

### Лабораторна робота № 2

#### Вивчення IDE PSoC Creator 4.2 для реалізації проектів з мікроконтролерами PSoC 6 фірми Cypress

**Мета роботи:** Ознайомитися з інтегрованим середовищем розробки PSoC Creator 4.2 та алгоритмом створення проектів в ньому.

#### Теоретичні відомості

За останнє десятиліття стрімко розвинувся новий напрямок проектування електронної апаратури, а саме, проектування апаратури у вигляді систем на кристалі (SoC). У сучасній технічній літературі сформувався таке визначення SoC: SoC – це HBIC, яка вміщує функціональні вузли закінченого пристрою, інтегрованого на кристалі, для автономного використання в електронній апаратурі. Типовим застосуванням таких схем є широке різноманіття вбудованих систем. SoC може містити як цифрові, так і аналогові вузли. Основним цифровим вузлом зазвичай є процесор, що виконує програмне опрацювання цифрових даних та сигналів. Для виконання спеціалізованих функцій SoC має різноманітні цифрові вузли та аналогові схеми, орієнтовані на конкретну галузь застосування SoC. Залежно від застосування SoC прикладом таких периферійних вузлів є різноманітні таймери, лічильники, логічні елементи, ЦАП, АЦП та інші. До складу SoC можуть входити або під'єднуватися до них у вигляді зовнішніх блоків різні типи модулів пам'яті (SRAM, DRAM, Flash, ROM, EEPROM). Взаємодія із зовнішнім середовищем відбувається з використанням послідовних чи паралельних портів, а також комунікаційних інтерфейсів. Конфігурація вузлів SoC визначається їх функціональним призначенням. Організація зв'язків між вузлами системи може бути цілком різноманітною, включаючи використання стандартизованих шин (типу AMBA) чи спеціалізованих локальних інтерфейсів. Сьогодні системи на кристалі займають одну із найбільших частин ринку мікроелектроніки. Це зумовлено великим інтересом як інженерів-розробників апаратури з використанням SoC, так і виробників елементної бази для SoC, зокрема Texas Instruments, Analog Devices, Altera, Atmel, Xilinx, Cirrus Logic Cypress, NXP, RDC, Sharp, Marvell, NetSilicon та інших.

Великий інтерес викликають мікроконтролери фірми Cypress, або, як вони ще називаються, PSoC (Programming System on Chip), тобто "програмована система на кристалі". Відмінною особливістю подібної системи є те, що, крім мікроконтролерного ядра, кристал містить розвинену аналогову периферію.

Мікрочіп PSoC BLE 6 - це енергоефективний, гнучкий у використанні мікроконтролер з вбудованою підтримкою бездротового зв'язку Bluetooth Low Energy і засобами апаратного захисту. Даний компонент успішно застосовують в різних вбудованих системах, використовуючи гнучку 2- ядерну архітектуру: ARM Cortex-M4 в якості хост-процесора і Cortex-M0+ для управління

периферійними функціями, наприклад, інтерфейсом ємнісного сенсора, зв'язком Bluetooth Low Energy, зчитуванням даних з давачів.

Сфера застосування мікроконтролерів включає переносну електроніку, персональні медичні прилади, бездротові акустичні системи. Для захисту даних передбачені вбудовані можливості.

Блок – діаграма PSoC 6 зображена на рис. 1.

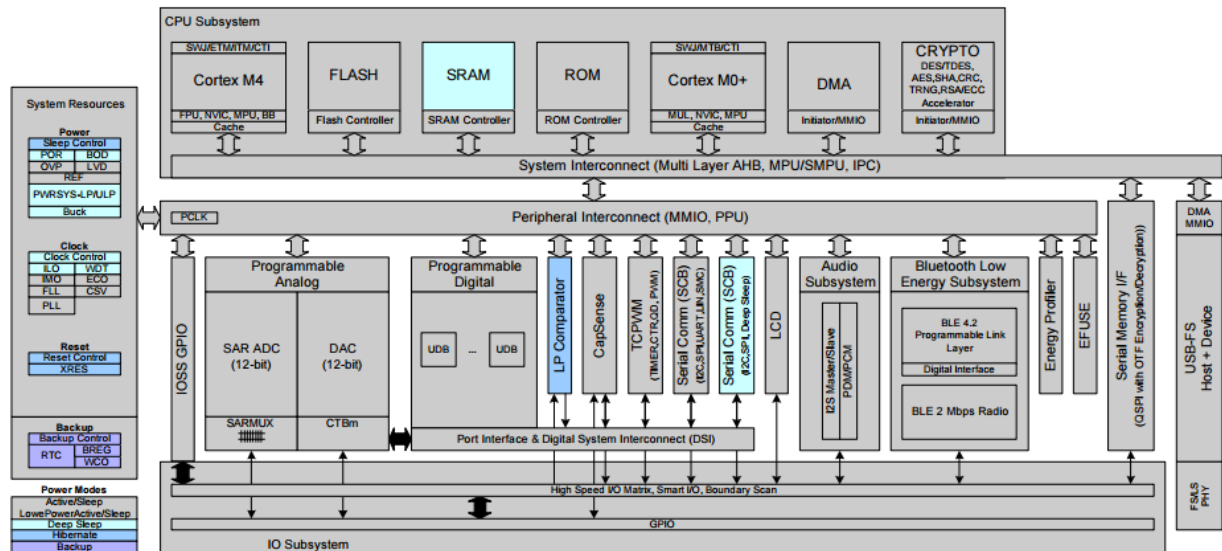


Рис. 1.1. Блок-діаграма мікроконтролера PSoC 6

Система на кристалі існує в декількох варіантах: PSoC 1 на базі ядра M8 (8-розрядні МК), PSoC 3 і PSoC 5 на основі ядер 8051 і Cortex-M3 відповідно, а також PSoC 5LP - удосконалений PSoC 5, основною відмінністю якого є збільшена швидкодія АЦП з 600 Ksps до 1 Msps.

Широкі можливості представляються розробникам: система на кристалі дозволить створювати проекти в найкоротші терміни за рахунок розвиненої аналогової частини в поєднанні з можливостями МК.

Середовище розробки IDE PSoC Creator розроблене для проектування систем на базі мікроконтролерів PSoC типу PSoC 3 та PSoC 5, включаючи PSoC 5LP, PSoC 4 і PSoC 6. Для інсталяції середовища розробки краще використовувати ОС не нижче Windows 7. Середовище PSoC Creator для особистих некомерційних використання та в навчальних цілях надається безкоштовно на сайті фірми [www.cypress.com](http://www.cypress.com), але вимагає онлайн реєстрації як в фірмі Cypress, так і в фірмі розробнику компілятора MDK фірми Keil. Версія IDE PSoC Creator 4.2 має об'єм 540.83 МБт.

Необхідно виконати декілька кроків, щоб почати роботу з першим проектом:

1. Налаштування інструментів. Середовище розробки PSoC Creator поширюється з версією компілятора GCC для ARM. Проте, PSoC Creator може бути налаштований на використання альтернативних компіляторів, таких як MDK фірми Keil або RVDS фірми ARM. Після налаштування компілятора його не потрібно більше налаштовувати знову.

2. Проектування додатків.

3. Побудова додатків. Після того, як додаток розроблений, він повинен бути побудований. Отже, PSoC Creator синтезує логіку, компілює код та назначає аналогові і цифрові виводи введення/виведення.

4. Налаштування стенду. Мікропроцесорний стенд може мати перемички для вибору рівня потужності або інші специфічних параметрів. Крім того, деякі стенди мають вбудовані програматори і для них потрібні драйвери при їх першому підключенні до комп'ютера.

5. Програма та її запуск. Кожний зі стендів має вбудований або автономний програматори. У всіх ситуаціях, описаних тут, програматори визначаються автоматично.

### ***Створення першого проекту в IDE PSoC Creator 4.2.***

При запуску PSoC Creator 4.2 відкриється вікно, зображене на рис. 2.2. Для кращого розуміння середовища IDE PSoC Creator, ознайомимося з наступними термінами і поняттями.

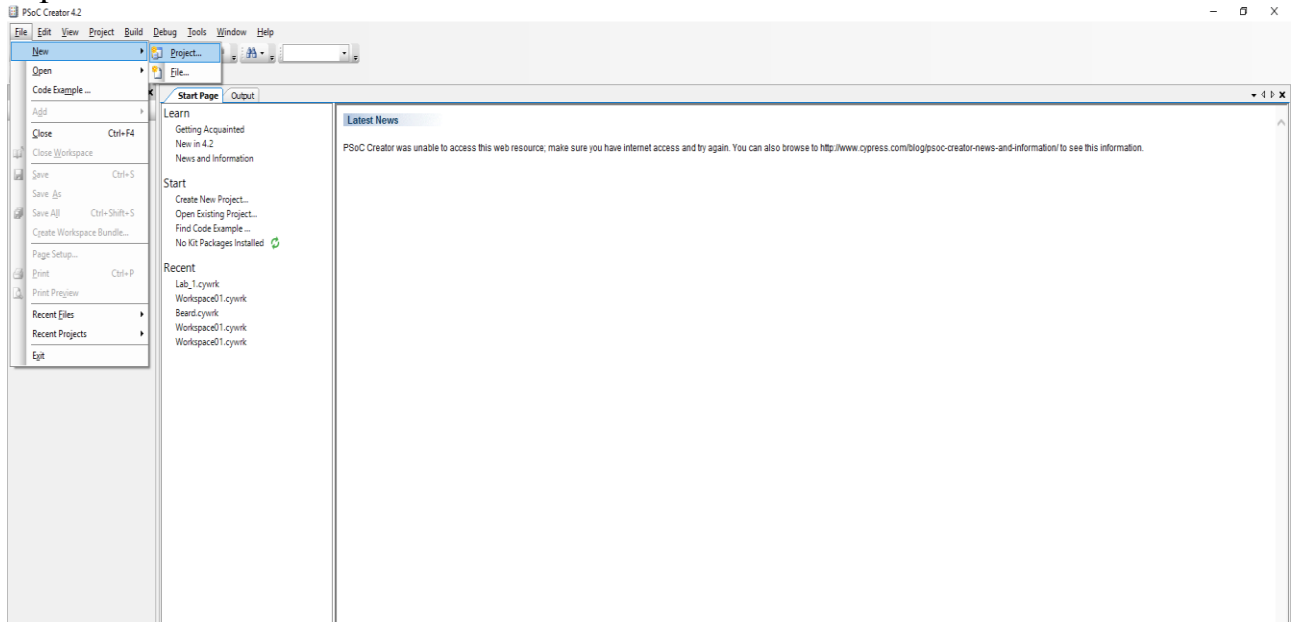


Рис. 2.2. Вигляд головного вікна IDE PSoC Creator

### ***Workspace/Project***

Інтегроване середовище розробки PSoC Creator (IDE) надає два контейнери для керування елементами у проектах: робочі області та проекти.

*Робоча область* є контейнером верхнього рівня в PSoC Creator. Контейнер містить один або кілька проектів, які можна відкрити, закрити та зберегти разом. Для кожного заданого файлу робочого простору PSoC Creator (\*.cywrk) можна мати тільки одну робочу область.

*Проект* містить декілька елементів, які представляють вашу розробку, такі як схеми, ресурси, що використовуються в розробці, вихідний код і файли з інформацією в Нех форматі. Типи елементів, які використовуються в проекті, залежать від призначення проекту. Проект завжди є частиною робочого простору. Можна створювати проекти в існуючій робочій області або створювати нову робочу область як частину створюваного нового проекту.

Середовище розробки PSoC Creator надає папки робочого простору для організації зв'язаних між собою проектів у групи, а потім виконує дії на цими групами проектів. Можна переглядати та керувати робочим простором, проектами та пов'язаними з ними елементами за допомогою провідника робочої області (Workspace Explorer). Робоча область та проекти дозволяють використовувати середовище розробки наступним чином:

- Управляти налаштуваннями для робочого простору в цілому або для окремих проектів.
- Використовувати Workspace Explorer, щоб обробляти деталі управління файлами, зосереджуючись на елементах, що утворюють вашу розробку.
- Додавати елементи, які корисні для декількох проектів в робочій області або в робочій області без посилань на елемент в кожному проекті.
- Працювати з різними файлами, незалежними від робочої області або проектів.

### *Типи проектів.*

Середовище розробки PSoC Creator надає можливість створювати і працювати з двома типами проектів: проектом і бібліотеками.

- *Дизайн-проект* використовується для створення та модифікації розробки. В дизайн-проекті необхідно вибрати та налаштувати компоненти для пристрою у вигляді схеми. Потім налаштувати загальнодоступні ресурси, такі як тактові імпульси і переривання. Потім написати код мовою C для програми. Нарешті, зібрати (і налагодити) проект, щоб згенерувати hex- файл і виконати програмування мікроконтролера. При першому створенні дизайн-проекту, PSoC Creator створює файли проекту/робочу область і структуру каталогів, а також схему верхнього рівня, файл оболонки main.c і файл ресурсів для всього проекту (.cydwr).

- *Бібліотечний проект* - це збірка одного або декількох компонентів і зв'язаного вихідного коду. З допомогою бібліотечного проекту, можна реалізувати компоненти, які будуть використані в розробці, а також повторно використані в багатьох проектах. Бібліотечні проекти також можуть використовуватися для створення статичних бібліотек, які можуть бути пов'язані з розробкою. Розробка компонентів включає створення графічного символу, визначення параметрів та визначення вимог перевірки. При першому створенні бібліотечного проекту PSoC Creator створює файли проекту/робочої області та структуру каталогів. Бібліотечні проекти входять до складу PSoC Creator як залежність, щоб визначити, які компоненти доступні для ваших проектів у каталозі компонентів.

### *Компонент/екземпляр.*

Компонент - це сукупність файлів, таких як символ, схеми, API-інтерфейси та документація, що визначають функціональні можливості у пристрої PSoC. Приклади компонентів включають таймер, лічильник та мультиплексор. Екземпляр - це компонент, вибраний із каталогу компонентів та використаний у проекті. Ви можете мати кілька копій (або екземплярів)

компонента в проєкті, якщо вибраний пристрій може підтримувати його. Ви також можете створювати та повторно використовувати власні компоненти/екземпляри.

Файли, що становлять компонент/екземпляр, включають наступне:

- *Символ* містить основне визначення компонента. Він містить зображення верхнього рівня, показане в каталозі компонентів, а також визначення параметрів. У компонента може бути лише один символ.

- *Схема* визначає, як компонент був реалізований візуально. Схема може бути загальною для будь-якого пристрою PSoC або може залежати від архітектури, сімейства та/або пристрою.

- *API* - інтерфейс програмування додатків. API- інтерфейси визначають, як взаємодіяти з компонентом, використовуючи код, реалізований мовою C. Вони можуть бути загальними для будь-якого пристрою PSoC або можуть залежати від архітектури, сімейства та/або конкретного пристрою.

- *Verilog* - Verilog може використовуватися для визначення функціональності компонента, реалізованого мовою Verilog. На будь-якому заданому рівні компонента буде тільки один файл Verilog. Файли Verilog, знайдені на різних рівнях компонента, наприклад, на рівні архітектури, сімейства та пристроїв, можуть не посилатися один на одного.

- *Файл управління* містить директиви для модуля генерації коду.

Для створення нових проєктів в середовищі розробки PSoC Creator використовується майстер створення нового проєкту.

Цей майстер створення нового проєкту використовується для:

- вибору комплекту, модуля або пристрою;
- вибору типу проєкту для його створення;
- задання назви та місця збереження проєкту;
- вибору, чи створити нову робочу область чи додати проєкт до вже існуючої робочої області.

*Приклади створення нового проєкту в IDE PSoC Creator 4.2.*

У меню File потрібно вибрати команду New → Project (рис. 2.2). Після цього потрібно вибрати зі спадних меню пристрій для проєкту (рис. 2.3).

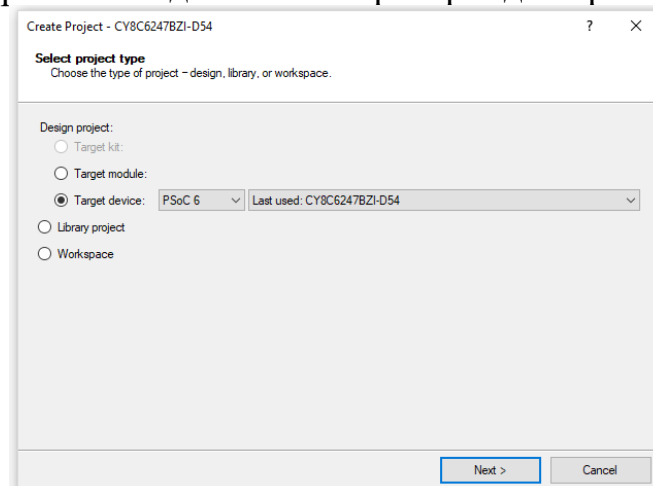


Рис. 2.3. Вигляд вікна вибору пристрою (мікроконтролера) для проєкту



*Вибір типу проекту.*

Після цього потрібно зі спадного меню (рис. 2.3) вибрати одне з наступних:

- *Комплект* (стенд) для реалізації проекту: Ця опція використовується для вибору конкретного або останнього використовуваного комплекту.

- *Модуль* для реалізації проекту. Ця опція використовується для вибору конкретного модуля або останнього використовуваного модуля.

- *Пристрій* для реалізації проекту. Цей параметр використовується для вибору сімейства і серії пристроїв, останнього використовуваного пристрою.

- *Бібліотечний проект*. Ця опція вибирається для створення бібліотечного проекту.

- *Робоча область*. Ця опція вибирається для створення пустої робочої області.

Вибираємо пристрій для реалізації проекту PSoC 6 (CY8C6347BZI-BLD53) згідно рис. 2.3 та натискаємо на кнопку 'Next'.

*Вибір шаблону проекту.*

Потрібно вибрати один з наступних варіантів:

- *Приклад коду*. Цей варіант дозволяє вибрати стартовий проект зі списку доступних прикладів.

- *Раніше реалізовані схеми*. Вибирається типова функція мікроконтролера (наприклад, UART, ADC і т.п.)

- *Пуста схема*. Цей параметр вибирається для створення шаблону пустої схеми (опція недоступна для PSoC BLE проектів).

Вибираємо опцію "Пуста схема" для нашого шаблону проекту та натискаємо на кнопку "Next":

Тільки для пристроїв PSoC 6 крок вибір середовища розробки для проекту дозволяє вибрати один або декілька IDEs для створення файлів (рис. 2.5). Якщо потрібно згенерувати файли для пакету CMSIS (для Eclipse або Arm MDK) з'явиться примітка з вказівкою місця розташування файлів, а також інформація про зміну імені і версії пакету.

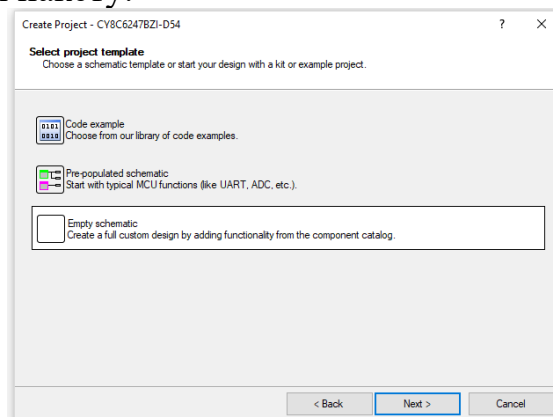


Рис. 2.4. Вигляд вікна вибору шаблону проекту

В нашому випадку не вибираємо інші середовища розробки, натискаємо на кнопку "Next" (рис. 2.5).

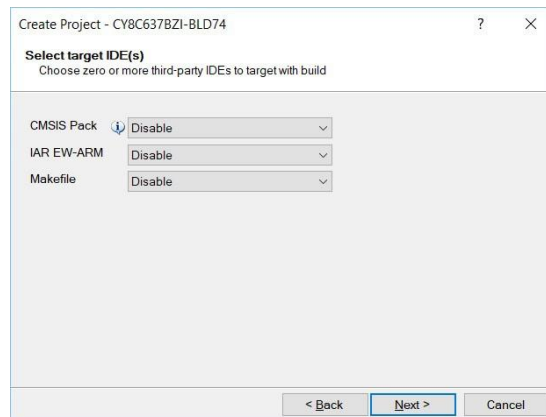


Рис. 2.5. Вигляд вікна вибору середовищ розробки  
Створення проекту. Вікно створення проекту зображено на рис. 2.6.

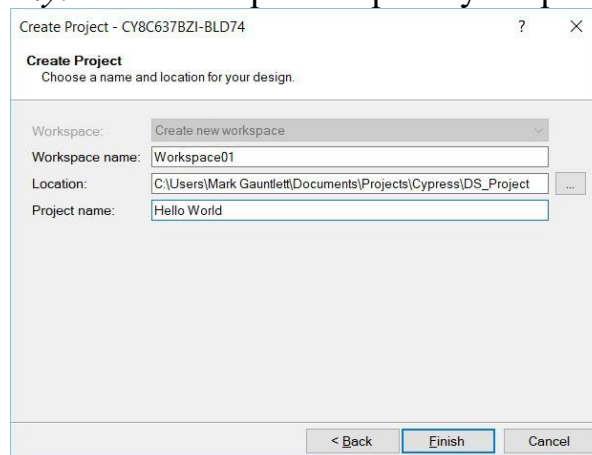


Рис. 2.6. Вигляд вікна створення проекту

*Робоча область* – потрібно вибрати відповідну опцію:

- *Створити новий робочий простір*. Цей параметр використовується для створення нового робочого простору. В цьому випадку потрібно вказати місце для зберігання цього робочого простору.

- *Додати в поточний робочий простір*. Цей параметр використовується, якщо створюється новий проект в вже відкритому робочому просторі. В цьому випадку новий проект додається в існуючий робочий простір.

*Ім'я робочого простору*. Для нового робочого простору потрібно ввести його ім'я.

*Розташування*. Потрібно вказати місце розташування для проекту/робочої області.

*Ім'я проекту*. Вводиться ім'я проекту.

*Примітки*. Ім'я проекту не може перевищувати 80 символів. При створенні пустого робочого простору на цій сторінці не буде опції "Project name".

Задаємо нашому проекту ім'я "Lab\_2" і натискаємо на кнопку "Finish" (рис. 2.6).

При цьому відкриється пустий робочий простір (рис. 2.7).

В лівій частині цього робочого простору знаходиться область "Workspace Explorer" (Провідник робочої області), в якій відображаються всі файли, пов'язані з проектом.

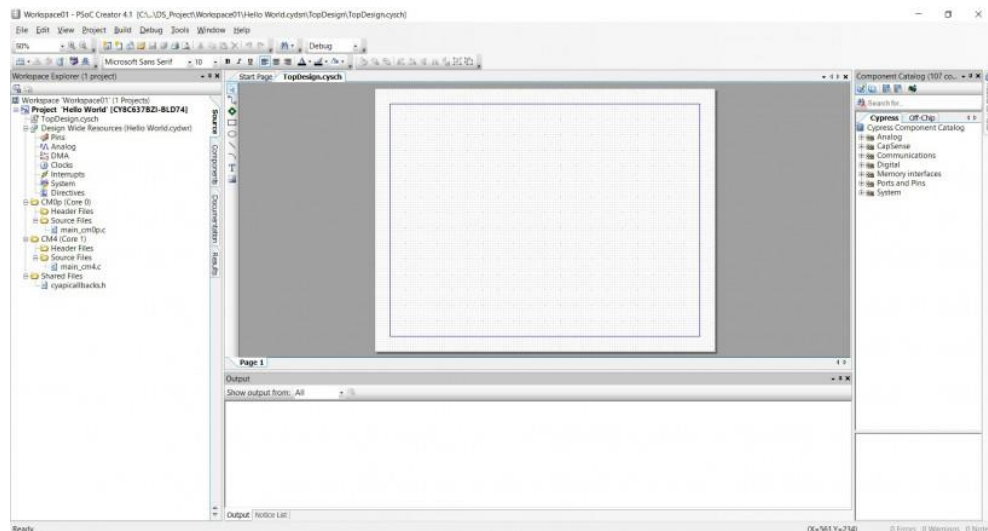


Рис. 2.7. Вигляд головного вікна IDE PSoC після створення нового проекту

У правій частині робочого простору розташований "Component Catalog" (Каталог компонентів). Він містить велику кількість (понад 100) конфігурованих компонентів, які вбудовані в структуру PSoC 6 і використовуються при проектуванні вбудованих систем.

Центральне вікно, за замовчуванням, позначене як "TopDesign.cysch", відображає графічний редактор схем, де можна розмішувати, конфігурувати та з'єднувати компоненти з "Каталогу компонентів".

В нижньому центральному вікні PSoC Creator виводиться інформація про компіляцію та системну збірку.

Переходимо в Component Catalog і вибираємо "Ports і Pins". Потім перетягуємо вивід цифрового виходу в вікно редактора схеми.

Вивід, який перетягуємо є достатньо малий. Його зображення можна збільшити клацнувши декілька разів по лупі з допомогою «+» або натиснути на кнопку [Ctrl] і лівою кнопкою мишки перетягнути рамку навколо області, яку хочемо збільшити.

Тепер у нас є зображення, яке ми можемо побачити (рис. 2.8). Одне клацання по компоненті виділить її зеленим кольором.



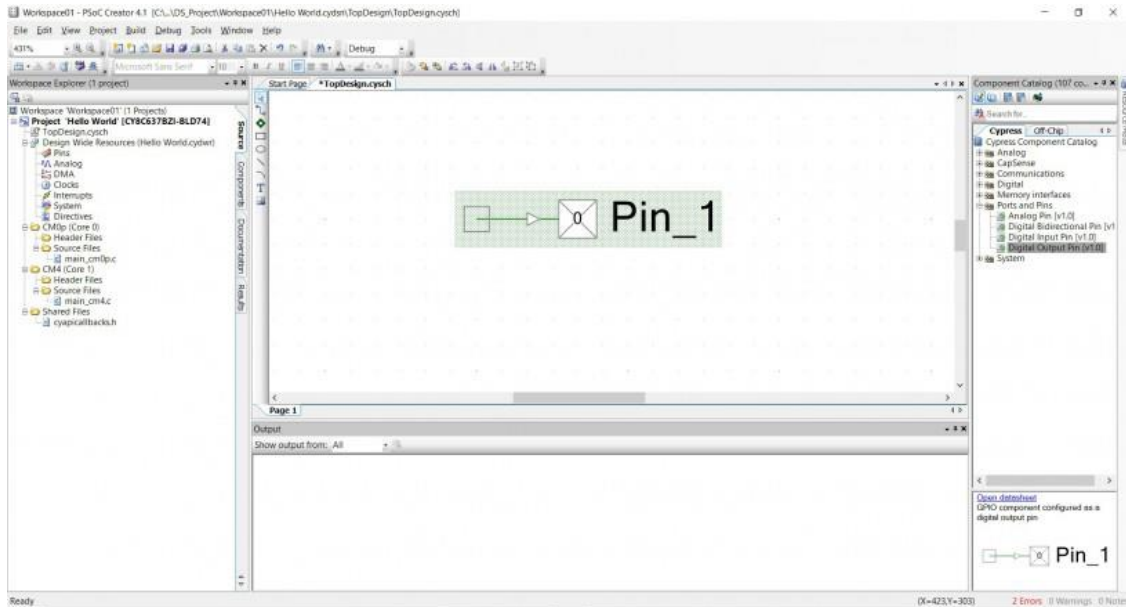


Рис. 2.8. Вигляд вікна графічного редактора з цифровим виводом

Подвійне клацання по компоненті дозволяє відкрити вікно конфігурації компоненти (рис. 2.9). Також для цього можна клацнути правою кнопкою мишки і вибрати в контекстному меню "Налаштувати".

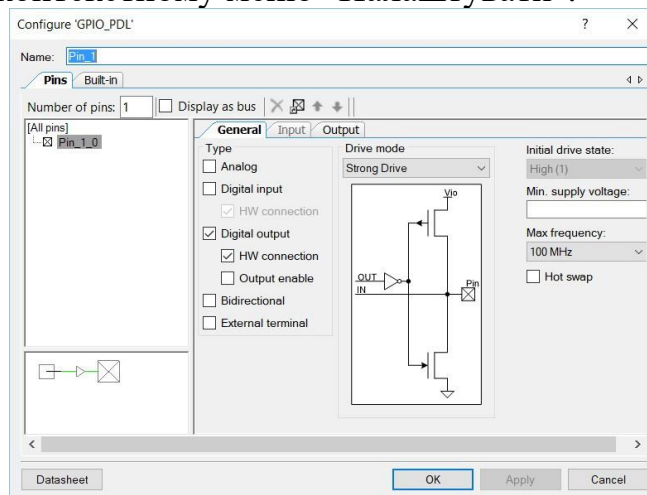


Рис. 2.9. Вигляд вікна конфігурації компоненти цифрового виводу (GPIO\_PDL)

Під'єднаємо цей вивід до синього світлодіода на стенді PSoC 6 BLE. Також змінимо назву контакту на "LED\_Blue". Для керування світлодіодом будемо використовувати мікропрограму PSoC, а не пряме апаратне з'єднання. Для цього потрібно встановити прапорець в позиції "External terminal" та зняти його для HW connection (рис. 2.10).

Після натискання на кнопку "OK" назва виводу та його форма будуть змінені. Якщо тепер подивитись на каталог компонентів, то побачимо, що існує друга вкладка з іменем "Off-Chip". Ці компоненти не мають фізичного впливу на систему, яку ми розробляємо. Вони дозволяють нам виявити, до чого під'єднуються наші зовнішні контакти. Додамо ці зовнішні контакти в нашу систему, перетягуючи резистор з меню "Passives" (пасивні компоненти) та світлодіод з меню "Diodes" (діоди), "землю" з меню "Power" (живлення) (рис. 2.11).

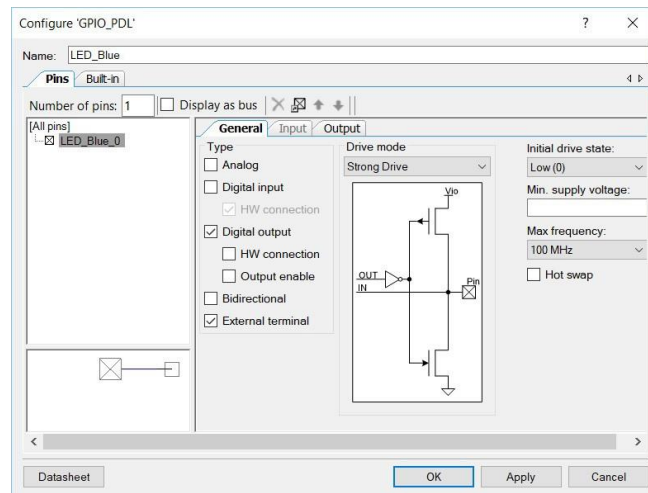


Рис. 2.10. Вигляд вікна конфігурації компоненти цифрового виводу після налаштування

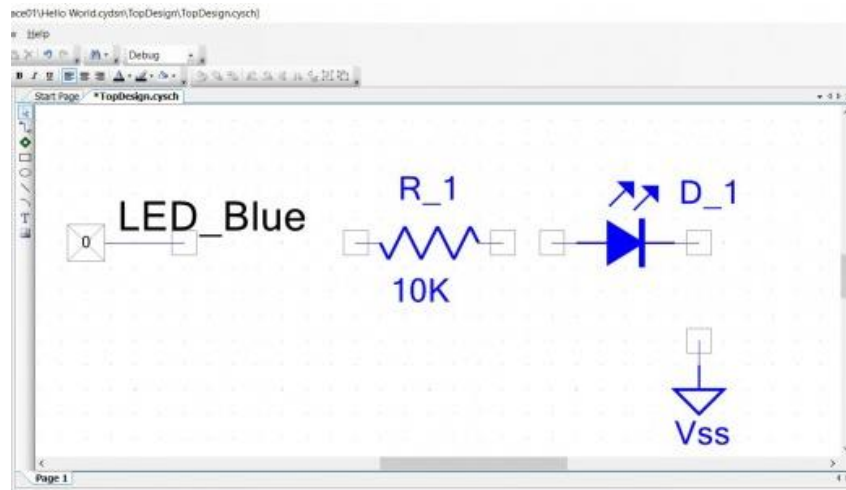


Рис. 2.11. Вигляд вікна графічного редактора з компонентами схеми на ньому  
З'єднаємо ці компоненти на схемі. Для цього можна скористатися спеціальним інструментом або використати кнопку "W".

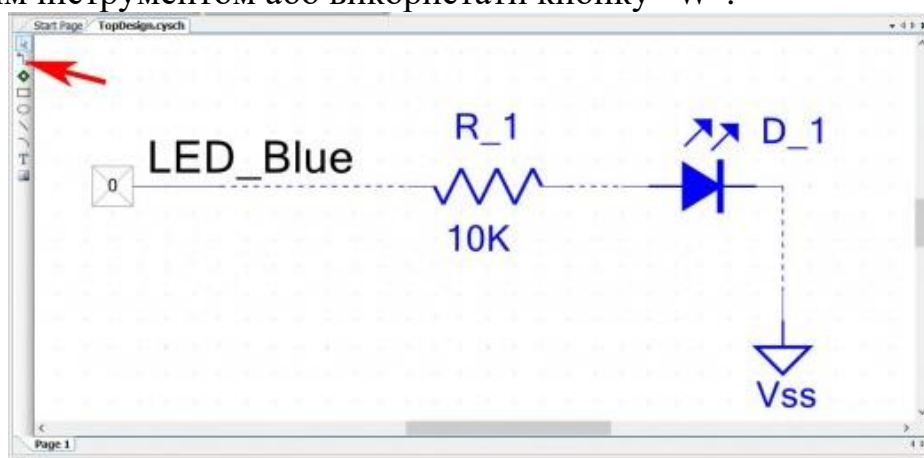


Рис. 2.12. Вигляд вікна графічного редактора зі схемою

Далі потрібно вказати системі до якого виводу на макеті під'єднаний синій світлодіод. Цю інформацію можна отримати, подивившись на напис на стенді

біля світлодіода RGB. З напису бачимо, що В (синій світлодіод) під'єднаний до порту 11, контакт 1; червоний світлодіод – до порту 0, контакт 3 та зелений світлодіод – до порту 1, контакт 1. Цю інформацію можна також отримати з документації на стенд PSoC 6.

Для призначення виводу мікроконтролера синьому світлодіоду в нашій схемі проекту, потрібно перейти в Workspace Explorer і двічі клацнути на "Pins" в розділі Design Wide Resources. Це призведе до появи карти виводів пристрою PSoC 6 (рис. 2.13).



Рис. 2.13. Вигляд карти виводів мікроконтролера PSoC 6

З рис. 2.13 бачимо, що вивід LED\_Blue вже знаходиться в списку, який можна назначати. Скористаємося меню Port, щоб знайти P11 [1] в списку і вибрати його. Наш вивід тепер виділений на карті BGA (рис. 2.14), і бачимо, що він відображається на фізичний вивід E5 BGA.



Рис. 2.14. Вигляд карти виводів мікроконтролера PSoC 6 з назначеним виводом

Після цього будуємо нашу систему. Для цього потрібно зайти в меню "Build" і натиснути Build Lab\_1. Певна інформація почне виводитися в вікно виводу Output. В кінці виводу нам бажано отримати наступну інформацію: "---- Build Succeeded: *date time* ----", де "*date*" і "*time*" замінюються поточними датою і часом.

### Програмування PSoC 6.

Апаратна частина системи реалізована. Тепер переходимо до програмної частини проекту. Реалізуємо невелику програму мовою програмування C. Використаємо ядро ARM® Cortex® M4. В такій же мірі ми могли скористатися ядром ARM® Cortex® M0+. Перейдемо в Workspace Explorer і двічі клацнемо

## Лабораторний практикум - Мікроконтролери (ч. II)

на main\_cm4.c. Базовий каркас програми відкриється в центральному вікні IDE PSoC 6 Creator (рис. 2.15).

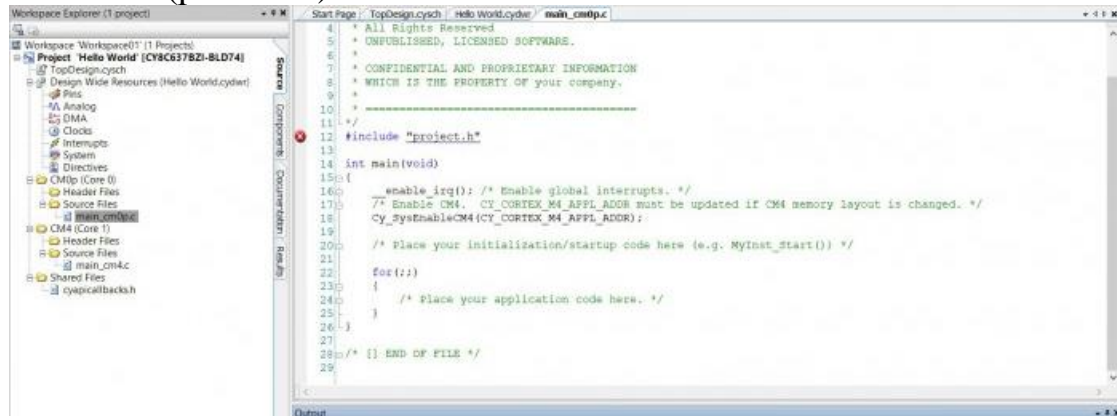


Рис. 2.15. Вигляд каркасу програми main\_cm4.c

До цієї програмної заготовки додамо фрагменти нашої програми. Кінцевий варіант програми зображений на рис. 2.16.

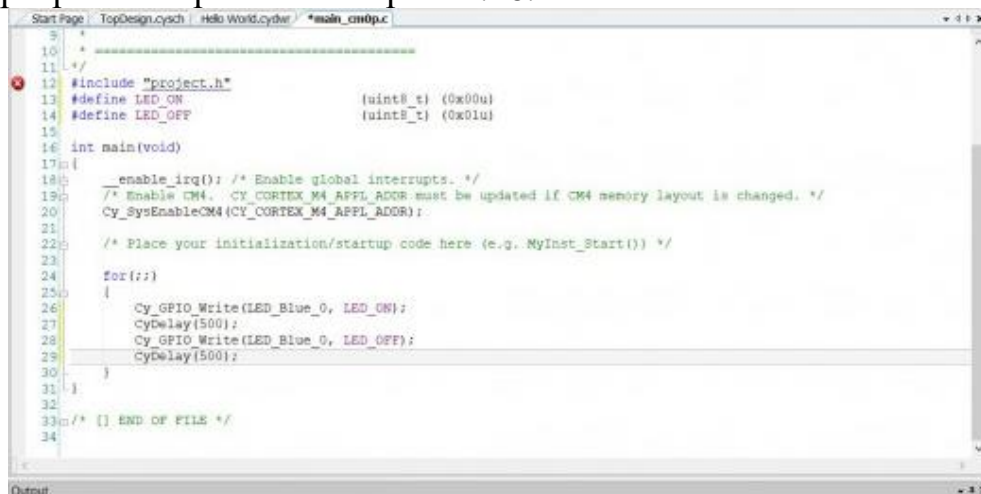


Рис. 2.16. Вигляд програми main\_cm4.c

Реалізована програма використовує функцію запису GPIO для запису значень ON та OFF на вивід світлодіода LED. Між кожним переключенням додаємо затримку 500 мс. Переключення з такою частотою ( $F_{\text{цик}}=2$  кГц) добре побачить око людини.

*Запуск програми на стенді PSoC 6.*

Під'єднаємо стенд PSoC 6 з допомогою кабелю USB до комп'ютера. Заходимо в меню Debug > Program (або натискаємо на кнопку Ctrl + F5, або натискаємо на піктограму програмування) і чекаємо поки програма скомпілюється та завантажиться в мікроконтролер. Бачимо блимання синього світлодіода на стенді.

*Реалізація проекту з використанням ШІМ.*

Для засвічування світлодіода тепер будемо використовувати ШІМ, а не ядро процесора. Можна знайти його, ввівши "pwm" в полі пошуку "Search for.." в верхній частині вікна Component Catalog. В результаті цього перейдемо до



## Лабораторний практикум - Мікроконтролери (ч. II)

каталогу Digital>Functions>PWM (TCPWM)[v1.0]. Перетягнемо компоненту ШІМ в проект та один з генераторів (рис. 2.17).

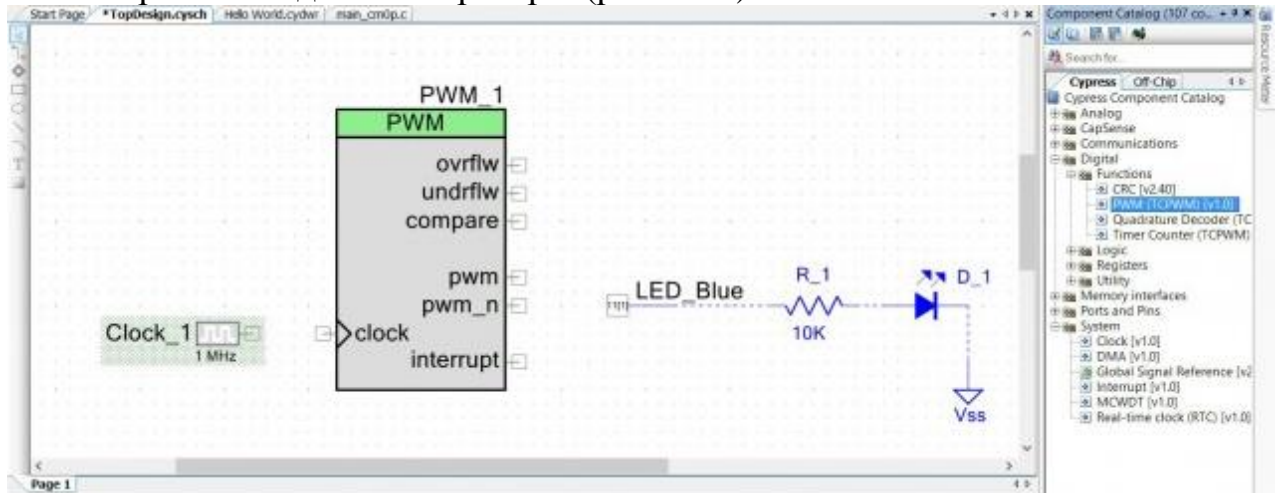


Рис. 2.17. Вигляд вікна з компонентами схеми PWM\_1, Clock\_1

Для кожної з компонент Component Catalog можна взяти, які функції API є доступними. Вибравши, наприклад, ШІМ можна побачити вікно опису в правому куті під вікном Component Catalog. Це вікно містить також посилання на datasheet компоненти в форматі PDF, помічене як Open datasheet. Файл Datasheet містить опис з'єднань вводу/виводу, функціональні параметри і характеристики компонент. В описі також є розділ API, в якому описані функції оболонки компоненти, доступні в даній частині.

### *Налаштування компонент.*

Перед використанням нових компонент в нашій схемі їх потрібно налаштувати. Спочатку змінимо налаштування виводу цифрового виходу. Для відкриття його вікна конфігурації двічі клацнемо на ньому (рис. 2.18).

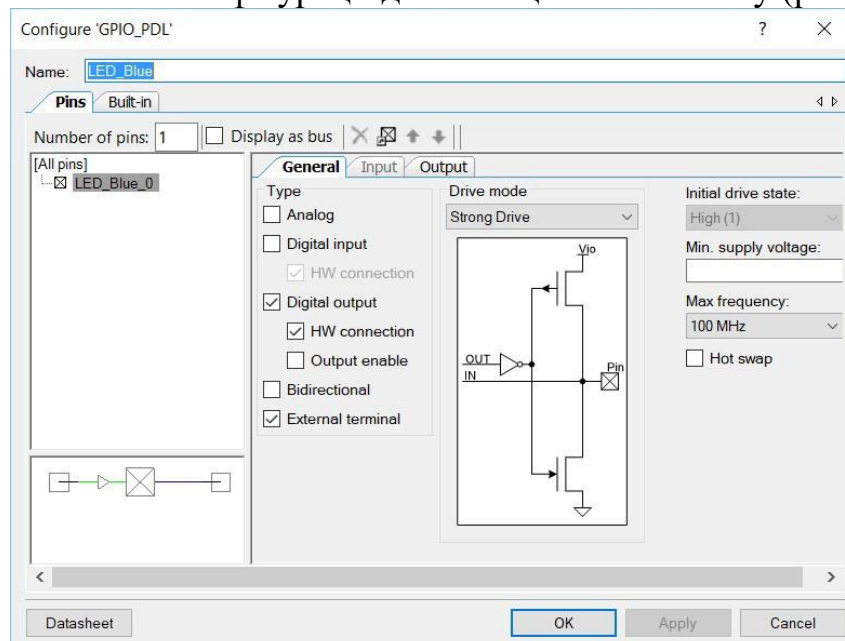


Рис. 2.18. Налаштування компоненти цифрового виводу

Потрібно знову включити прапорець "HW connection", щоб можна було під'єднувати інші внутрішні компоненти до виводу. Коли натиснемо кнопку

"OK" (рис. 2.18), то побачимо що вивід змінився. У нього є поле вводу, до якого можна під'єднатися.

Перейдемо до налаштування ШІМ. Двічі клацнемо на ньому. Відкриється його вікно конфігурації (рис. 2.19). Так як в схемі проекту у нас тільки один ШІМ, то перейменуємо його як "PWM". У вкладці "Basic" перемістимо смугу прокрутки вниз і встановимо період 1000 мсек, його робочий цикл задамо 50 мсек. Це надасть засвічуванню синього світлодіоду більш стробуючого ефекту.

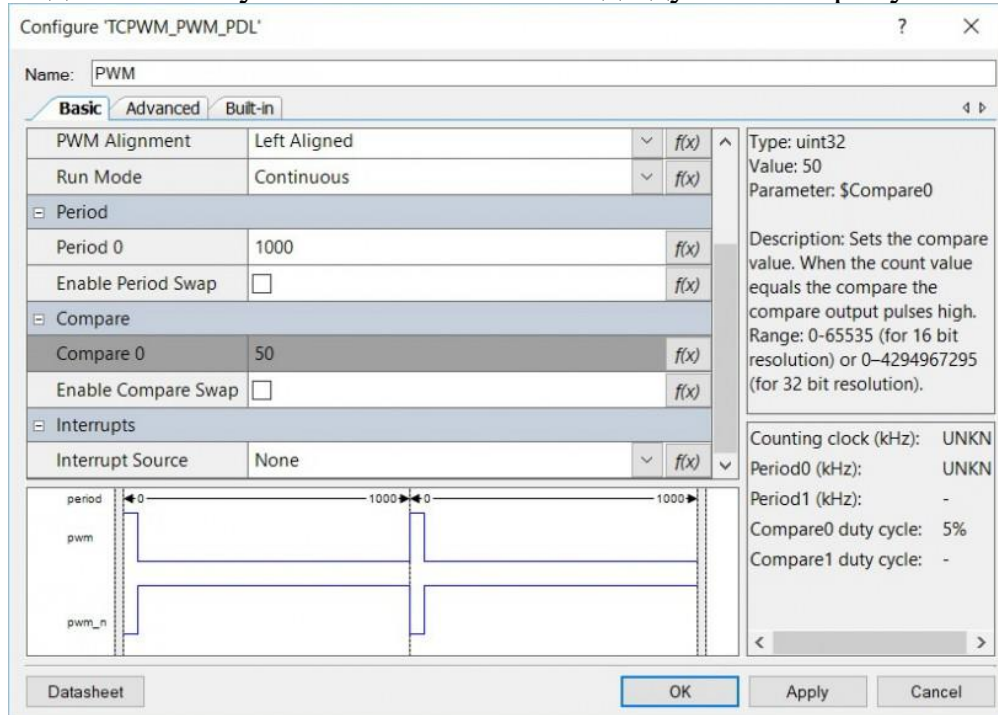


Рис. 2.19. Вигляд вікна компоненти PWM, після її налаштування  
Після цього налаштуємо наш генератор на частоту 1 кГц (рис. 2.20).

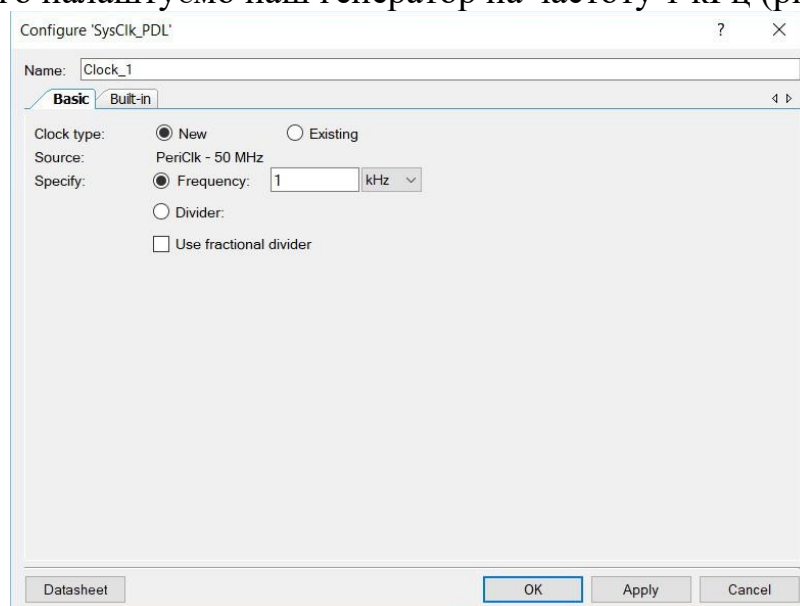


Рис. 2.20. Вигляд вікна компоненти Clock\_1, після її налаштування  
Реалізуємо з'єднання в схемі нашого проекту. Генератор під'єднаний до входу clock ШІМ і вихід ШІМ (pwm\_n) – до цифрового виходу (рис. 2.21).



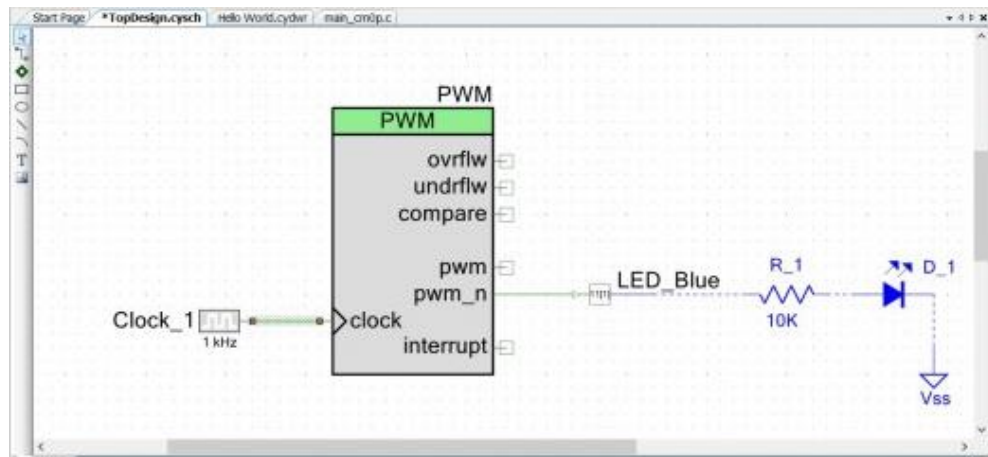


Рис. 2.21. Вигляд досліджуваної схеми

Побудуємо наш проект. Для цього можна скористатися меню Build > Build Lab\_1, або просто натиснути [shift] + [F6], або натиснути на піктограму Build.

Змінимо програму для цього проекту. Основну програмну роботу за нас виконав PSoC 6 Creator Syress. Наші доповнення в програму – це один рядок коду мовою C – виклик функції PWM\_start() (рис. 2.22).

```
Start Page: TopDesign.cysch Hello World.cydw main_cm0p.c
5  UNPUBLISHED, LICENSED SOFTWARE.
6
7  CONFIDENTIAL AND PROPRIETARY INFORMATION
8  WHICH IS THE PROPERTY OF your company.
9
10
11  */
12  #include "project.h"
13
14  int main(void)
15  {
16      enable_irq(); /* Enable global interrupts. */
17      /* Enable CM4. CY_CORTEX_M4_APPL_ADOR must be updated if CM4 memory layout is changed. */
18      Cy_SysEnableCM4(CY_CORTEX_M4_APPL_ADOR);
19
20      PWM_Start();
21
22      for(;;)
23      {
24      }
25
26  }
27
28  /* [] END OF FILE */
29
30
```

Рис. 2.22. Вигляд коду для засвічування світло діоду з використанням PWM

Скомпілюємо і завантажимо програму в PSoC 6. Для цього достатньо зайти в меню Debug> Program або натиснути на кнопки [Ctrl] + [F5]. В цьому випадку світлодіод блимає по іншому.

### Завдання:

1. Реалізувати проект засвічування синього RGB LED стенду з використанням схеми, зображеної на рис. 2.12. Світлодіод засвічується на протязі 1.5 сек і не світиться на протязі 1 сек.
2. Реалізувати проект засвічування червоного RGB LED стенду з використанням схеми, зображеної на рис. 2.21. Світлодіод засвічується на протязі 0.75 сек і не світиться на протязі 0.5 сек.
3. Реалізувати проект послідовного засвічування синього та зеленого RGB LED стенду з частотою 2 Гц.

### Література.

1. PSoC® Creator™ User Guide. Електронний ресурс. – Режим доступу: <https://www.cypress.com/file/137441/download>
2. PSoC® Creator™ Quick Start. Електронний ресурс. – Режим доступу: <https://www.cypress.com/file/44826/download>
3. AN213924. PSoC 6 MCU Device Firmware Update Software Development Kit Guide. Електронний ресурс. – Режим доступу: <https://www.cypress.com/file/385706/download>