

Лабораторна робота № 7

Вивчення UART мікроконтролерів PSoC 6 фірми Cypress

Мета роботи: Вивчити функціонування USB UART з'єднання для забезпечення передачі даних від PC до PSoC 6 та в зворотньому напрямку.

Теоретичні відомості

Розглянемо проект, в якому використовується процесорне ядро Cortex-M4 (CM4) мікроконтролера PSoC 6 для передачі даних по UART та управління світлодіодами. При перезавантаженні пристрою процесор Cortex-M0+ (CM0+) включає процесор CM4. ЦП CM4 використовує компоненту UART для передачі повідомлення на термінал комп'ютера та при натисканні клавіші "Enter", світлодіод на стенді починає блимати.

На рис. 1 зображена схема проекту PSoC Creator 4.2 для цього проекту.

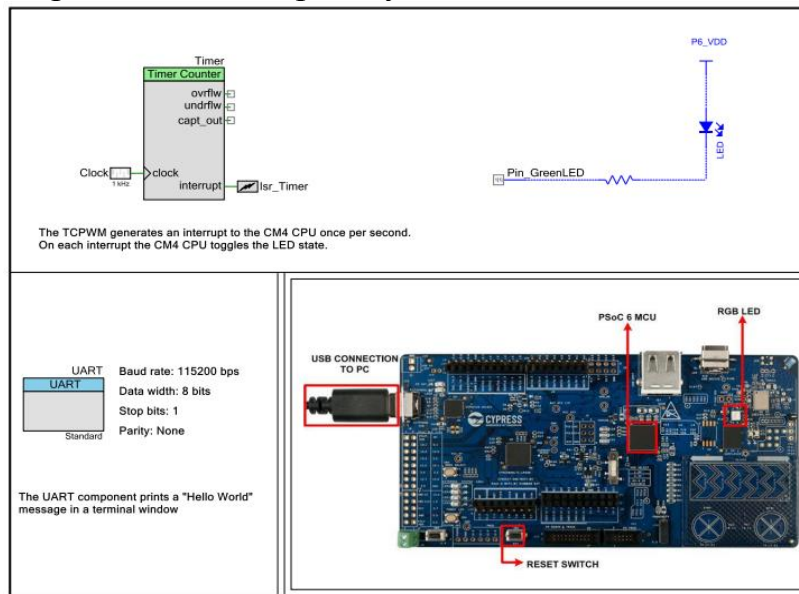


Рис. 1. Схема проекту

На рис. 2 показані параметри конфігурації для компоненти таймера. Компонент таймера/лічильника (TCPWM) налаштований як таймер і генерує переривання кожну секунду. Вхід синхронізації компоненти TCPWM під'єднаний до джерела тактових імпульсів частотою 1 кГц.

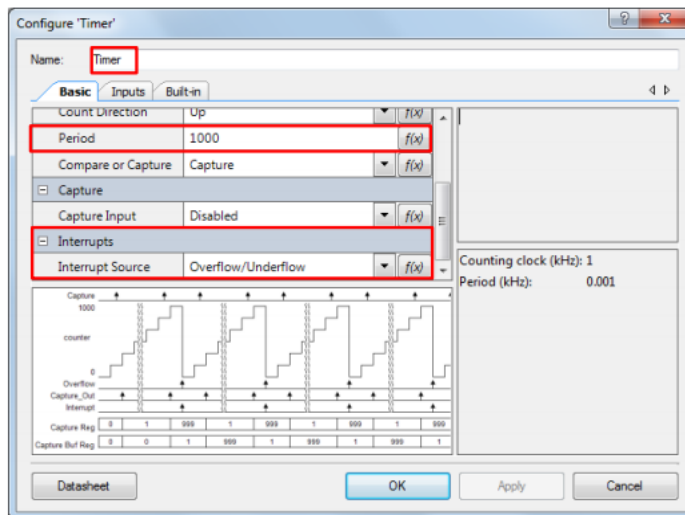


Рис. 2. Налаштування компонента таймера (TCPWM)

Конфігурація UART компонента. Двічі клацнувши на компоненті UART, відкриється вікно конфігурації (рис. 2). Змінимо в ньому ім'я екземпляра компонента на UART. Для інших налаштувань використовуються значення за замовчуванням.

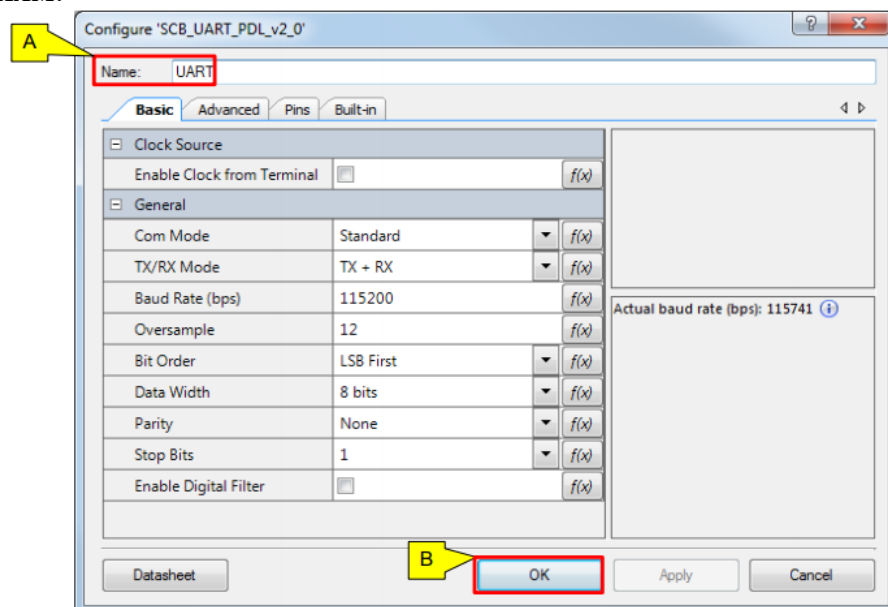


Рис. 2. Вікно налаштувань UART компонента на основі SCB

Налаштування компонента переривання SysInt для співставлення переривання TCPWM з процесором CM4. Двічі клацнемо на ньому і в вікні, що відкриється, поміняємо ім'я на Isr_Timer (рис. 4).

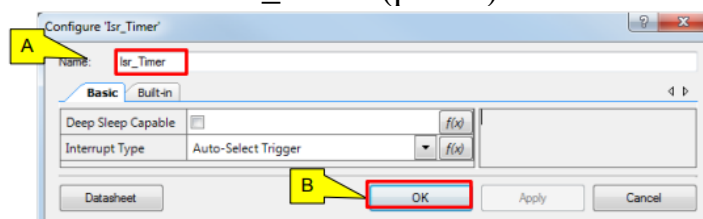


Рис. 3. Налаштування компонента SysInt_PDL

Під'єднаємо вихід компоненти переривання Isr_Timer до виходу переривання компоненти Timer. Це маршрутизує переривання TCPWM на ЦП CM4. Вибір ЦП CM4 для цього переривання встановимо в системі налаштування переривання пізніше.

Встановлення фізичних виводів для кожної компоненти. Вибір використовуваного виводу визначається дизайном плати стенду. На рис. 4 зображено результат цього кроку.



Рис. 4. Призначення виводів в схемі проекту

Налаштування системних імпульсів синхронізації. Проект використовує значення за замовчуванням для високочастотних системних налаштувань імпульсів синхронізації.

В багатьох випадках буває необхідно змінити частоту системних імпульсів синхронізації. Для цього потрібно:

- У вкладці Workspace Explorer двічі клацнути на елементі Clocks в розділі Design Wide Resources. З'явиться список імпульсів синхронізації.

- Натиснути Edit Clock. Відкриється діалогове вікно Configure System Clocks. Тут можна побачити дерево тактових імпульсів і змінити їх частоту при потребі. Потрібно зауважити, що є вкладки для різних типів тактових імпульсів, таких як тактові генератори, FLL/PLL, високочастотні генератори тактових імпульсів, інші генератори тактових імпульсів.

- Натиснути на вкладку FLL/PLL. За замовчуванням PSoC Creator 4.2 включає FLL і встановлює частоту 100 МГц.

- Натиснути на вкладку високочастотні генератори тактових імпульсів. Можна встановити тактову частоту процесора CM4, встановивши дільник в Clk_Fast. За замовчуванням дільник встановлений на 1.

- Можна встановити тактову частоту процесора CM0+, встановивши дільник в Clk_Slow. За замовчуванням дільник встановлений в 1 (рис. 5).

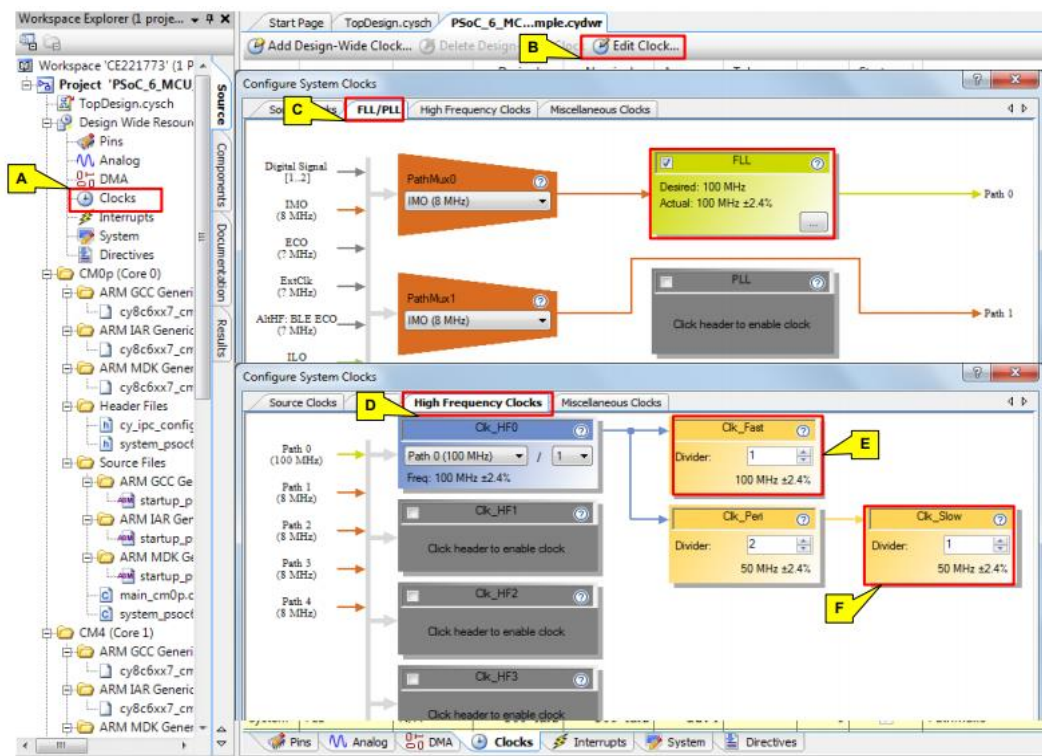
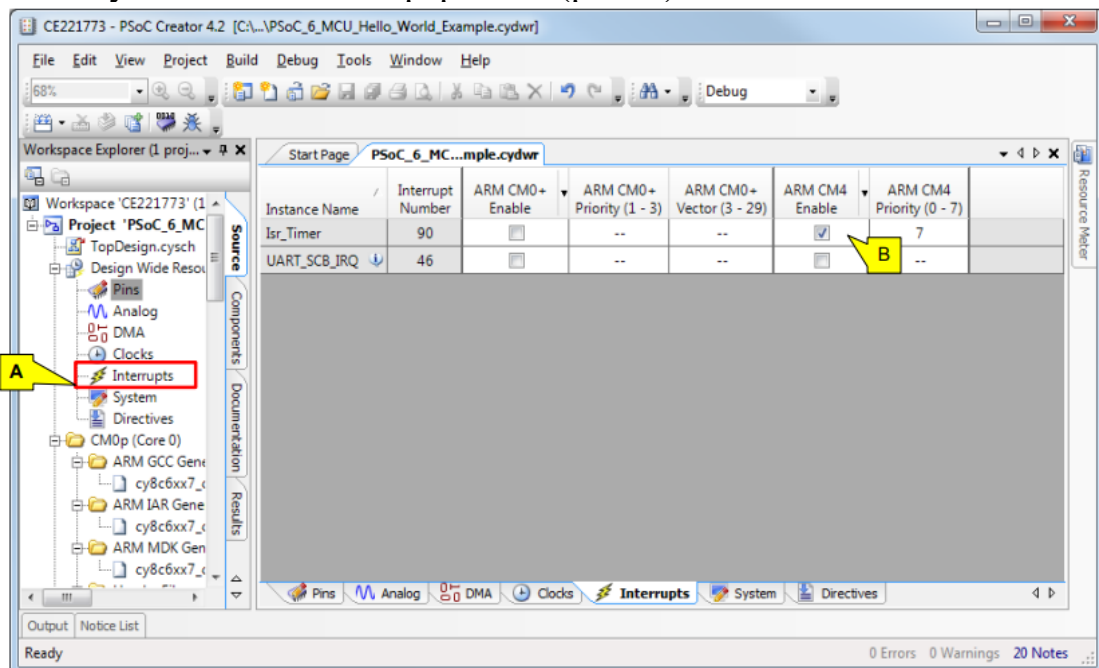


Рис. 5. Конфігурація тактових імпульсів.

Налаштування системи переривань (рис. 6).



Керуючі символи ANSI (англ. ANSI escape code) - символи, що вбудовуються в текст, для управління форматом, кольором та іншими опціями виводу в текстовому терміналі. Майже всі емулятори терміналів, призначені для відображення текстового виводу з віддаленого комп'ютера і для відображення текстового виводу локального програмного забезпечення, здатні інтерпретувати принаймні деякі керуючі послідовності ANSI.

Керуючі послідовності починаються з символу ESC (ASCII:27/0x1B/033). Для послідовностей з двох символів другим символом може бути символ ASCII

в діапазоні від 64 до 95 (від @ до _). Проте багато послідовностей включають більше двох символів і починаються з символів ESC і [. Сукупність символів ESC і [називають CSI або Control Sequence Introducer (також Control Sequence Initiator). Останнім символом такої послідовності може бути ASCII-символ в діапазоні від 64 до 126 (від @ до ~).

Структура більшості керуючих послідовностей ANSI виглядає наступним чином:

CSI [символ(-и) режиму] n1; n2... [закрывающий(-и) символ(-и)] буква

Останній байт (буква), змінений символами режиму і замикаючими символами, визначає команду. Числа n1, n2, ... - необов'язкові параметри. Якщо їх значення не вказані, то за замовчуванням зазвичай використовуються 0 або 1, залежно від операції. Якщо не вказані такі числа, то і крапки з комами теж можуть бути пропущені.

Останнім байтом, формально, може бути будь-який символ від 64 до 126 (від 0x40 до 0x7e, ASCII: від @ до ~). Він може бути змінений проміжними байтами, що стоять перед ним в діапазоні від 32 до 47 (від 0x20 до 0x2f).

Деякі керуючі послідовності ANSI.

Код	Назва	Ефект
CSI n A	CUU — Cursor Up	Переміщує курсор на n (за замовчуванням 1) позицій у зазначеному напрямку. Якщо курсор вже знаходиться скраю вікна терміналу, нічого не відбувається.
CSI n B	CUD — Cursor Down	
CSI n C	CUF — Cursor Forward	
CSI n D	CUB — Cursor Back	
CSI n E	CNL — Cursor Next Line	Переміщує курсор в початок n-го (за замовчуванням 1-го) рядку знизу щодо поточного.
CSI n F	CPL — Cursor Previous Line	Переміщує курсор в початок n-го (за замовчуванням 1-го) рядку зверху щодо поточного.
CSI n G	CHA — Cursor Horizontal Absolute	Переміщує курсор в стовпець n.
CSI n ; m H	CUP — Cursor Position	Переміщує курсор в рядок n, стовпець m. Значення n і m за замовчуванням 1 (лівий верхній кут), якщо зазначений тільки один аргумент без крапки з комою, вважається що це n. Наприклад, послідовність CSI; 5H аналогічна CSI 1; 5H, CSI 17; H аналогічна CSI 17H і CSI 17; 1H.
CSI n J	ED — Erase Data	Очищає частину вікна терміналу. Якщо n дорівнює нулю (або відсутнє), очищає все від курсора до кінця вікна терміналу. Якщо n дорівнює одиниці, очищає все від курсору до початку вікна терміналу. Якщо n дорівнює двом, очищає все вікно терміналу (і переміщує курсор в лівий верхній кут в MS-DOS ANSI.SYS). У випадку n рівного 2, замість очищення може статися просто створення нових рядків так, щоб попередній вміст терміналу витіснився вгору.
CSI n K	EL — Erase in Line	Видаляє частину рядка. Якщо n дорівнює нулю (або відсутнє), очищає все від курсора до кінця рядка. Якщо n дорівнює одиниці, очищає все від курсора до початку рядка. Якщо n дорівнює двом, очищає весь рядок. Положення курсору не змінюється.
CSI n S	SU — Scroll Up	Прокручує поточну сторінку терміналу вгору на n (за

CSI n T	SD — Scroll Down	замовчуванням 1) рядків. Нові рядки додаються знизу, рядки в верхній частині поточної сторінки затираються.
CSI n ; m f	HVP — Horizontal and Vertical Position	Прокручує поточну сторінку терміналу вниз на n (за замовчуванням 1) рядків. Нові рядки додаються зверху, рядки в нижній частині поточної сторінки видаляються.
CSI n [$;$ k] m	SGR — Select Graphic Rendition	Переміщує курсор в рядок n , стовпець m . Значення за замовчуванням обох параметрів 1. Той же ефект, що і CUP.
CSI 6 n	DSR — Device Status Report	Установка SGR параметрів, що визначають кольори тексту і фону. Після CSI можуть перебувати нуль або більше параметрів, розділених крапками з комами. Якщо параметри не зазначені, CSI m поводить себе як CSI 0 m (reset / normal).
CSI s	SCP — Save Cursor Position	Повідомляє поточний стан курсору в форматі ESC [n ; m R, де n - рядок, m - стовпець.
CSI u	RCP — Restore Cursor Position	Зберігає положення курсору.
CSI ?25l	DECTCEM	Відновлює положення курсору.
CSI ?25h	DECTCEM	Приховує курсор.
		Робить курсор видимим.

Реалізація програмного забезпечення.

```
#include "project.h"

// Визначення макросів
#define LED_ON      (0)
#define LED_OFF    (!LED_ON)

void UartInit(void);
void TimerInit(void);
void Isr_Timer(void);

// Глобальні змінні
bool LEDUpdateFlag = false;

// Основна функція проекту.
// Це основна функція для процесорного ядра CM4. Запускає UART та TCPWM
// периферію. TCPWM налаштований як таймер, який генерує переривання
// кожної секунди. При кожному перериванні змінюється стан світлодіоду.

int main(void)
{
    // Запуск UART периферії
    UartInit();

    // Запуск глобальних переривань
    __enable_irq();

    // \x1b[2J\x1b[;H - ANSI ESC послідовність для очищення екрану
    Cy_SCB_UART_PutString(UART_HW, "\x1b[2J\x1b[;H");
    Cy_SCB_UART_PutString(UART_HW, "***Project - CE221773 - PSoC 6 MCU:"\
        " Hello World! Example*****\r\n\n");
    Cy_SCB_UART_PutString(UART_HW, " --- Lab_Work_7 ---\r\n\n");
}
```



```

    Cy_SCB_UART_PutString(UART_HW, "Press Enter key to start blinking the
LED\r\n\r\n");

// Очікування, поки користувач натисне клавішу Enter
while(Cy_SCB_UART_Get(UART_HW) != '\r');

// Запуск TCPWM периферії. Компонента TCPWM налаштована як таймер
TimerInit();
Cy_SCB_UART_PutString(UART_HW, "Observe the LED blinking on the
kit!!!\r\n");

    for(;;)
    {
        if(LEDUpdateFlag)
        {
            // Очищення прапорця
            LEDUpdateFlag = false;

            // Зміна стану світлодіода LED
            Cy_GPIO_Inv(Pin_GreenLED_0_PORT, Pin_GreenLED_0_NUM);
        }
    }
}

// Функція UartInit. Ця функція ініціалізує периферійні пристрої UART з
// налаштуваннями конфігурації UART, визначеної в структурі UART_config.
// Вона також запускає UART периферію.
void UartInit(void)
{
    /* Конфігурація UART периферії.
    Структура UART_config визначається компонентою UART_PDL на основі
    параметрів, введених в конфігурацію компоненти */
    Cy_SCB_UART_Init(UART_HW, &UART_config, &UART_context);
    /* Запуск UART периферії */
    Cy_SCB_UART_Enable(UART_HW);
}

/* Функція TimerInit. Ця функція конфігурує та запускає TCPWM периферію в
режимі таймера. Вона також конфігурує та відображає переривання TCPWM на
процесорне ядро CM4. */

void TimerInit(void)
{
    /* Конфігурація TCPWM периферії.
    Структура Counter_config визначається на основі введених параметрів
    в конфігурації компоненти */
    Cy_TCPWM_Counter_Init(Timer_HW, Timer_CNT_NUM, &Timer_config);

    /* Запуск ініціалізації лічильника */
    Cy_TCPWM_Counter_Enable(Timer_HW, Timer_CNT_NUM);

    /* Запуск увімкненого лічильника */
    Cy_TCPWM_TriggerStart(Timer_HW, Timer_CNT_MASK);

    /* Configure the ISR for the TCPWM peripheral*/
    Cy_SysInt_Init(&Isr_Timer_cfg, Isr_Timer);

    /* Включення переривання в NVIC */
    NVIC_EnableIRQ((IRQn_Type)Isr_Timer_cfg.intrSrc);
}

/* Функція Isr_Timer. Це функція обробника переривань для переривання TCPWM.
void Isr_Timer(void)
{

```

```

/* Очищення периферійного переривання TCPWM */
Cy_TCPWM_ClearInterrupt(Timer_HW, Timer_CNT_NUM, CY_TCPWM_INT_ON_TC);

/* Очищення очікуваних переривань CM4 NVIC для TCPWM */
NVIC_ClearPendingIRQ(Isr_Timer_cfg.intrSrc);
LEDUpdateFlag = true;
}

/* [] END OF FILE */

```

Для реалізації зв'язку між комп'ютером та PSoC 4 не достатньо лише правильно забезпечити з'єднання UART. Для візуального спостереження пакету даних, які надходять з PSoC на комп'ютер та для формування даних які надсилаються в зворотньому напрямку розробники компанії Cypress рекомендують використовувати програму «Putty». Завантажити програму можна за адресою: <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>.

Розглянемо налаштування, які потрібно здійснити для того, щоб реалізувати двосторонній обмін інформації між комп'ютером та PSoC 6.

Програма Putty не потребує інсталяції. Після її запуску відкриється вікно, зображене на рис. 3.

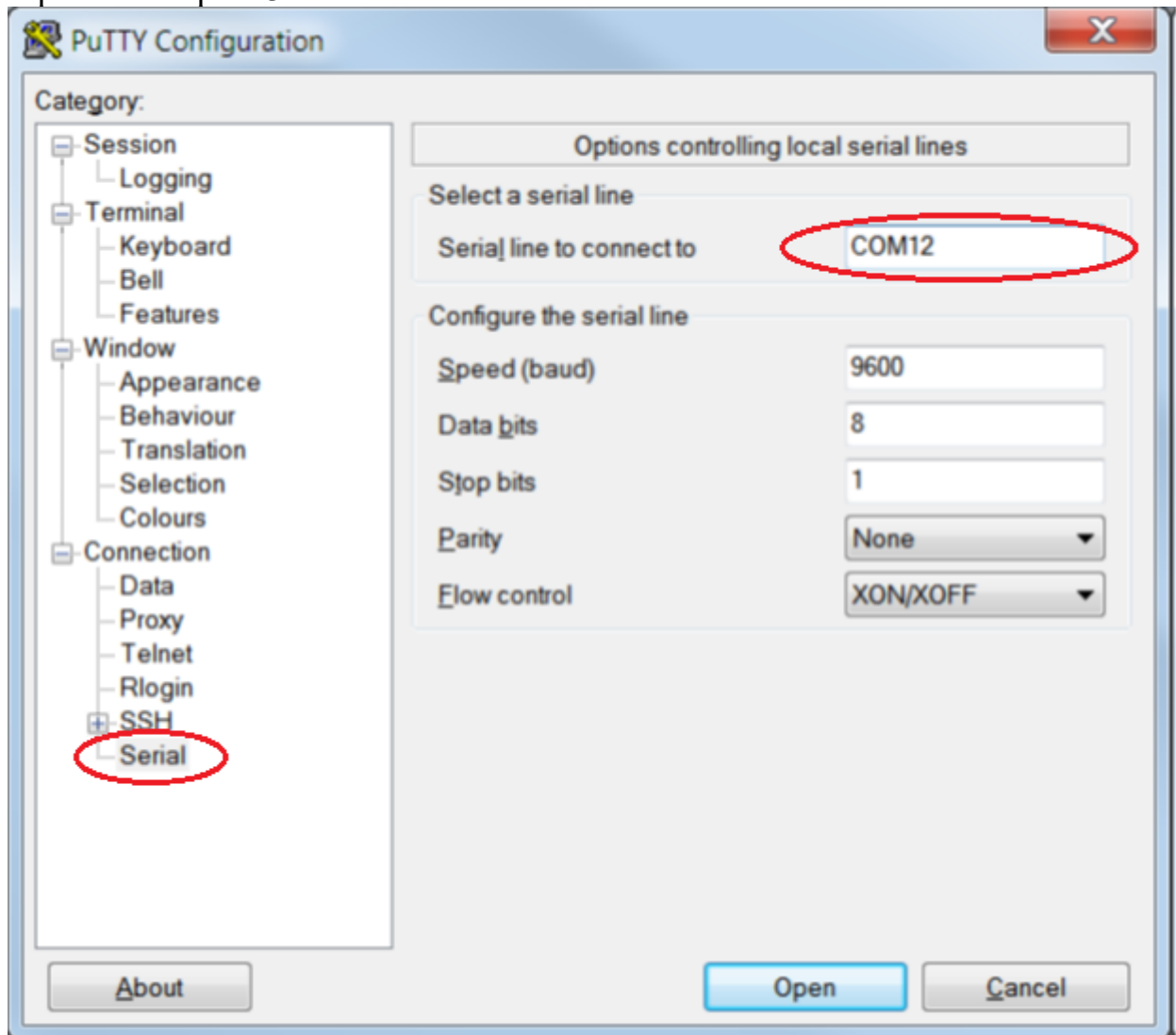


Рис. 3. Вигляд головного вікна програми Putty

В даному вікні в блоці “Category” вибираємо категорію “Serial” як вказано на рисунку, а в блоці “Serial line to connect to” вказуємо відповідний номер віртуального COM- порту, до якого підключений стенд PSoC 6. Якщо номер порту невідомий, то його можна взяти зайшовши в меню "Диспетчер пристроїв". Вказавши відповідне значення COM- порту, до якого приєднано PSoC 6. В пункті “Speed (baud)” вказуємо значення швидкості передачі даних. Це значення повинно співпадати із значенням, яке вказується при налаштуванні модуля UART.

Після внесення необхідних змін переходимо на вкладку “Session” та ставимо прапорць в блоці “Serial”, як це вказано на рис. 4. Після цього у колонках “Serial line” та “Speed” автоматично запишуться вказані раніше значення COM-порту та швидкості передачі даних.

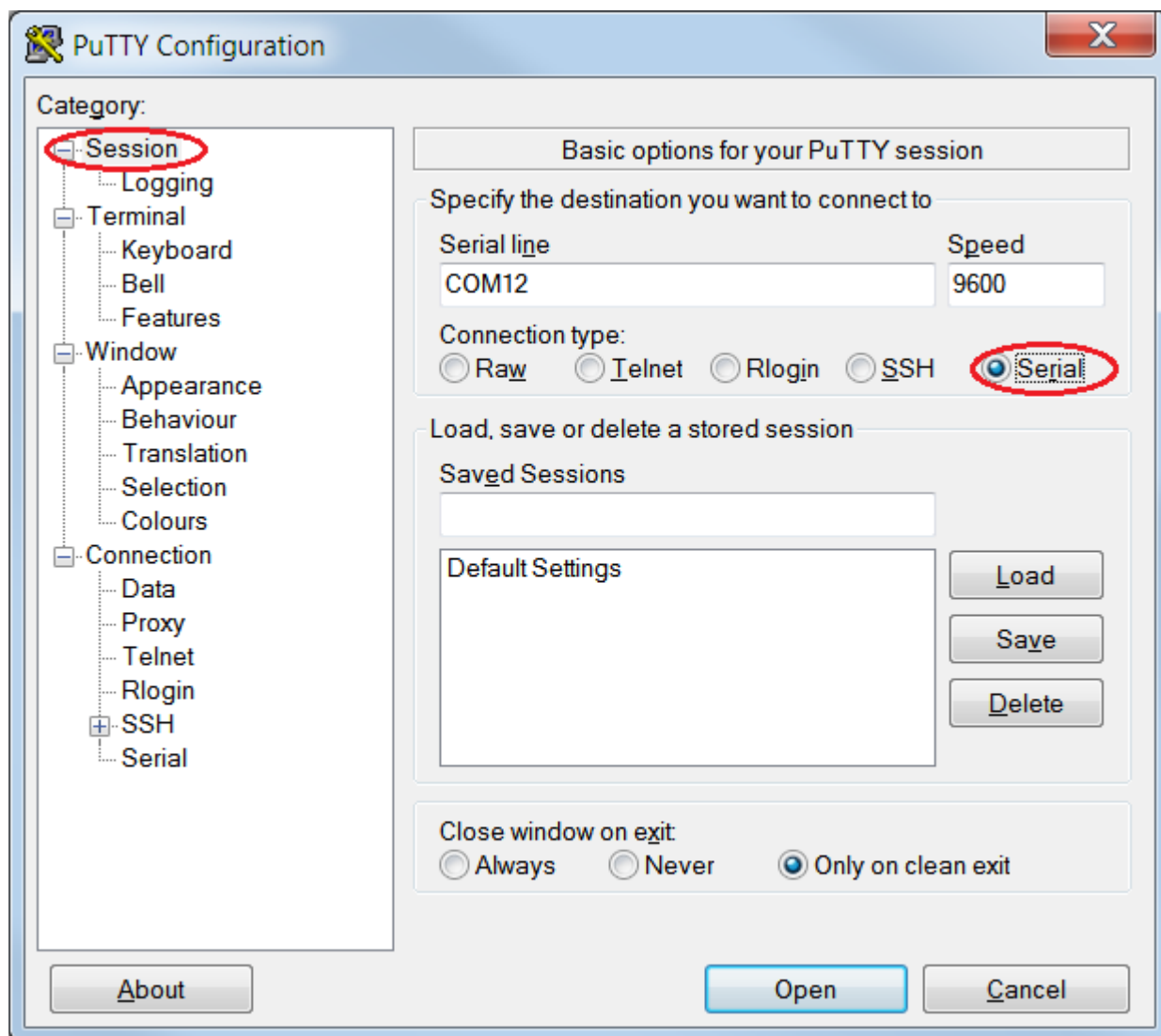


Рис. 4. Налаштування даних вкладки Session програми Putty

Після завершення налаштувань у категорії “Terminal” в підкатегорії “Line discipline options” налаштування встановлюються згідно рис. 5.

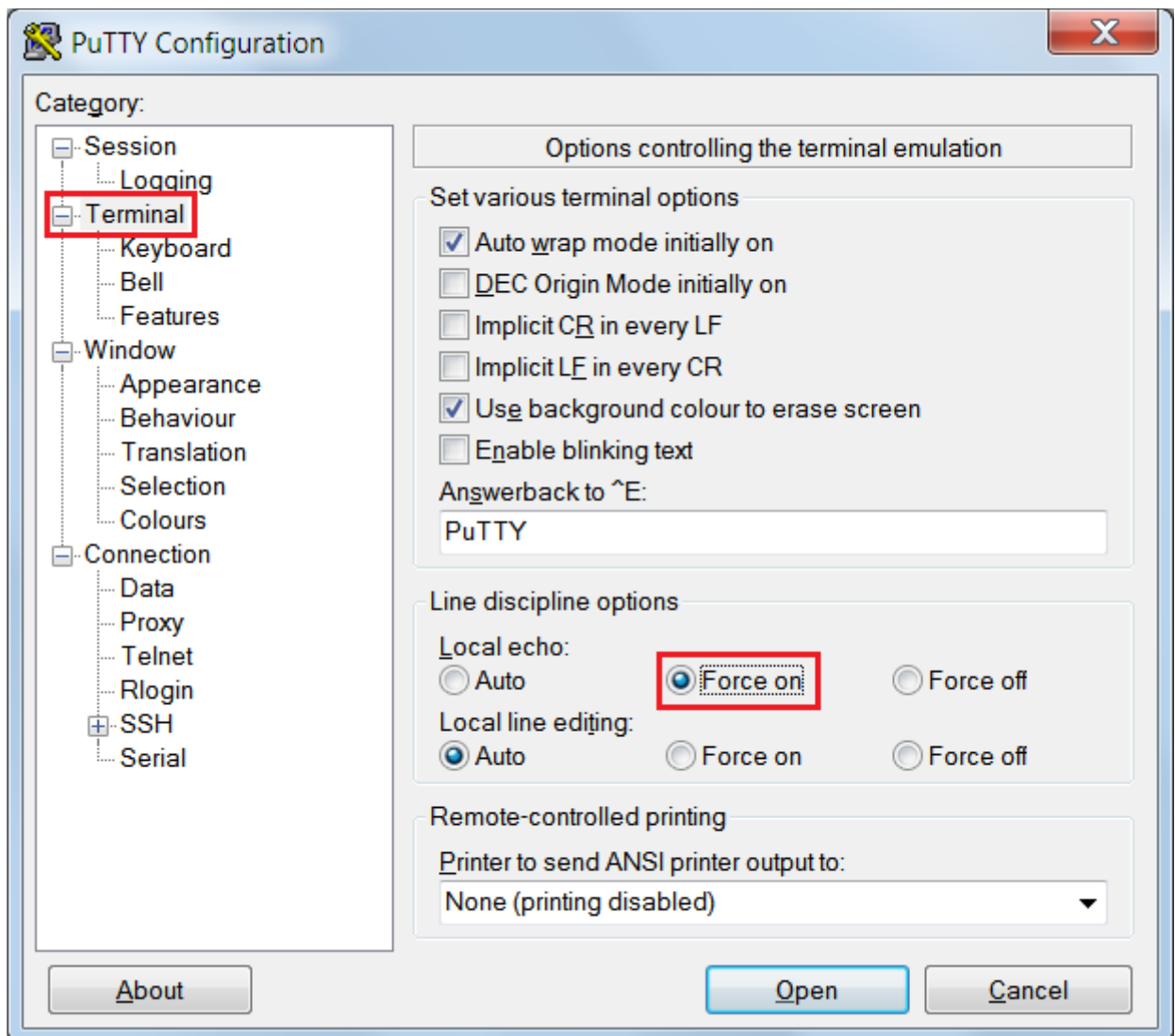


Рис. 5. Налаштування вкладки Terminal програми Putty

Після виконання всіх налаштувань натискаємо на кнопку “Open”, після чого появляється вікно з командною стрічкою, яке відображає дані, отримані з мікроконтролера PSoC 6, та дозволяє надсилати дані в зворотньому напрямку.

Опис літератури:

CE220169 – PSoC 6 MCU: Hello World Example. Електронний ресурс. Режим доступу: <https://www.cypress.com/documentation/code-examples/ce221773-psoc-6-mcu-hello-world-example>.

Завдання.

1. Реалізувати проект засвічування RGB світлодіодів стенду PSoC 6 BLE PIONER KIT з використанням переривань по таймеру та переривань компоненти UART. Алгоритм засвічування світлодіодів: при включенні живлення стенду на термінал виводиться інструкція про роботу цього проекту та дані про розробника. Жодний з RGB світлодіодів не засвічується. При натисканні на кнопку 'r' клавіатури починає блимати червоний світлодіод з частотою 0.5 Гц. При натисканні на кнопку 'b' клавіатури починає блимати

синій світлодіод з частотою 1.0 Гц. При натисканні на кнопку 'g' клавіатури починає блимати зелений світлодіод з частотою 2.0 Гц. При натисканні на кнопку Enter блимання світлодіодів припиняється.

2. Реалізувати проект засвічування RGB світлодіодів стенду PSoC 6 BLE PIONER KIT з використанням переривань по таймеру, переривань виводу GPIO, до якого під'єднано перемикач SW_2 та переривань компоненти UART. Алгоритм засвічування світлодіодів: при включенні живлення стенду на термінал виводиться інструкція про роботу цього проекту та дані про розробника. Жодний з RGB світлодіодів не засвічується. При натисканні на кнопку 'r' клавіатури починає блимати червоний світлодіод з частотою 0.5 Гц. При натисканні на кнопку 'b' клавіатури починає блимати синій світлодіод з частотою 0.5 Гц. При натисканні на кнопку 'g' клавіатури починає блимати зелений світлодіод з частотою 0.5 Гц. При натисканні на кнопку Enter блимання світлодіодів припиняється. Кожний зі світлодіодів блимає з однаковою частотою. При першому натисканні на кнопку SW_2 частота блимання змінюється на 1.0 Гц. При другому натисканні на кнопку SW_2 частота блимання стає 2.0 Гц. Після третього натискання на кнопку SW_2 частота блимання стає знову 0.5 Гц і т. д. Кожна дія проекту супроводжується виводом інформації на термінал.