Лабораторна робота № 4

Вивчення PWM мікроконтролерів PSoC 6 фірми Cypress

Мета роботи: Ознайомитися з компонентою PWM мікроконтролерів PSoC 6 фірми Cypress; основними її застосуваннями та параметрами налаштування.

Теоретичні відомості

Особливості компоненти TCPWM_PWM_PDL:

- 16-ти або 32-х бітний лічильник.
- Два програмованих регістри періоду, які можна міняти місцями.
- Два вихідних регістри порівняння, які можна міняти місцями при переповненні і/або до переповнення.
- Режими вирівнювання по лівому краю, вирівнювання по правому краю, вирівнювання по центрі и асиметричне вирівнювання.
- Режими неперервного або однократного виконання.
- Псевдовипадковий режим.
- Два ШІМ- виходи з вставкою ″мертвого″ часу (Dead Time) та програмованою полярністю.
- Переривання та вихід при переповненні, пониженні або порівнянні.
- Входи старт (Start), перезавантаження (Reload), очистки (Kill), обміну (Swap) та підрахунку (Count).
- Декілька компонент можуть бути синхронізовані разом для таких додатків як управління трифазним двигуном.
- Компонента бібліотеки периферійних драйверів (PDL - Peripheral Driver Library) (тільки інтерфейс прикладного програмування (API - Application Programming Interface) PDL).

Загальний опис.

Компонента TCPWM_PWM_PDL є графічним об’єктом, що конфігурується, створеним як верхній рівень драйвера cy_tcpwm, доступного в PDL. Вона допускає з’єднання на основі схеми і апаратну конфігурацію, яка визначена в діалоговому вікні ″Component Configure″.

Компонента TCPWM_PWM_PDL є обгорткою навколо апаратного елементу TCPWM, яка дозволяє налаштовувати апаратний елемент TCPWM для функції ШІМ. З допомогою неї можна синтезувати довільні цифрові сигнали та контролювати робочий цикл (шпаруватість) і період вихідного сигналу TCPWM_PWM_PDL. Компонента TCPWM_PWM_PDL також забезпечує додатковий вихід з можливістю вставки мертвого часу між двома виходами.

Компонента TCPWM_PWM_PDL має декілька параметрів вирівнювання: по лівому краю, по правому краю, по центру і асиметричне. У всіх режимах період та величина, що порівнюється, можуть бути поміняні місцями для створення сигналу довільної форми. Компонента також має псевдовипадковий режим роботи.

Компонента TCPWM_PWM_PDL використовується у випадках, коли потрібно на її виході отримати імпульси прямокутної форми з певним періодом та робочим циклом. Наприклад:

- управління швидкістю обертання двигунів постійного струму;
- управління світлодіодами (LED), зміна яскравості їх свічення.

Швидкий старт роботи з компонентою TCPWM_PWM_PDL.

1. Перетягнути компоненту TCPWM_PWM_PDL з папки Cypress/Digital/ Function/ в каталозі компонентів на схему проекту (екземпляр цієї компоненти приймає ім'я PWM_1).
2. Двічі клацнути на екземплярі компоненти, щоб відкрити діалогове вікно налаштування (Configure) і налаштувати параметри компоненти.
3. В цій компоненті немає внутрішніх джерел тактових імпульсів. Тому потрібно подати на вхід TCPWM_PWM_PDL тактові імпульси з зовнішнього джерела. В компоненті TCPWM_PWM_PDL доступна функціональність попереднього масштабування тактової частоти, щоб додатково виконати ділення тактової частоти.
4. Побудувати проект. Для того, щоб перевірити правильність дизайну, потрібно додати необхідні модулі PDL в Workspace Explorer та згенерувати задану конфігурацію для екземпляра PWM_1.
5. В файлі main.c, ініціалізувати периферійні пристрої та запустити додаток за допомогою API драйверів та макросів компонентів-препроцесорів.

Якщо не використовується жоден вхідний термінал (start), для початку підрахунку потрібно викликати програмну подію Cy_TCPWM_TriggerStart або Cy_TCPWM_TriggerReloadOrIndex. Наприклад:

```
(void) Cy_TCPWM_PWM_Init(PWM_1_HW, PWM_1_CNT_NUM, &PWM_1_config);
Cy_TCPWM_Enable_Multiple(PWM_1_HW, PWM_1_CNT_MASK);
Cy_TCPWM_TriggerStart(PWM_1_HW, PWM_1_CNT_MASK);
```

6. Виконати компіляцію та запрограмувати мікроконтролер PSoC 6 стенду.

Вхідні/вихідні виводи компоненти.

Розглянемо входи та виходи компоненти TCPWM_PWM_PDL. Зірочка (*) у наведеному нижче списку вказує, що вона може не відображатися на символі компонента за умов, перелічених в описі цього вводу/виводу.

- clock (Digital Input) - вхід тактових імпульсів, визначає робочу частоту цієї компоненти. Період виводу ШІМ дорівнює 1/(тактова частота * період).
- swap [1]* (Digital Input) – цей вхід фіксує команду SWAP. Він є видимим тільки у випадку, якщо для параметра SwapInput встановлено значення, відмінне від від'єднаного. Обмін не відбувається до наступної події ov/un. Потрібно переконатися, що період замінюється тільки на подію OV.
- reload[1]* (Digital Input) - цей вхід запускає перезавантаження ШІМ та запускає ШІМ. Цей вхід видимий лише в тому випадку, якщо для параметра ReloadInput встановлено значення, відмінне від від'єднаного. У режимі вирівнювання по лівому краю, лічильник ініціалізується значенням ′0′. Для режимів центрування/асиметричного вирівнювання лічильник ініціалізується

значенням ′1′. У режимі вирівнювання по правому краю лічильник ініціалізується значенням періоду. Потрібно використовувати тільки тоді, коли лічильник не працює.

- count* (Digital Input) - цей вхід змушує ШІМ підраховувати залежно від того, як він налаштований. Наприклад, якщо цей вхід налаштований як чутливий до рівня, ШІМ змінюватиме свій підрахунок на кожний попередньо масштабований фронт тактових імпульсів. Цей вхід є видимим, лише якщо для параметра CountInput встановлено значення, відмінне від від'єднаного. Не використовується і не відображається для режиму Pseudo Random PWM.
- start [1]* (Digital Input) - цей вхід запускає ШІМ. Він видимий тільки в тому випадку, якщо для параметра StartInput встановлено значення, відмінне від від'єднаного. Використовується лише тоді, коли лічильник не працює.
- kill [1]* (Digital Input) – цей вхід ″вбиває″ PWM, базуючись на виборі режиму Kill. Він видимий тільки в тому випадку, якщо для параметра KillInput встановлено значення, відмінне від від'єднаного або Stop on Kill.
- ovrflw* (Digital Output) – цей вихід приймає значення логічної ′1′, коли значення лічильника переповнюється від періоду до нуля. Перезавантаження також викличе переповнення прт вирівнюванні по лівому краю. Не використовується і невидимий для псевдовипадкового режиму ШІМ.
- undrflw* (Digital Output) - цей вихід приймає значення логічної ′1′, коли значення лічильника переходить з нуля до значення періоду. Перезавантаження також генерує подію недостатнього заповнення в режимах правого, центрального та асиметричного вирівнювань. Не застосовується і не відображається для псевдовипадкового режиму ШІМ.
- compare (Digital Output) - цей вихід приймає значення логічної ′1′, коли величина порівняння рівна значенню підрахунку (ця подія генерується, коли повинно відбутися співпадіння (COUNTER не рівний COMPARE і змінюється на COMPARE, в режимах вирівнювання по лівому краю або по правому краю) або коли співпадіння не повинно відбутися (COUNTER рівний COMPARE і змінюється на інше значення в режимах центрування/асиметричного вирівнювання)).
- pwm (Digital Output) – вихід ШІМ.
- pwm_n (Digital Output) – доповнюючий вихід ШІМ.
- ìnterrupt (Digital Output) – цей вихід можна під'єднати тільки до переривання.

1. Вхідна подія вступає в силу на наступних тактових імпульсах лічильника, якщо подія підрахунку стає активною (це залежить від режиму детектування фронту події підрахунку). Наприклад, якщо використовується зовнішній сигнал підрахунку і зовнішній сигнал запуску компоненти, перший активний сигнал підрахунку не буде рахуватися, він використовується для запуску лічильника. Тому значення лічильника буде на одиницю меншим.

Параметри компоненти,

Діалогове вікно ″Налаштування компоненти″ TCPWM_PWM_PDL дозволяє редагувати параметри конфігурації для екземпляру компоненти.

Основна вкладка.

Ця вкладка містить параметри компоненти, які використовуються в загальних налаштуваннях ініціалізації периферійних пристроїв.

- PwmMode – вибір режиму роботи ШІМ.
- ClockPrescaler – ділення вхідних тактових імпульсів.
- RunMode – якщо вибрано режим неперервного виконання, лічильник працює неперервно. Якщо ж вибрано режим однократного виконання, лічильник працює протягом одного періоду і зупиняється.
- PwmAlignment – вибір напряму, в якому виконується підрахунок ШІМ (Left = Up, Right = Down, Center = Up/Down1, Asymmetric = Up/Down2).
- Resolution – вибір ширини ШІМ.
- DeadClocks – кількість періодів тактової частоти ″мертвого″ часу між виходами ШІМ. Діапазон 0-255.
- Period0 – встановлює період лічильника. Діапазон 0-65535 - для 16- ти бітної роздільної здатності або 0–4294967295 - для 32- х бітної роздільної здатності.
- EnablePeriodSwap – якщо встановлений цей прапорець, то періоди будуть змінюватися між 0- м періодом та 1- м періодом при наступному OV/UN, коли подія обміну була зареєстрована. В центральному та асиметричному режимах періоди змінюються місцями тільки при події недостатнього заповнення.
- Period1 – встановлює період лічильника після першого обміну. Діапазон 0-65535 - для 16- ти бітної роздільної здатності або 0–4294967295 - для 32- х бітної роздільної здатності.
- Compare0 – встановлює значення порівняння. Коли значення лічильника рівне значенню порівняння, то вихідні імпульси порівняння мають високий рівень. Діапазон 0-65535 - для 16- ти бітної роздільної здатності або 0–4294967295 - для 32- х бітної роздільної здатності.
- EnableCompareSwap – при виборі регістр порівняння змінюється між порівнянням 0 та порівнянням 1 на наступному OV/UN після регістрації обміну.
- Compare1 – встановлює значення порівняння. Якщо значення лічильника рівне значенню порівняння, вихідні імпульси порівняння мають високий рівень.
- InterruptSource – вибирає, які події можуть викликати порівняння.

Вкладка розширення.

- ReloadInput – визначає чи потрібний вхід перезавантаження і як зареєстрований вхід сигналу перезавантаження.
- SwapInput – цей вхід управляє тим, коли відбувається порівняння і заміна періодів, і як цей вхід реєструється.
- KillInput – визначає, як поводить себе вхід ″знищення″ і як цей вхід зареєстрований.
- StartInput – визначає, чи потрібний вхід запуску і як цей вхід зареєстрований.
- CountInput – визначає, чи потрібний лічильний вхід і як цей вхід зареєстрований.

- KillMode – визначає, що сигнал знищення робить з ″ШІМ″.

- InvertPwm – якщо вибраний цей параметр, то головний вихід ШІМ (PWM) інвертується.

- InvertPwm_n - якщо вибраний цей параметр, то головний вихід ШІМ (PWM_n) інвертується.

Процедури інтерфейсу прикладного програмування (Application Programming Interface – API) дозволяють налаштовувати компоненту з допомогою програмного забезпечення.

За замовчуванням PSoC Creator назначає ім’я екземпляру "PWM_1" першому екземпляру компоненти в проекті. Його можна перейменувати в будь-яке унікальне ім’я, яке відповідає синтаксичним правилам для ідентифікаторів. Ім’я екземпляру стає префіксом кожного глобального імені функції, змінної та символу константи.

Ця компонента використовує модуль драйвера cy_tcpwm з PDL. Драйвер копіюється в каталог “pdl\drivers\peripheral\tcpwm\” проекту додатку після успішної компіляції.

Ця компонента генерує структури конфігурації базову адресу, необхідну в API PDL. Для налаштування периферійного пристрою потрібно передати згенеровану структуру даних і базову адресу відповідної функції драйвера cy_tcpwm в програму ініціалізації додатку. Як тільки периферійний пристрій ініціалізований, код додатку може виконувати зміни під час виконання, посилаючись на надану базову адресу в функціях API драйверу.

Глобальні змінні.

Компонента TCPWM_PWM_PDL заповнює наступну структуру даних ініціалізації периферії. Згенерований код поміщається в вихідні та заголовочні файли мови C, які мають імена пов’язані з екземпляром компоненти (наприклад, PWM_1.c). Кожна змінна також має префікс з іменем екземпляру компоненти.

cy_stc_tcpwm_pwm_config_t PWM_1_config

Структура конфігурації конкретного екземпляру. Її потрібно використовувати в функції PDL Init ().

Макроси препроцесора.

Компонента TCPWM_PWM_PDL генерує наступні макроси препроцесора. Кожному макросу передує ім’я екземпляра компоненти (наприклад, «PWM_1»).

#define PWM_1_HW (PWM_1_TCPWM__HW)

Це вказівник на базову адресу екземпляра TCPWM.

#define PWM_1_CNT_HW (PWM_1_TCPWM__CNT_HW)

Це вказівник на базову адресу екземпляра TCPWM CNT.

#define PWM_1_CNT_NUM (PWM_1_TCPWM__CNT_IDX)

Це номер екземпляру лічильника, вибраного TCPWM.

#define PWM_1_CNT_MASK (1uL << PWM_1_TCPWM__CNT_IDX)

Це бітове поле, що представляє екземпляр лічильника в вибраному блоці TCPWM.

Завдання.

1. Реалізувати проект засвічування червоного, синього та зеленого світлодіодів RGB LED стенду PSoC 6 BLE PIONER KIT з використанням схеми, зображеної на рис. 4.1. Алгоритм засвічування світлодіодів зображений на рис. 4.2.

Рис. 4.1. Схема проекту завдання 1

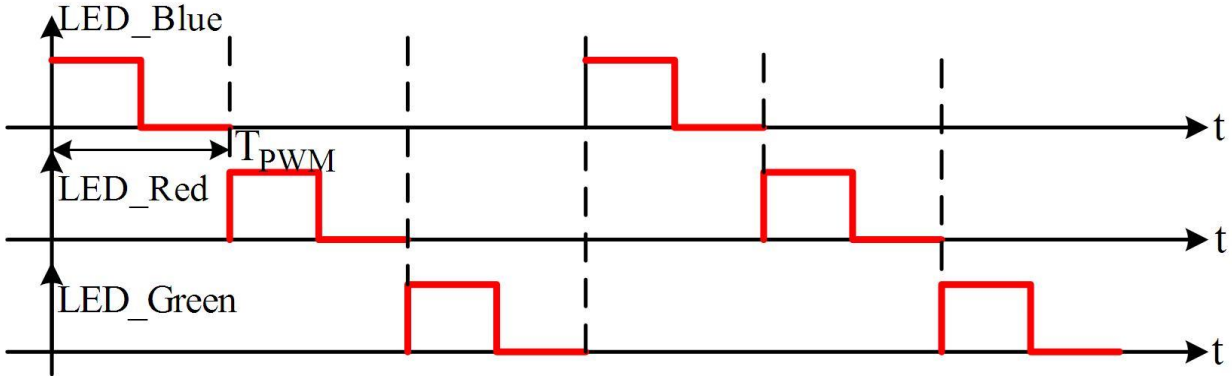

Рис. 4.2. Часові діаграми засвічування світлодіодів RGB LED стенду

2. Реалізувати проект засвічування червоного, синього та зеленого світлодіодів RGB LED стенду PSoC 6 BLE PIONER KIT з використанням схеми, зображеної на рис. 4.3. Алгоритм засвічування світлодіодів зображений на рис. 4.4.

Рис. 4.3. Схема проекту завдання 2

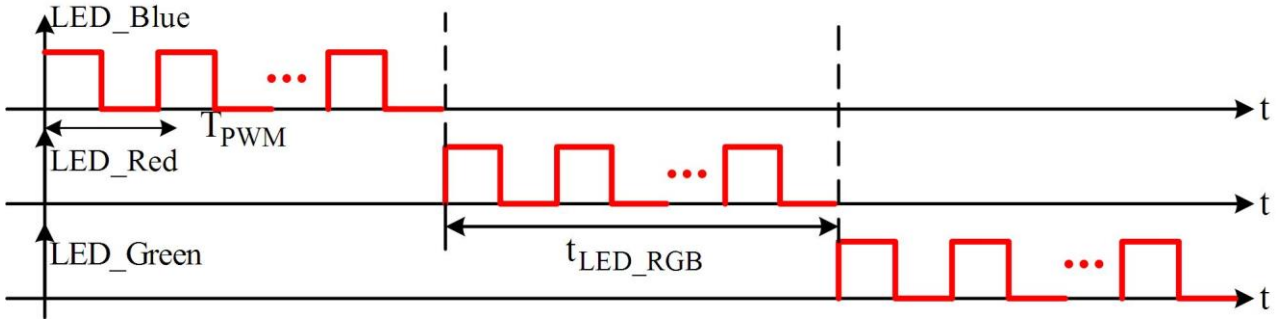

Рис. 4.4. Часові діаграми засвічування світлодіодів RGB LED стенду

3. Реалізувати проект засвічування червоного, синього та зеленого світлодіодів RGB LED стенду з використанням схеми, зображеної на рис. 4.5. Алгоритм засвічування світлодіодів зображений на рис. 4.6. При натисканні кнопки SW_2 та на протязі її утримання алгоритм засвічування зображений на рис. 4.7.

Рис. 4.5. Схема проекту завдання 3

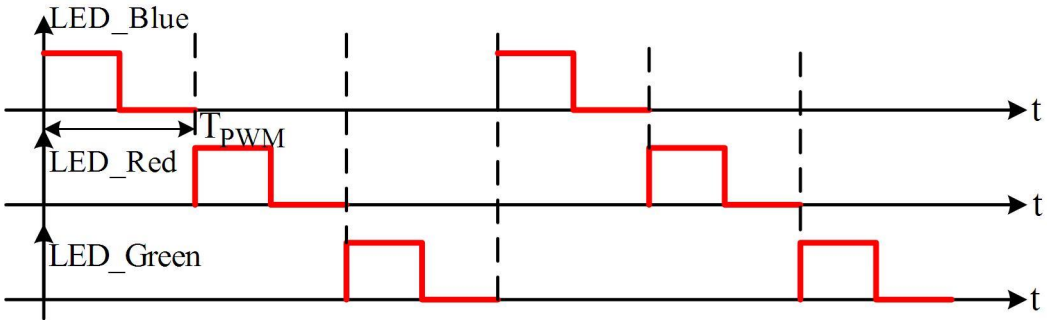

Рис. 4.6. Часові діаграми засвічування світлодіодів RGB LED стенду при ненатиснутій кнопці SW_2

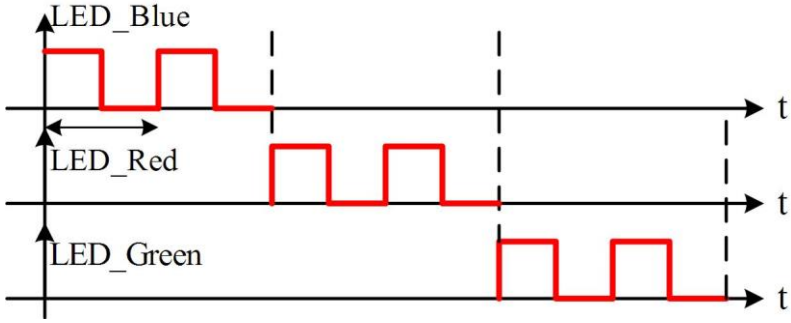

Рис. 4.7. Часові діаграми засвічування світлодіодів RGB LED стенду при натиснутій кнопці SW_2