

### Лабораторна робота № 9

#### Вивчення E-INK дисплею стенду PSoC 6 BLE Pioneer Kit фірми Cypress

**Мета роботи:** Вивчити функціонування E-INK дисплею для відображення на ньому текстової та графічної інформації.

#### Теоретичні відомості

В цьому проекті розглянемо реалізацію з допомогою компоненти DAC (12-ти розрядного) і бібліотеки периферійних драйверів CTDAC (PDL) генератор синусоїди за допомогою DMA.

Компонента DMA передає дані з таблиці пошуку в регістр значень ЦАП без участі процесора. Крім викликів функцій для ініціалізації та включення апаратного забезпечення, інші програмні функції не використовуються. Показані як 12-ти розрядна компонента ЦАП напруги PSoC Creator (12-бітний), так і базові низькорівневі виклики функцій PDL ЦАП постійного часу (CTDAC).

Схема, зображена на рис. 1, реалізує постійне оновлення вихідної напруги компоненти VDAC за допомогою DMA. Передачі з DMA ініціюється сигналом запуску, який формується після завершення попереднього оновлення. Компонента VDAC розміщується у доступному апаратному блоці CTDAC, а сигнал вихідної напруги передається на вивід P9 [6]. Осцилограф, підключений до цього виводу, відображатиме форму сигналу.

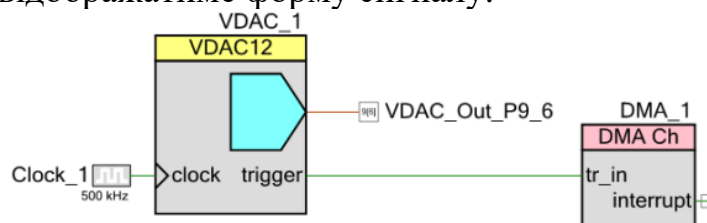


Рис. 9.1. Схема проекту генерації сигналу синусоїдальної форми

Вихід тригера VDAC повідомляє компоненті DMA, коли VDAC готовий прийняти нове значення. DMA оновлює значення VDAC з таблиці 100 записів. При входному тактовому сигналі частотою VDAC 500 кГц частота синусоїди складає 5 кГц.

За замовчуванням у прикладі коду використовуються виклики функції API- компонентів. Крім того, макрос PDL\_CONFIGURATION можна встановити на (1u), щоб продемонструвати, як використовувати низькорівневі виклики функції PDL CTDAC.

Основні етапи роботи з проектом:

1. Підключити стенд CY8CKIT-062 BLE до USB-порту комп'ютера.
2. Створити новий проект, реалізувати апаратну та програмну частини та запрограмувати його в мікроконтролер PSoC 6.

3. Підключити щуп осцилографа до виводу P9[6] (на роз'ємі J2 стенду). При правильному налаштуванні осцилографа побачимо, що сигнал є синусоїдою з частотою 5 кГц та подвійною амплітудою рівною діапазону напруги 0-VDDA (рис. 9.1).

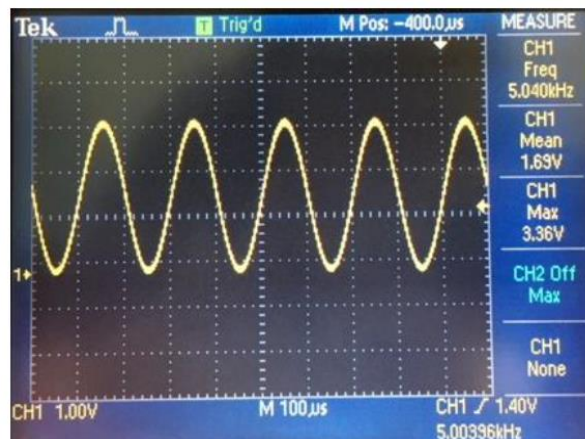


Рис. 9.1. Вигляд синусоїдального сигналу на виході VDAC

У табл. 9.1 перераховані компоненти PSoC Creator 4.2, використані в цьому проекті, апаратні ресурси, що використовуються кожним з них, та його налаштування.

Табл. 9.1. Використані компоненти в середовищі PSoC Creator

Компонент	Ім'я екземпляру	Апаратні ресурси
12-ти розрядний ЦАП	VDAC_1	1 Continuous time DAC
DMA	DMA_1	1 DMA channel
Clock	Clock_1	1 Peripheral clock divider
Analog Pin	VDAC_Out_P9_6	1 Analog output pin

#### *Компонент DMA.*

Означення:

**DMA** – Direct Memory Access (прямий доступ до пам'яті). Загальний термін, який визначає принцип передачі даних без використання процесора.

**DMA controller** – контролер DMA. Блок HW в мікроконтролері PSoC 6.

**DMA channel** – окремий одиночний канал блоку DMA.

**DW – Data Wire.** Інша назва блоку HW DMA. Використовується декілька раз в API як посилання на певний номер блоку/каналу DMA.

**DMA driver** – це частина бібліотеки периферійних драйверів (PDL) для управління каналом DMA.

**DMA Component** – відноситься до DMA\_PDL компоненти середовища розробки PSoC Creator, яка управляє каналом DMA з використанням драйвера DMA.

**Descriptor** – дескриптор. Встановлює параметри передачі для передачі по каналу DMA. Дескриптор ініціалізується в SRAM і на нього виконується посилання з допомогою вказівника в каналі DMA. Декілька дескрипторів можуть бути зв'язані між собою.

Компонент DMA передає дані в пам'ять, регістри і з пам'яті та регістрів. Ці передачі відбуваються незалежно від процесора. Передача в DMA може бути налаштована довжиною байт, півслово (2 байти) або слово (4 байти). DMA починає кожну транзакцію через зовнішній тригер, який може надходити з каналу DMA (включаючи його самого), іншого каналу DMA, периферійного пристрою або центрального процесора.

*Основні властивості компоненти DMA:*

- Пристрій підтримує до двох апаратних блоків DMA.
- Кожний блок DMA підтримує до 16-ти каналів DMA.
- Підтримує дескриптори каналів в SRAM.
- Налаштовувані адреси джерела та приймача.
- Налаштовувані режими передачі: одиночна передача, 1D передача (з використанням циклу по X) та 2D передача (з використанням циклів по X та Y).
- Налаштовувана операція по входу тригера:
  - передача одного елементу даних на тригер;
  - один X (внутрішній) цикл передачі на тригер;
  - весь дескриптор на тригер;
  - весь канал дескрипторів на тригер.
- Налаштовуваний вихід тригера.
- Нелаштовувана генерація переривань.

*Основні використання DMA.*

Компонент каналу DMA може використовуватися в проектах, де потрібно передавати дані без участі процесора на основі апаратного сигналу запуску від іншого компонента.

Поширеним використанням є передача даних із пам'яті на периферійний пристрій, наприклад UART. DMA може бути викликане не повним сигналом UART FIFO. DMA буде завантажувати дані в UART до заповнення FIFO.

DMA може також використовуватися для зчитування даних з UART та зберігання їх у пам'яті. Наприклад, DMA може бути викликаний не пустим сигналом FIFO. Та DMA буде передавати дані до тих пір, поки FIFO не стане пустим.

Інше поширене використання – передача даних з АЦП в пам'ять. Сигнал закінчення перетворення АЦП може використовуватися для запуску DMA для передачі даних з АЦП в пам'ять.

DMA може також використовуватися для переміщення блоків пам'яті з однієї області пам'яті в іншу (RAM в RAM, FLASH в RAM). DMA не може записувати дані в FLASH пам'ять.

Кожний канал DMA може бути запущений апаратним сигналом або записом в регістр вбудованого програмного забезпечення, або обома.

Кожний канал DMA може бути пов'язаний з дескриптором. Користувач може створити кілька дескрипторів і прив'язати їх один до одного. Користувач також може переналаштувати дескриптор, поки він не є активним дескриптором у DMA.

Архітектура DMA контролера зображена на рис. 9.2.

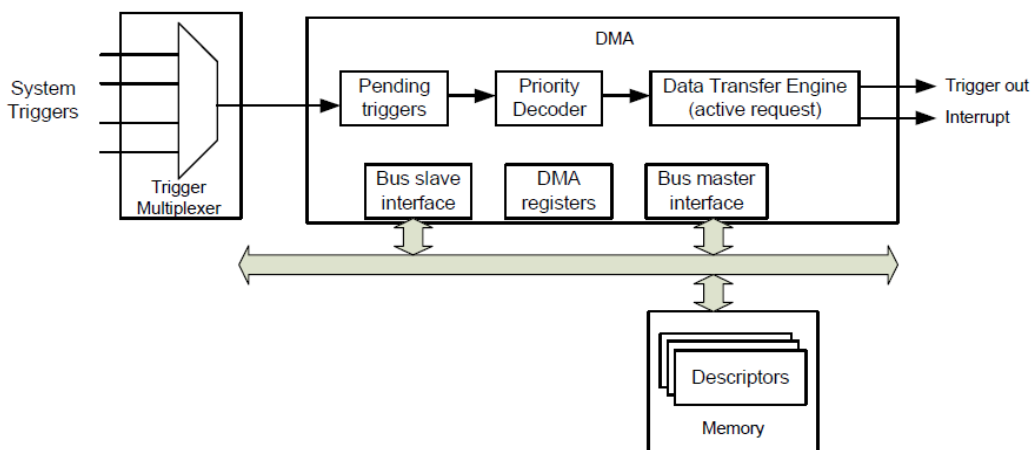


Рис. 9.2. Архітектура контролера DMA

Контролер DMA підтримує декілька незалежних передач даних, якими керує канал. Кожен канал підключається до певного системного тригера через мультиплексор тригера, який знаходиться поза контролером DMA.

Пріоритет каналу: каналу призначається пріоритет (CHi\_CTL.PRIO) від 0 до 3, де 0 – самий високий пріоритет, а 3 – самий низький. Канали з однаковим пріоритетом утворюють групу пріоритетів. Пріоритетне декодування визначає очікуючий канал – з найвищим пріоритетом, який визначається наступним чином:

- група з найвищим пріоритетом з очікуючими каналами ідентифікується першою;

- в рамках цієї пріоритетної групи використовується круговий арбітраж. Циклічний арбітраж всередині пріоритетної групи дає самий високий пріоритет нижнім індексам каналу в пріоритетній групі.

Стан каналу: в будь-який момент часу один канал активно виконує передачу даних. Цей канал називається активним каналом. Канал може знаходитися в одному з 4-х станів:

*Disabled.* Канал відключається установкою CHi\_CTL.ENABLED в логічний '0'. В цьому стані тригер каналу ігнорується.

*Blocked.* Канал включений і очікує тригер для ініціювання передачі даних.

*Pending.* Канал включений і атримав активний тригер. В цьому стані канал готовий почати передачу даних, але очікує, що це буде назначено.

*Active.* Канал включений, отримав активний тригер і був запланований. Він активно здійснює передачу даних. Якщо очікується декілька каналів, запланований канал з найвищим пріоритетом.

Передача даних, пов'язана з тригером, складається з однією або декількох "атомарних передач" або "одиначних передач". Один тригер може бути налаштований для передачі декількох "одиначних передач".

Канал можна помітити як пріоритетний (CHi\_CTL.PREEMPTABLE). Якщо пріоритет можливий і є канал з більш високим пріоритетом, то цей канал з більш високим пріоритетом може розвантажувати поточний канал між одиначними передачами.

Канал має два атрибути управління доступом, які використовуються блоками захисту загальної пам'яті (SMPU) та блоками периферійного захисту

(PPU) для контролю доступу. Ці поля, як правило, успадковуються від головного, який змінив регістр керування каналу.

- Атрибут Privileged Mode - привілейований режим (CHi\_CTL.P) може бути встановлений як привілейований або користувацький.

- Атрибут Non-secure - незахищений (CHi\_CTL.NS) може бути встановлений як безпечний або незахищений.

Дескриптор, пов'язаний з кожним каналом, описує передачу даних. Дескриптор зберігається в пам'яті, і CHi\_CURR\_PTR представляє адресу дескриптора, пов'язаного з каналом 'i', а Chi\_IDX представляє поточні індекси X та Y в дескриптор.

Стан дескриптора каналу кодується як частина стану регістра каналу. Наступні регістри представляють стан дескриптора каналу:

- CH\_CTL. Цей регістр забезпечує загальне управління каналом інформації.

- CH\_CURR\_PTR. Цей регістр представляє адресу комірки пам'яті, в якій знаходиться поточний дескриптор. Користувацька прошивка повинна ініціалізувати цей регістр. Якщо дескриптори зв'язані між собою, апаратне забезпечення DMA автоматично встановлює цей регістр для наступного вказівника дескриптора.

- CH\_IDX. Цей регістр представляє поточні індекси X та Y каналу в поточний дескриптор. Користувацька прошивка повинна ініціалізувати цей регістр. Апаратне забезпечення DMA встановлює індекси X та Y рівними 0 при переході від поточного дескриптора до наступного дескриптора в списку дескрипторів.

Потрібно пам'ятати, що стан каналу зберігається в режимі глибокого сну.

### *Вибір тригера.*

Сигнали тригера можуть генеруватися з різних частин мікросхеми. Блок мультиплексора тригера допомагає направити ці тригерні сигнали до місця призначення. DMA є одним з таких місць призначення. Блок мультиплексора тригера знаходиться поза блоком DMA.

### *Тригери, що очікують на розгляд.*

Тригери, що очікують, відслідковують активовані тригери, локально зберігаючи їх у бітах очікування. Це важливо, оскільки декілька каналів тригерів можуть бути активовані одночасно, тоді як механізм передачі даних може обслуговувати тільки один канал одночасно. Цей компонент дозволяє використовувати тригери, чутливі до рівня (високий/'1'), так і чутливі до імпульсу (два високих/'1' циклу clk\_slow).

- Чутливі до рівня тригери пов'язані з певним станом, наприклад, FIFO є заповненим. Ці тригери залишаються активними до тих пір, доки стан зберігається. Не потрібно відстежувати чутливі до рівня тригери, що очікують, в контролері DMA, оскільки тригери підтримуються поза контролером.

- Чутливі до імпульсу тригери пов'язані з певною подією, наприклад, зразок АЦП став доступним. Необхідно відстежувати ці тригери в контролері DMA, оскільки імпульс тригера може зникнути до того, як він буде оброблений

механізмом передачі даних. Імпульсні тригери повинні бути високими/'1' протягом двох циклів `clk_slow`.

Пріоритетний декодер визначає очікуючий канал з найвищим пріоритетом.

Data transfer engine – механізм передачі даних – відповідає за передачу даних з початкового місцезнаходження в кінцеве. В режимі очікування механізм передачі даних готовий прийняти активований канал з найвищим пріоритетом. Він також відповідає за читання дескриптора каналу з пам'яті.

Master I/F – це майстер шини АНВ-Lite, який дозволяє контролеру DMA ініціювати передачу даних АНВ-Lite в місце джерела і призначення, а також зчитати дескриптор з пам'яті.

Slave I/F – це ведений пристрій шини АНВ-Lite, який дозволяє головному процесору отримувати доступ до регістрів управління/стану контролера DMA.

*Вихідні тригери.*

Кожний канал має вихідний тригер. Цей тригер є високим для двох повільних тактових імпульсів. Тригер генерує по закінченні передачі даних. На системному рівні ці вихідні тригери можуть бути підключені до компоненти мультиплексора тригера. Це з'єднання дозволяє підключити вихідний тригер контролера DMA до вхідного тригера контролера DMA. Отже, закінчення передачі в одному каналі може активувати інший канал або навіть повторно активувати той самий.

*Підключення входів/виходів.*

Розглянемо опис вхідних та вихідних з'єднань для компонента DMA. Зірочка (\*) у наведеному нижче списку вказує, що з'єднання може не відображатися на символі компонента для умов, перелічених в описі цього вводу/виводу.

Terminal Name	I/O Type	Description
tr_in*	Digital Input	Відображається лише в тому випадку, якщо для параметра "Trigger input" встановлено значення True. Параметр вхідного запуску встановлює вхідний сигнал запуску для компонента DMA.
tr_out*	Digital Output	Відображається лише в тому випадку, якщо для параметра "Trigger output" встановлено значення True. Параметр вихідного запуску встановлює вихідний сигнал запуску для компонента DMA.
interrupt	Digital Output	Вихідний термінал, що дозволяє підключити компонент переривання.

У діалоговому вікні компонента DMA є наступні вкладки з різними параметрами.

*Вкладка Channel.*

Ця вкладка містить налаштування каналу DMA та кількості дескрипторів (рис. 9.2).

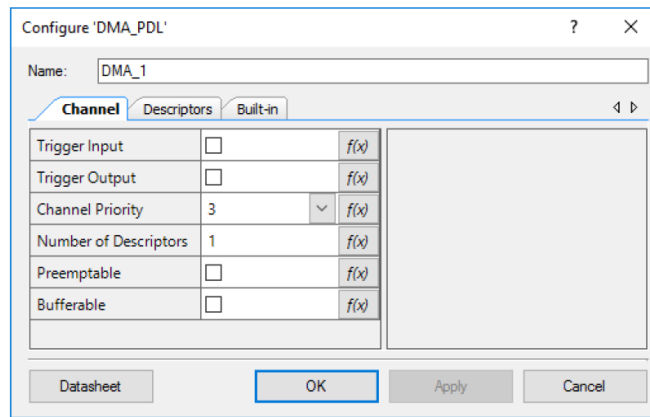


Рис. 9.2. Вигляд вкладки Channel компоненти DMA

До цієї вкладки входять наступні параметри:

- Trigger input – Параметр вхідного тригера, який встановлює вхідний сигнал тригера для компонента DMA.
- Trigger output – Параметр вихідного тригера, який встановлює вихідний сигнал тригера для компонента DMA.
- Channel Priority – Задає пріоритет каналу в блоці DMA.
- Number of Descriptors – Задає кількість дескрипторів. Виходячи з числа, встановленого в цьому параметрі, у налаштовувачі створюється стільки ж екземплярів дескрипторів.
- Preemptable – Якщо цей параметр включений, і є канал з більш високим пріоритетом, який перебуває в очікуванні, то цей канал може вивантажувати поточний канал між одиничними передачами.
- Bufferable – Цей параметр вказує, чи може закінчитися транзакція запису, не дочекавшись, поки адресат прийме дані транзакції запису.

Вкладка, зображена на рис. 9.3, містить параметри конфігурації дескрипторів. Для кожного з дескрипторів створюється однаковий набір параметрів.

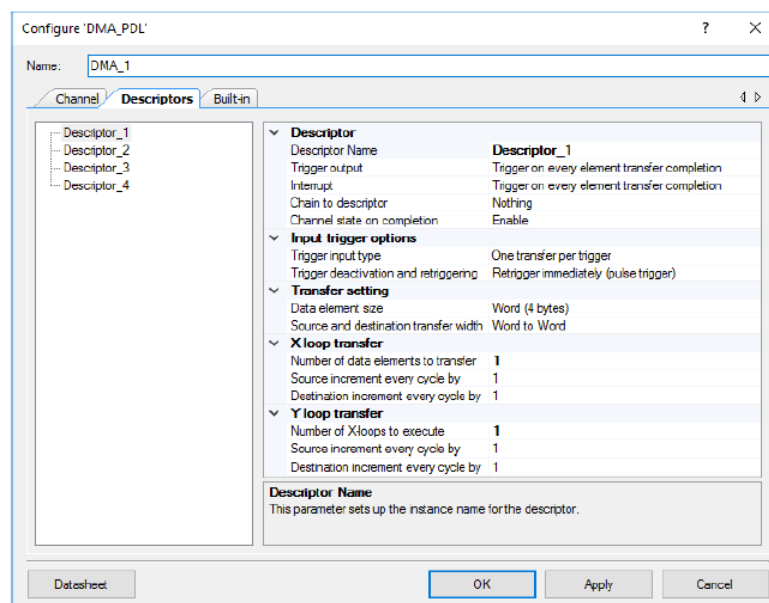


Рис. 9.3. Вигляд вкладки з параметрами конфігурації дескрипторів

Вкладка Descriptors включає функцію налаштування copy/paste дескриптора. Потрібно клацнути правою кнопкою мишки та вибрати Copy (у контекстному меню) будь-якого дескриптора (у списку дескрипторів зліва). Це копіює його налаштування у внутрішній буфер. Потім потрібно клацнути правою кнопкою мишки та вибрати Paste для будь-якого дескриптора, щоб замінити його налаштування (усі, окрім "Descriptor Name" та "Chain to descriptor"), на попередньо скопійовані з внутрішнього буфера. Також це можна виконати з допомогою клавіш Ctrl+C і Ctrl+V.

#### *Опис параметрів дескрипторів:*

- Descriptor Name. Цей параметр встановлює ім'я екземпляра для дескриптора. За замовчуванням використовується значення "Descriptor\_N", де N – порядковий номер дескриптора.

- Trigger output. Цей параметр вибирає, яка подія викличе запуск виходу DMA:

- запуск при кожному закінченні передачі елемента (за замовчуванням);
- запуск при кожному закінченні передачі X циклу;
- запуск при завершенні дескриптора;
- запуск при завершенні всього каналу дескриптора.

- Interrupt. Цей параметр вибирає, яка подія викличе переривання DMA:

- запуск при кожному закінченні передачі елемента (за замовчуванням);
- запуск при кожному закінченні передачі X циклу;
- запуск при завершенні дескриптора;
- запуск при завершенні всього каналу дескриптора.

- Channel state on completion (стан каналу по завершенні). Цей параметр визначає стан (enable/disable) включення/виключення каналу, коли дескриптор завершений. У випадку "disable" канал повинен бути повторно включений (Cy\_DMA\_Channel\_Enable) після закінчення поточного дескриптора для подальшого функціонування. За замовчуванням цей параметр "enable".

- Chain to descriptor (коло до дескриптора). Цей параметр дозволяє налаштувати те, що буде виконуватися після поточного дескриптора. Список, що відображається, показує список всіх дескрипторів каналу. За замовчуванням використовується значення "Nothing", яке є еквівалентне вказівнику "NULL" в C- коді.

#### *Параметри запуску входу.*

- Trigger input type (тип запуску входу). Тип запуску входу задає характеристику кожного запуску:

- одна передача за запуск (за замовчуванням);
- передача одного циклу X за запуск;
- ціла передача дескриптора за запуск;
- все коло дескрипторів за дескриптор.

#### *Деактивація тригера і повторний запуск.*

Цей параметр встановлює параметри деактивації тригера для дескриптора:

- негайний повторний запуск (імпульсний тригер) (за замовчуванням);
- повторний запуск після 4 циклів Clk\_Slow;



- повторний запуск після 16 циклів Clk\_Slow;
- очікування активації тригера.

*Налаштування параметрів компонентів:*

На рис. 9.2, ..., рис. 9.5 зображено налаштування параметрів для компонентів, що використовуються в цьому проекті.

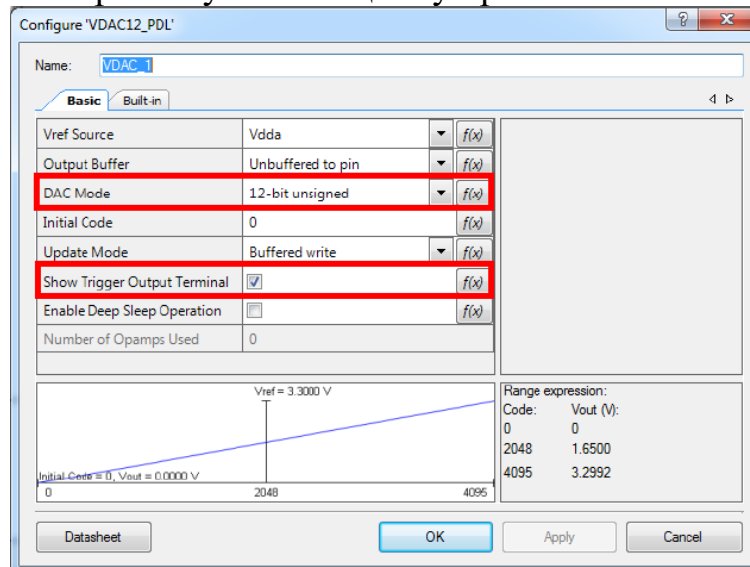


Рис. 9.2. Налаштування параметрів компонента VDAS

Значення синусоїди в таблиці DMA є беззнакові. Тому режим DAC для VDAS встановлюється беззнаковий. Також потрібно включити тригерний вихідний термінал, щоб він міг бути підключений до DMA.

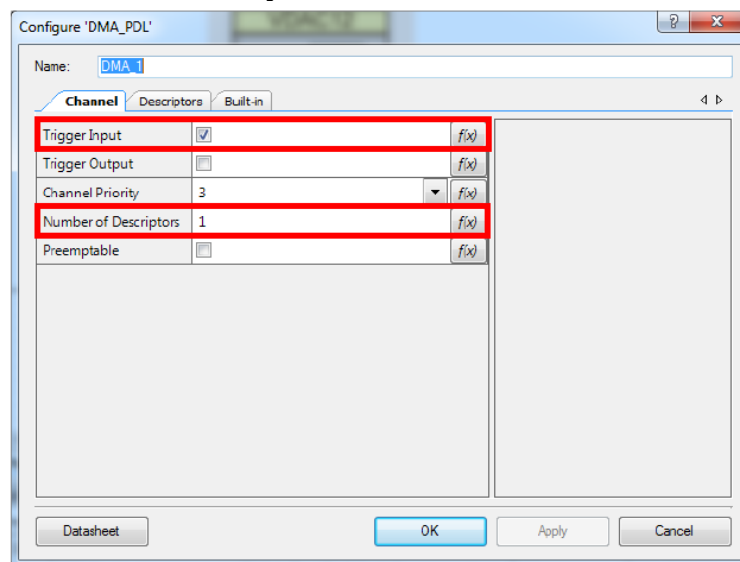


Рис. 9.3. Налаштування параметрів каналу компоненту DMA

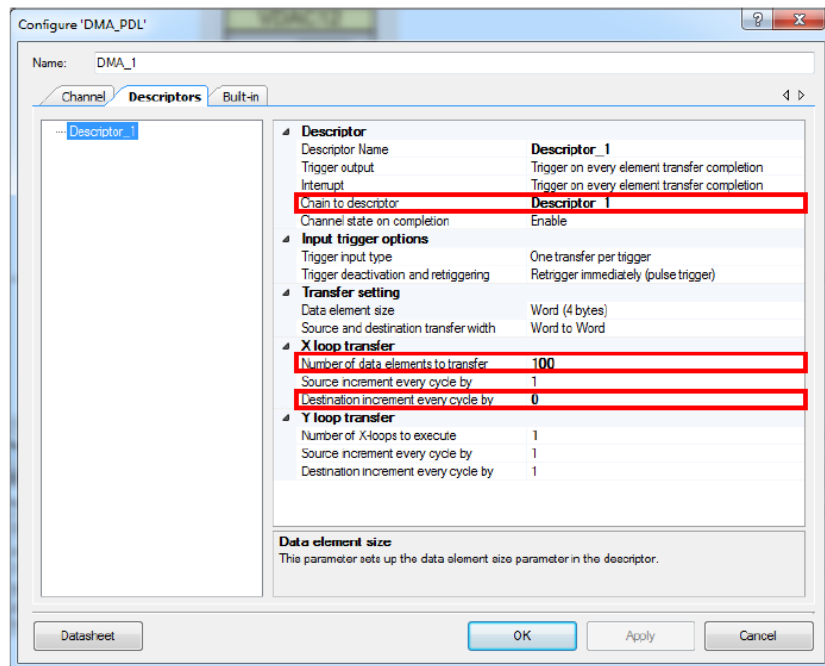


Рис. 9.4. Налаштування параметрів дескрипторів компонента DMA

Виконати прив'язку Descriptor\_1 до себе (Chain to descriptor – Descriptor\_1), щоб синусоїдальний сигнал був неперервним. Кількість елементів даних для передачі потрібно встановити на 100, щоб ця кількість співпадала з розмірністю таблиці синуса. Значення адреси (адреса регістру буферезованих значень VDAC) не буде змінюватися. Тому потрібно “Destination increment every cycle by” встановити на 0.

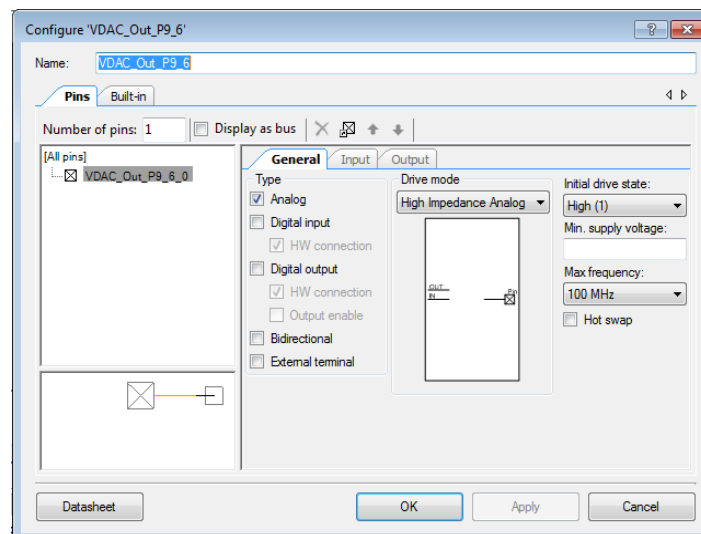


Рис. 9.5. Налаштування параметрів компонента GPIO

### Програмна реалізація проекту.

```
#include "project.h"
#include "ctdac/cy_ctdac.h"
#include "dma/cy_dma.h"
```

```
/* This project assumes a configuration tool will automatically configure system */
```

```
/* Для демонстрація того, як драйвери PDL використовуються для ручної настройки VDAC, потрібно встановити для PDL_CONFIGURATION в #define значення 1 */
```

```

#define PDL_CONFIGURATION    (0u)

/* Число точок таблиці синусоїди */
#define NUM_POINTS           (100u)

/* Таблиця значень синусоїди в без знаковому форматі. */
uint32_t sineWaveLUT[] = {0x7FF, 0x880, 0x900, 0x97F, 0x9FC, 0xA78, 0xAF1, 0xB67, 0xBD9, 0xC48,
                          0xCB2, 0xD18, 0xD79, 0xDD4, 0xE29, 0xE77, 0xEC0, 0xF01, 0xF3C, 0xF6F,
                          0xF9A, 0xFBE, 0xFDA, 0xFEE, 0xFFA, 0xFFF, 0xFFA, 0xFEE, 0xFDA, 0xFBE,
                          0xF9A, 0xF6F, 0xF3C, 0xF01, 0xEC0, 0xE77, 0xE29, 0xDD4, 0xD79, 0xD18,
                          0xCB2, 0xC48, 0xBD9, 0xB67, 0xAF1, 0xA78, 0x9FC, 0x97F, 0x900, 0x880,
                          0x7FF, 0x77E, 0x6FE, 0x67F, 0x602, 0x586, 0x50D, 0x497, 0x425, 0x3B6,
                          0x34C, 0x2E6, 0x285, 0x22A, 0x1D5, 0x187, 0x13E, 0x0FD, 0x0C2, 0x08F,
                          0x064, 0x040, 0x024, 0x010, 0x004, 0x000, 0x004, 0x010, 0x024, 0x040,
                          0x064, 0x08F, 0x0C2, 0x0FD, 0x13E, 0x187, 0x1D5, 0x22A, 0x285, 0x2E6,
                          0x34C, 0x3B6, 0x425, 0x497, 0x50D, 0x586, 0x602, 0x67F, 0x6FE, 0x77E};

/* В головній функції main виконуються наступні дії:
  1. Ініціалізуються компоненти VDAC та DMA
  2. DMA постійно оновлює значення VDAC з таблиці синусів */

int main(void)
{
    /* Дозвіл глобальних переривань. */
    __enable_irq();

#ifdef PDL_CONFIGURATION
    /* Configure VDAC for VDDA reference and unbuffered output.
     * FastInit will configure the VDAC for unsigned mode and buffered updates
     * with a 500 kHz clock by default. */
    Cy_CTDAC_FastInit(CTDAC0, &Cy_CTDAC_Fast_VddaRef_UnbufferedOut);

    /* Configure Trigger Mux to route VDAC trigger output to DMA input trigger
     */
    (void)Cy_TrigMux_Connect(TRIG14_IN_PASS_TR_CTDAC_EMPTY,
        TRIG14_OUT_TR_GROUP0_INPUT50, CY_TR_MUX_TR_INV_DISABLE,
        TRIGGER_TYPE_PASS_TR_CTDAC_EMPTY);
    (void)Cy_TrigMux_Connect(TRIG1_IN_TR_GROUP14_OUTPUT7,
        TRIG1_OUT_CPUSW_DW1_TR_IN0, CY_TR_MUX_TR_INV_DISABLE,
        TRIGGER_TYPE_TR_GROUP_OUTPUT__LEVEL);

    /* Enable CTDAC hardware blocks */
    Cy_CTDAC_Enable(CTDAC0);

    DMA_1_Start(sineWaveLUT, &CTDAC0->CTDAC_VAL_NXT);
#else
    VDAC_1_Start();
    DMA_1_Start(sineWaveLUT, (uint32_t *)&(CTDAC0->CTDAC_VAL_NXT));
#endif

    /* Жодні операції з процесором не потрібні, так як в проекті
    використовується DMA для всіх передач пам'яті. */
    for(;;)
    {
    }
}

/* [] END OF FILE */

```

### **Опис літератури:**

1. CE220924 - PSoC 6 MCU VDAC Sine Wave Generator Using DMA  
 CE23727. Електронний ресурс. Режим доступу: <https://www.cypress.com/file/419746/download>.

### ***Завдання.***

1. Реалізувати з допомогою схеми, приведеної на рис. 9.1 сигнал, зображений на рис. 9.7.