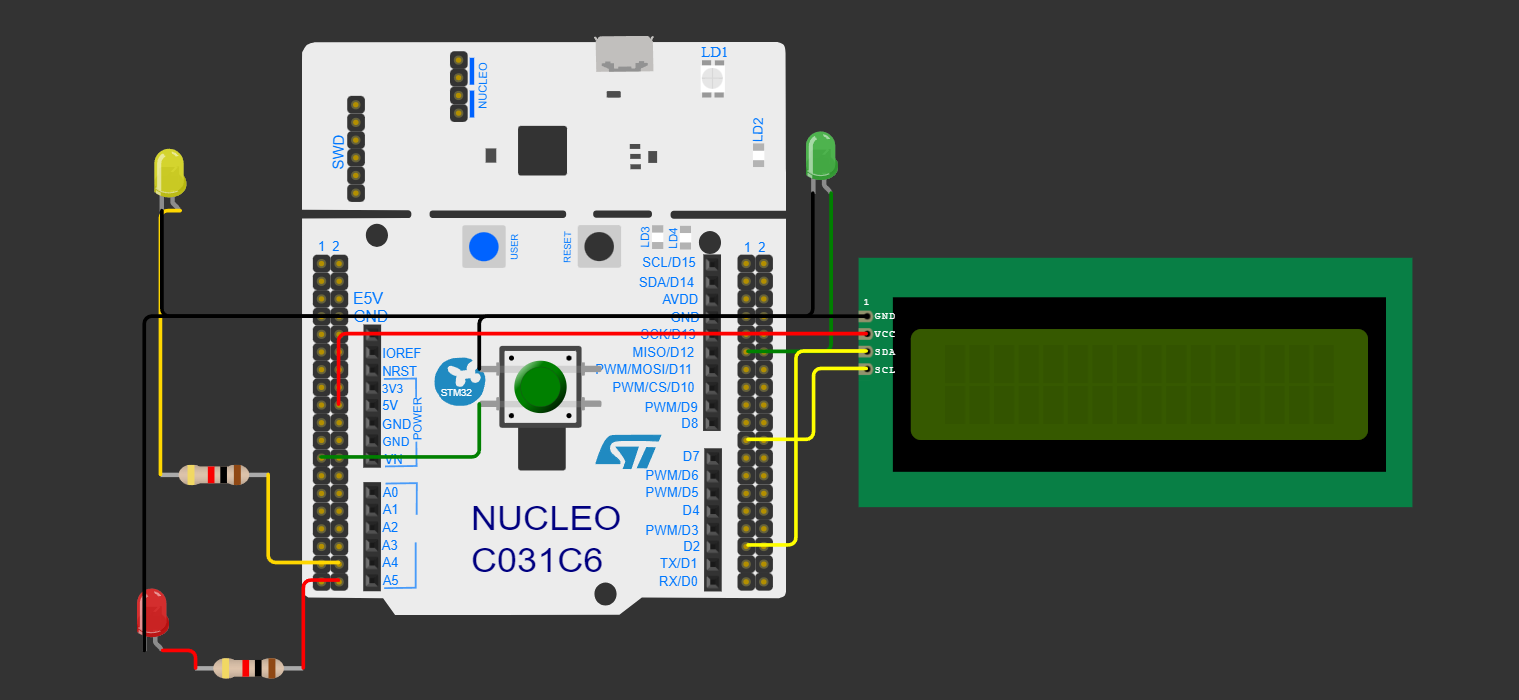




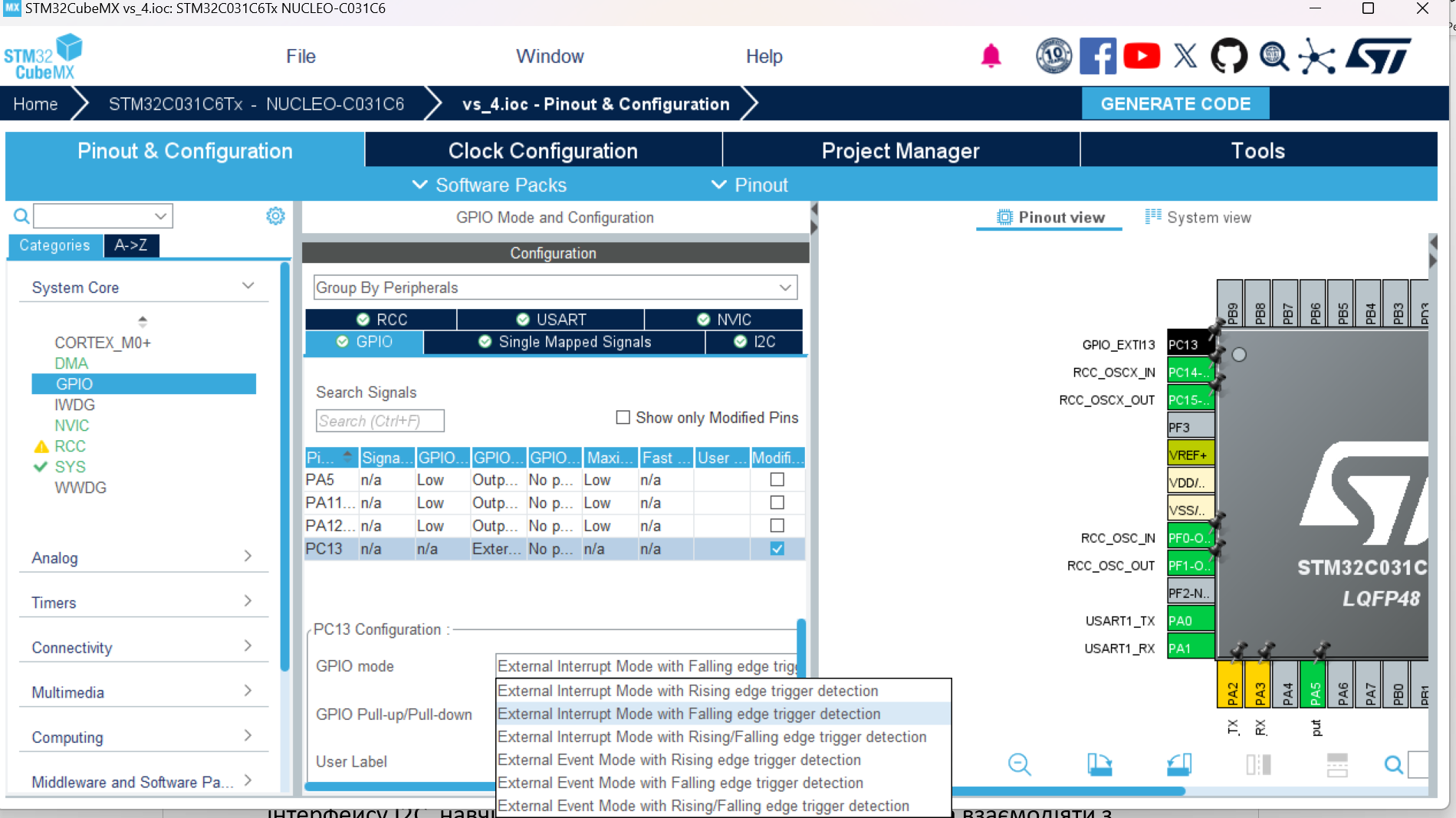
1. Створимо декілька Task за допомогою FreeRTOS:



1. (додатково) Додамо кнопки (обовʼязково з обробкою їх натиску та інші периферійні пристрої, такі як LEDs, UART, I2C сенсори та прилади на вибір:



Обробка натиску кнопки відбувається в методі HAL\_GPIO\_EXTI\_Falling\_Callback, адже при конфігурації обирали наступний режим GPIO: **External Interrupt Mode with Falling edge trigger detection** для піну PC13. Це означає, що переривання буде викликано при **перехідному фронті з HIGH (1) в LOW (0)**, що відповідає натисканню кнопки (оскільки більшість кнопок мають **pull-up резистор** і за замовчуванням перебувають у стані HIGH).



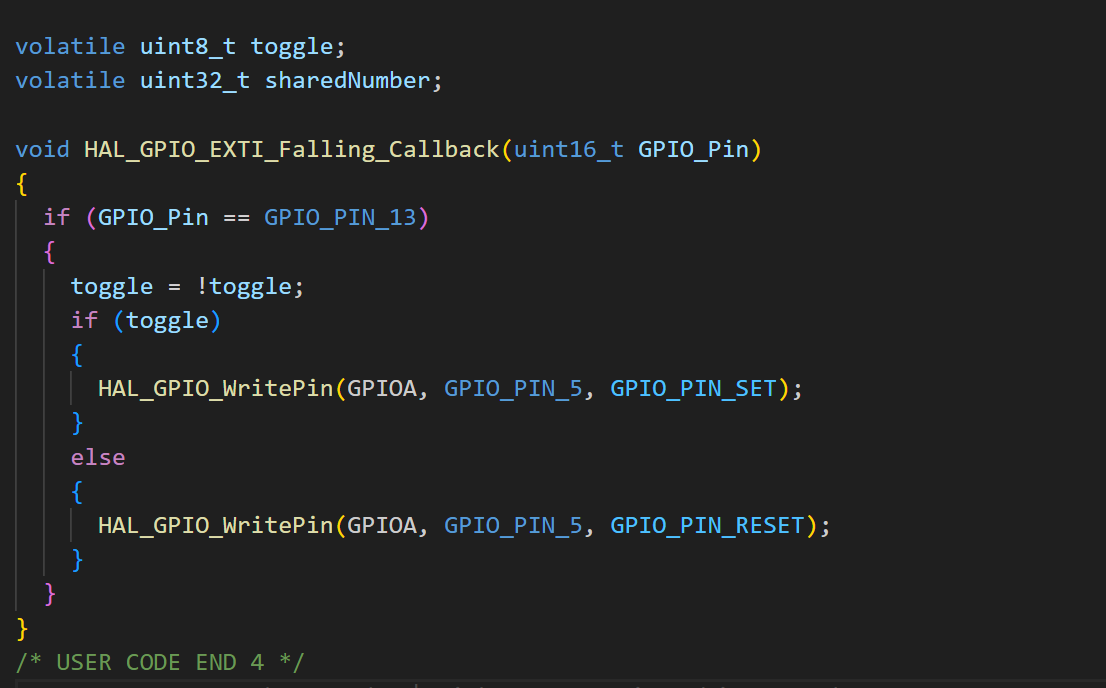
1. Створимо застосунок у якому відповідальність буде розділено між Tasks.

У цьому застосунку відповідальність між процесами була розподілена між кількома задачами (Tasks), що забезпечує ефективне виконання різних функцій у багатозадачному режимі.

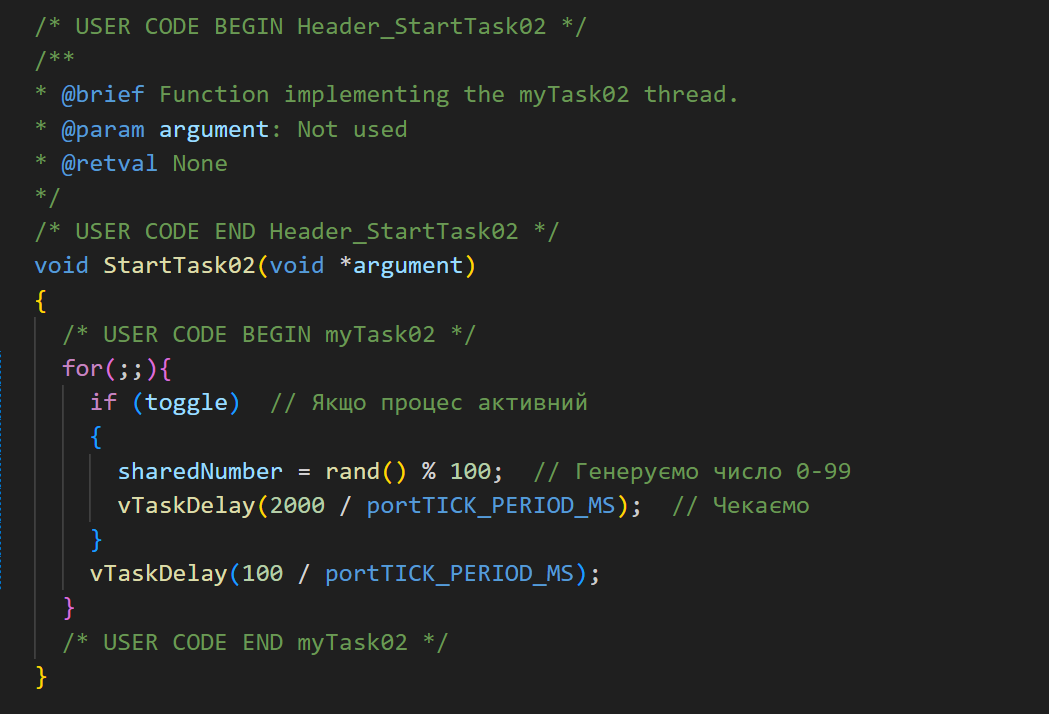
Застосунок складається з трьох основних потоків. Перший потік (defaultTask) є базовим і відповідає за загальну ініціалізацію та підтримку роботи системи. Другий потік (myTask02) виконує генерацію випадкових чисел у межах 0-99, якщо активний прапорець toggle. Цей прапорець змінюється за допомогою обробника переривань від кнопки користувача, що дозволяє вмикати або вимикати роботу основних задач. Третій потік (myTask03) перевіряє, чи є згенероване число парним чи непарним, і відповідно змінює стан світлодіодів, а також виводить значення числа на LCD-дисплей.

Таким чином, у застосунку реалізовано чітке розділення відповідальності між задачами: одна відповідає за генерацію чисел, інша — за обробку та візуалізацію даних. Це забезпечує гнучку і модульну архітектуру, яка може легко розширюватися.

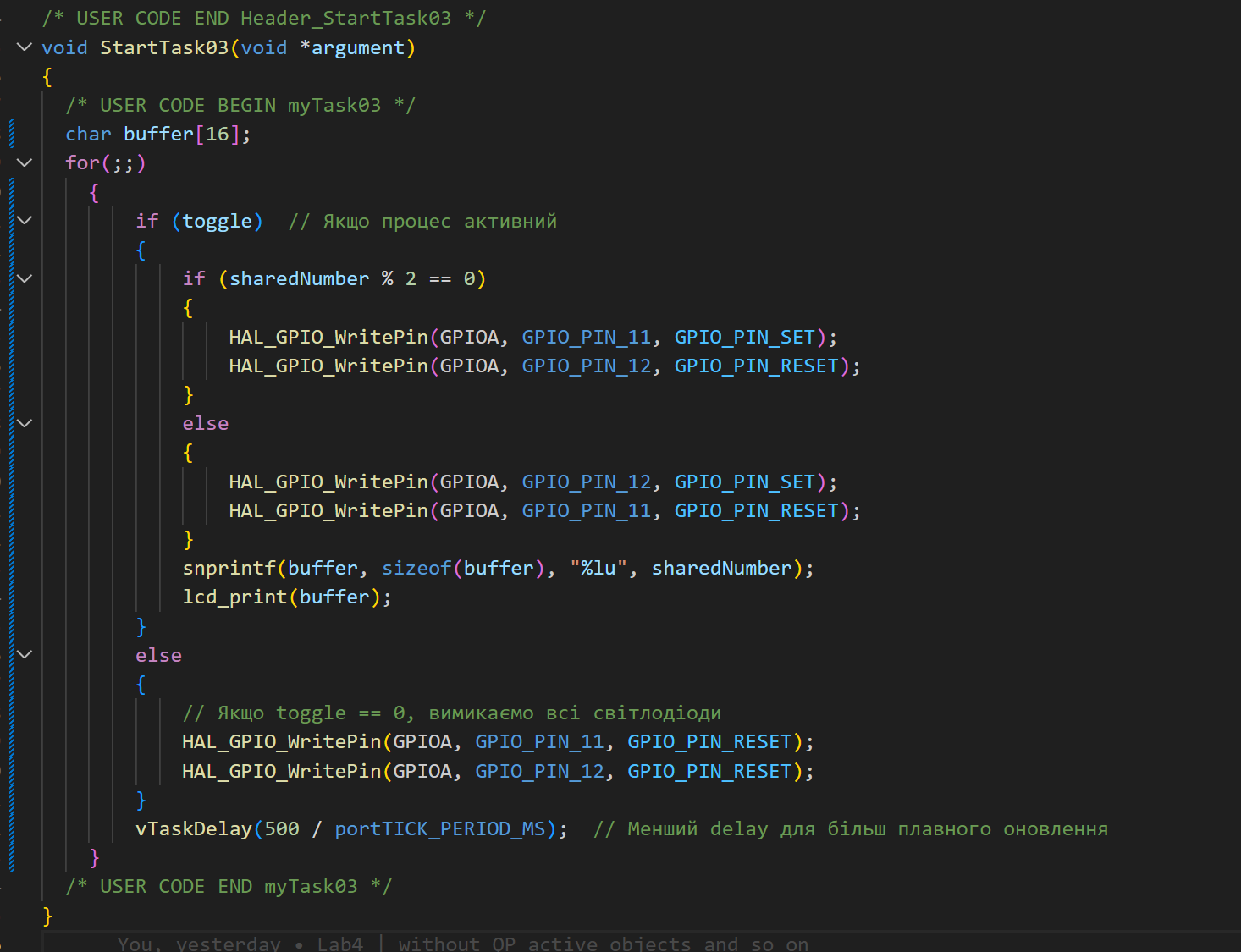
Код методу, який виконується при натисканні на кнопку:



StartTask02:

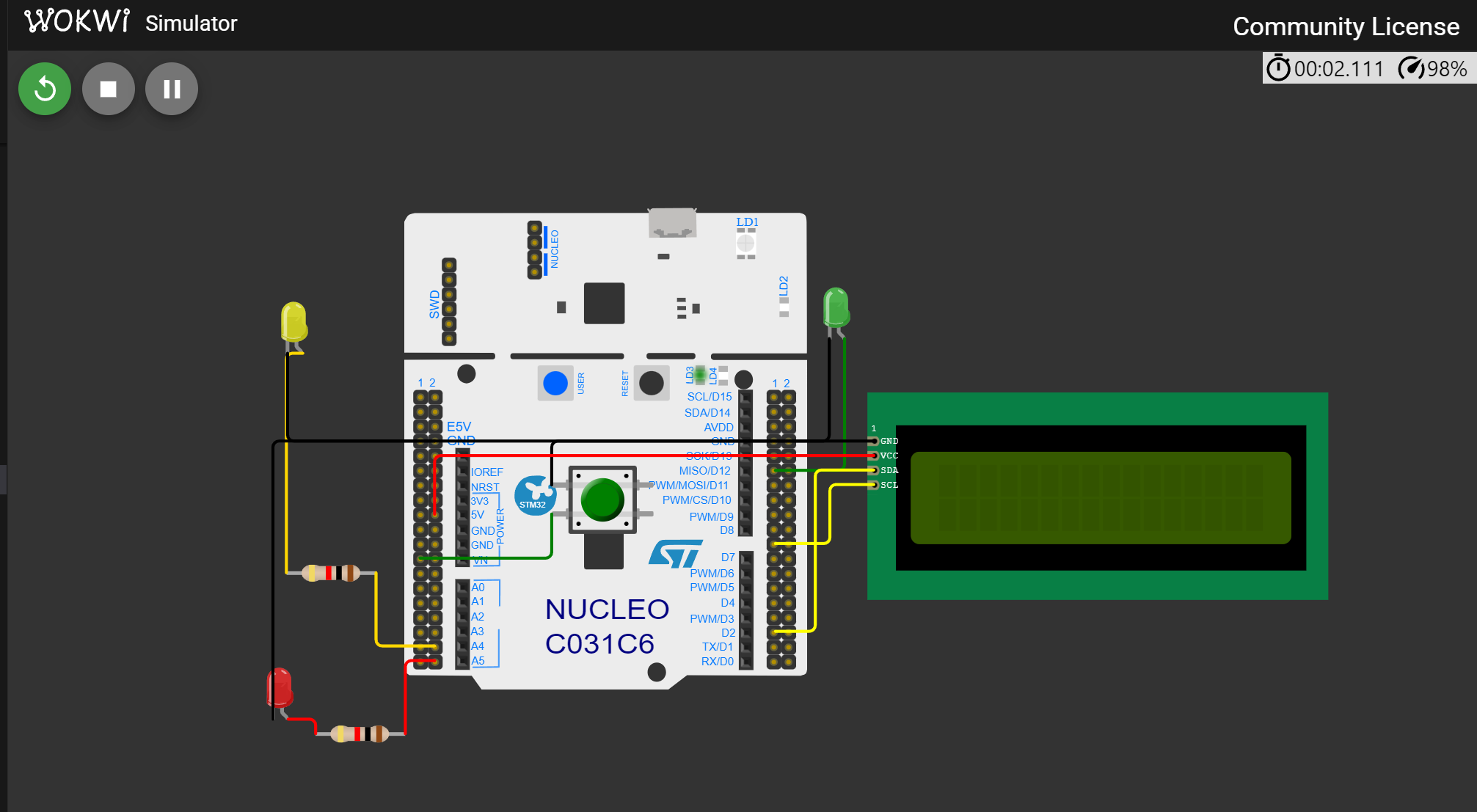


StartTask03:

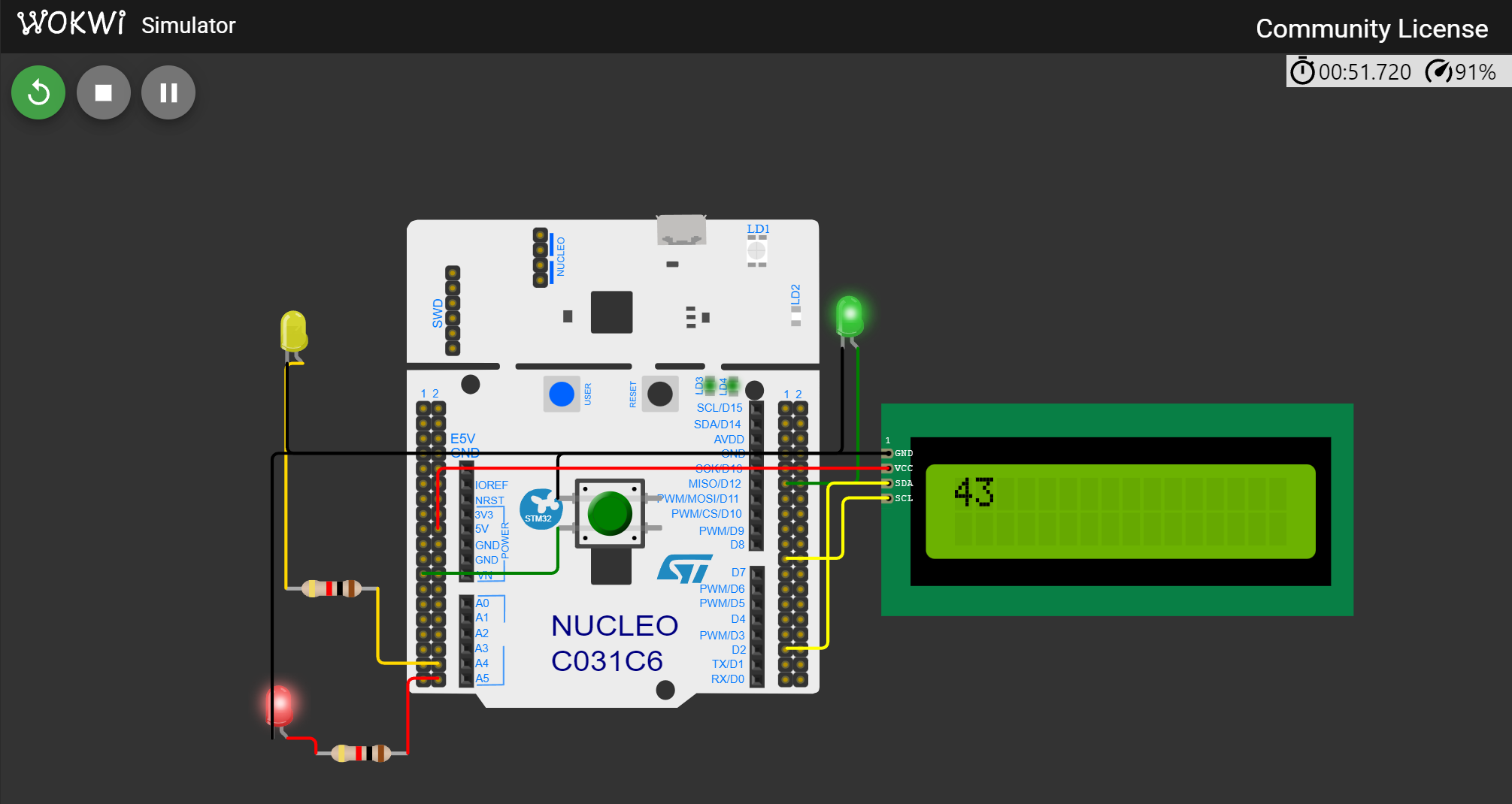


Результати:

* Стан системи, поки не було натиснуто на кнопку (стан спокою):

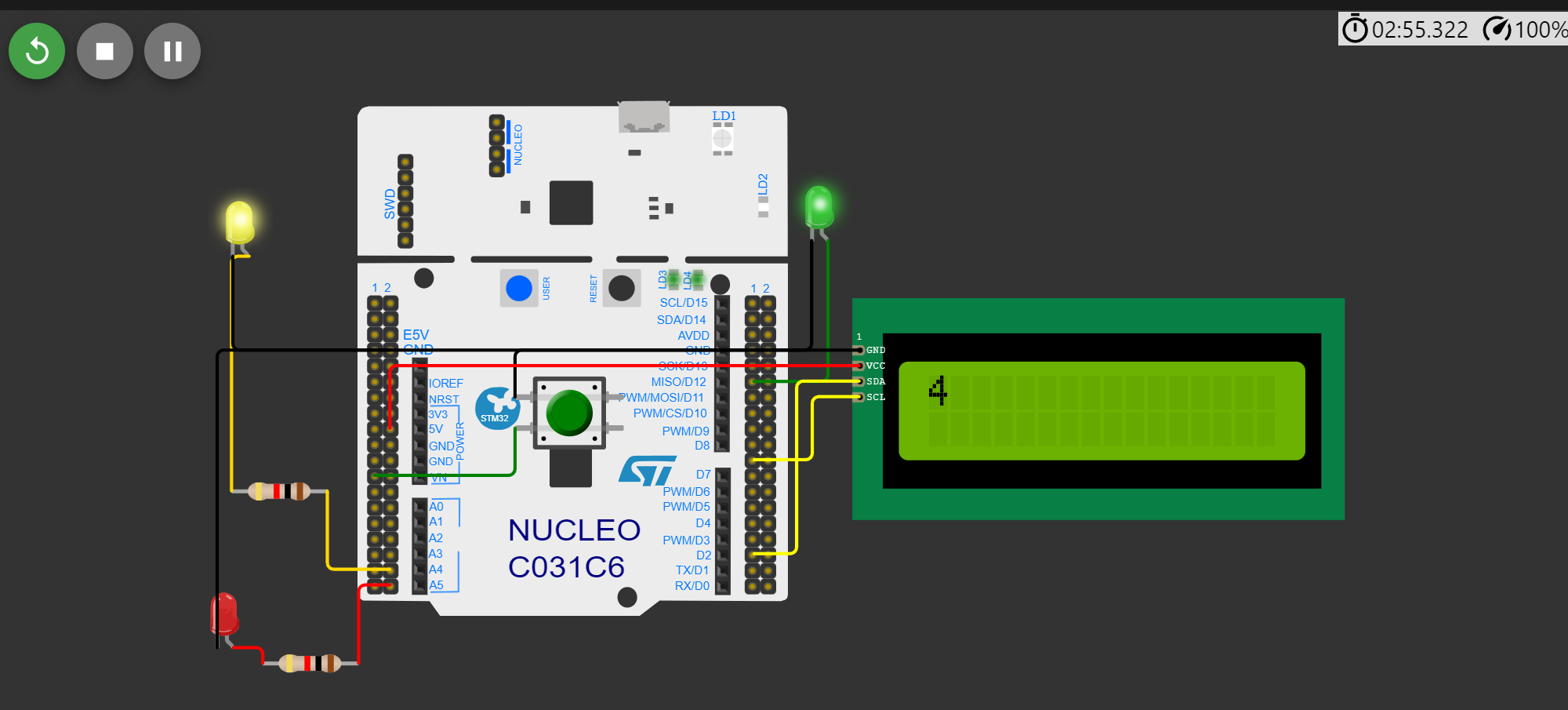


* Натискаємо кнопку:



Відразу бачимо, що було згенеровано випадкове число 43, яке є непарним, тому було передано живлення на пін 12, тим самим, червона лампа загорілася.

* Стан системи, коли згенеровано парне число:



**Висновок:** У ході лабораторної роботи було реалізовано застосунок для STM32 з використанням FreeRTOS та QP, у якому відповідальність між процесами розподілено між кількома задачами. Було створено три потоки: один для загальної ініціалізації, другий — для генерації випадкових чисел, третій — для їх обробки та відображення на LCD. Реакція на натискання кнопки реалізована через переривання, що дозволяє керувати станом системи. Робота продемонструвала переваги багатозадачності у вбудованих системах та закріпила навички використання FreeRTOS у реальних проєктах.