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.

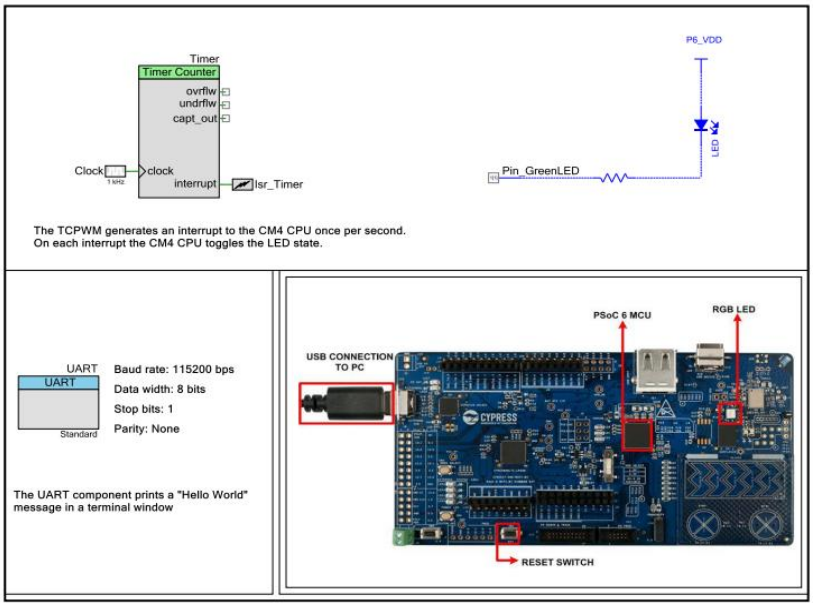


Рис. 1. Схема проекту

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.

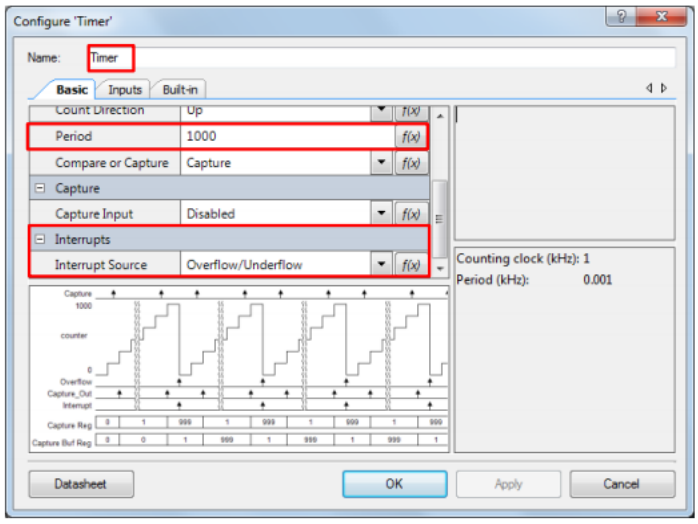


Рис. 2. Настройка конфігурації компонента таймера (TCPWM)

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

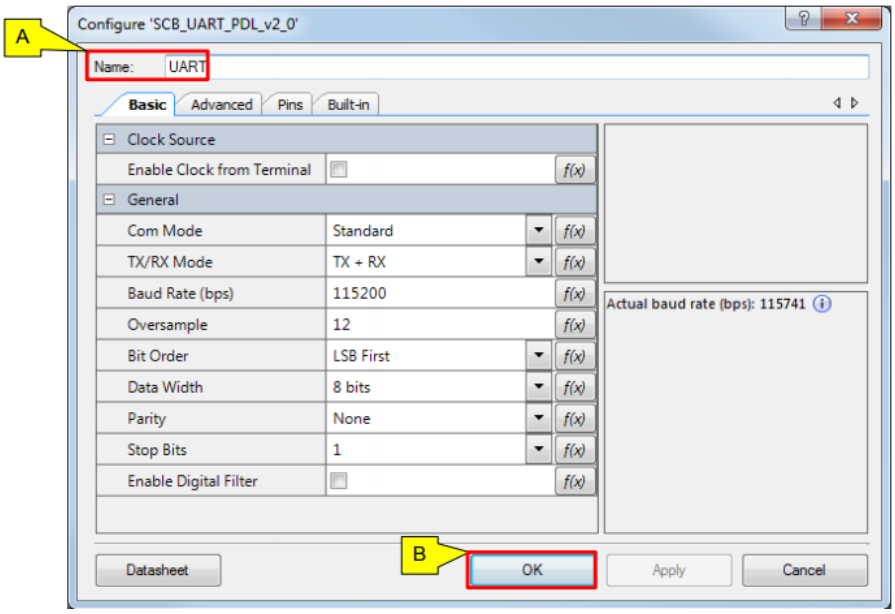


Рис. 2. Вікно налаштувань UART компонента на основі SCB

Налаштування компоненти 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).

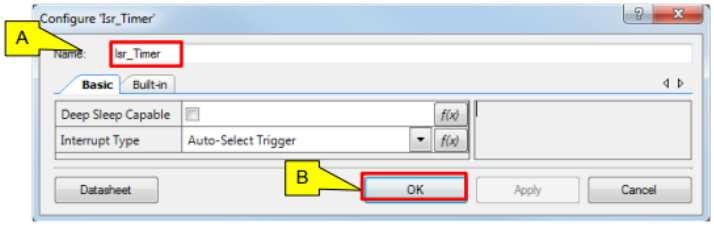


Рис. 3. Налаштування компоненти SysInt_PDL

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.

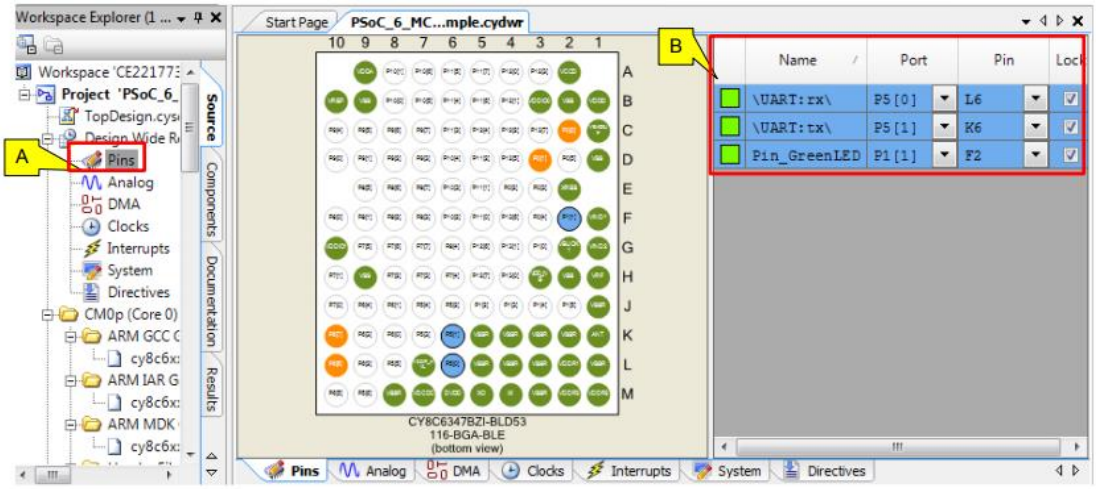


Рис. 4. Призначення виводів в схемі проекту

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).

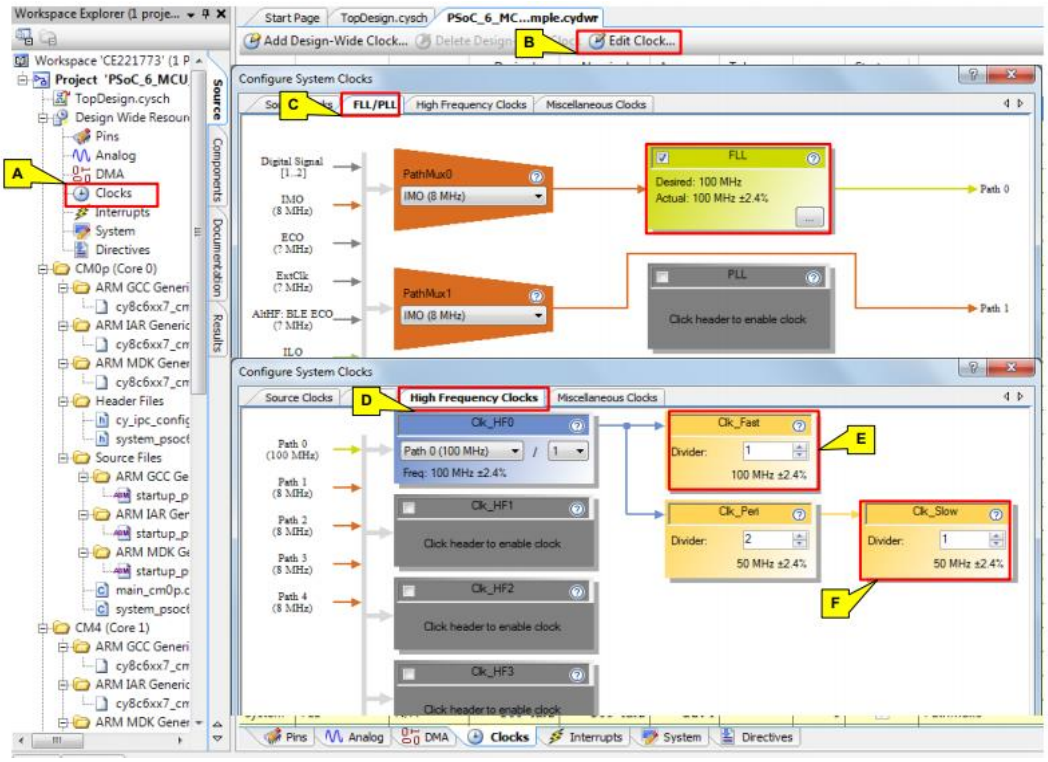
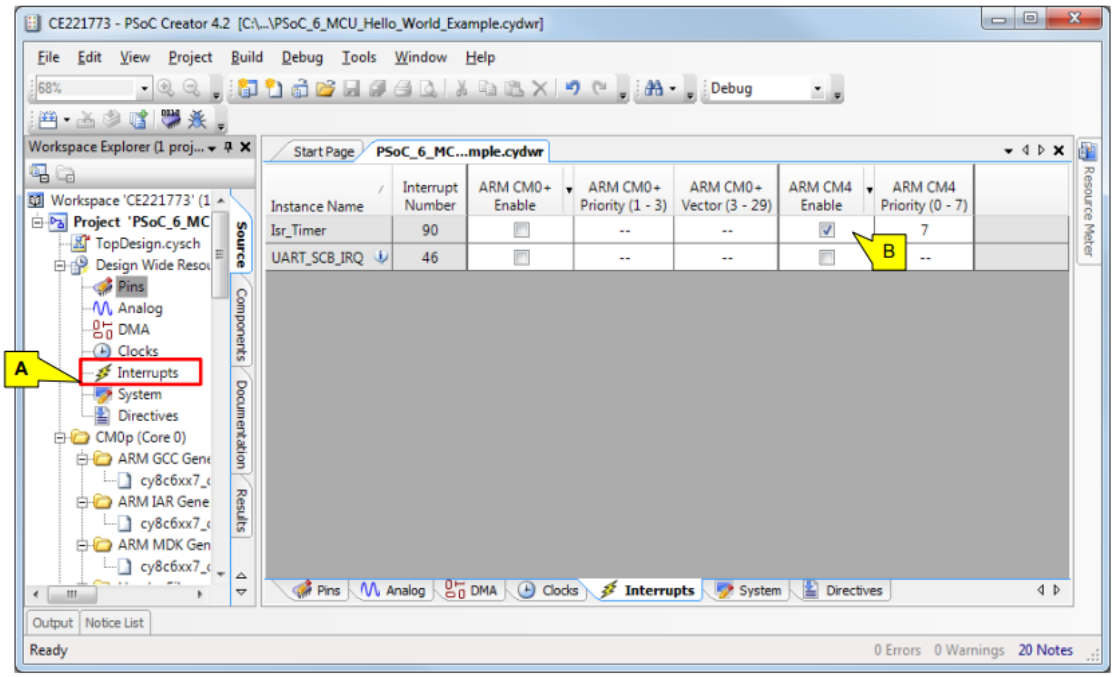


Рис. 5. Конфігурація тактових імпульсів.

Налаштування системи переривань (рис. 6).



Реалізація програмного забезпечення.

```c
/* Header files includes*/
#include "project.h"
/*******************************************************************************
* Macros
*******************************************************************************/
#define LED_ON     (0)
#define LED_OFF    (!LED_ON)
/*******************************************************************************
* Function Prototypes
*******************************************************************************/
void UartInit(void);
void TimerInit(void);
void Isr_Timer(void);


/*******************************************************************************
* Global Variables
*******************************************************************************/
bool LEDupdateFlag = false;
/*******************************************************************************
* Function Name: main
*******************************************************************************/
int main(void)
{
    /* Start the UART peripheral */
    UartInit();


    /* Enable global interrupts. */
    __enable_irq();


    /* \x1b[2J\x1b[;H - ANSI ESC sequence for clear screen */
    Cy_SCB_UART_PutString(UART_HW, "\x1b[2J\x1b[;H");


    Cy_SCB_UART_PutString(UART_HW, "******************CE221773 - PSoC 6 MCU:"\
                " Hello World! Example******************\r\n\n");


    Cy_SCB_UART_PutString(UART_HW, "Hello World!!!\r\n\n");

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

    /* Wait for the user to Press Enter key */
    while(Cy_SCB_UART_Get(UART_HW) != '\r');

    /* Start the TCPWM peripheral. TCPWM is configured as a Timer */
    TimerInit();

    Cy_SCB_UART_PutString(UART_HW, "Observe the LED blinking on the kit!!!\r\n");

    for(;;)
    {
        if(LEDupdateFlag)
        {
            /* Clear the flag */
            LEDupdateFlag = false;

            /* Invert the LED state*/
            Cy_GPIO_Inv(Pin_GreenLED_0_PORT, Pin_GreenLED_0_NUM);
        }
    }
}
/*******************************************************************************
* Function Name: UartInit
*******************************************************************************/
void UartInit(void)
{
    /* Configure the UART peripheral.
       UART_config structure is defined by the UART_PDL component based on

       parameters entered in the Component configuration*/

    Cy_SCB_UART_Init(UART_HW, &UART_config, &UART_context);
```

```
    /* Enable the UART peripheral */
    Cy_SCB_UART_Enable(UART_HW);
}
/*******************************************************************************
* Function Name: TimerInit
********************************************************************************/
void TimerInit(void)
{
    /* Configure the TCPWM peripheral.
       Counter_config structure is defined based on the parameters entered
       in the Component configuration */
    Cy_TCPWM_Counter_Init(Timer_HW, Timer_CNT_NUM, &Timer_config);

    /* Enable the initialized counter */
    Cy_TCPWM_Counter_Enable(Timer_HW, Timer_CNT_NUM);

    /* Start the enabled counter */
    Cy_TCPWM_TriggerStart(Timer_HW, Timer_CNT_MASK);

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

    /* Enable interrupt in NVIC */
    NVIC_EnableIRQ((IRQn_Type)Isr_Timer_cfg.intrSrc);
}
/*******************************************************************************
* Function Name: Isr_Timer
********************************************************************************/
void Isr_Timer(void)
{
    /* Clear the TCPWM peripheral interrupt */
    Cy_TCPWM_ClearInterrupt(Timer_HW, Timer_CNT_NUM, CY_TCPWM_INT_ON_TC );

    /* Clear the CM4 NVIC pending interrupt for 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.

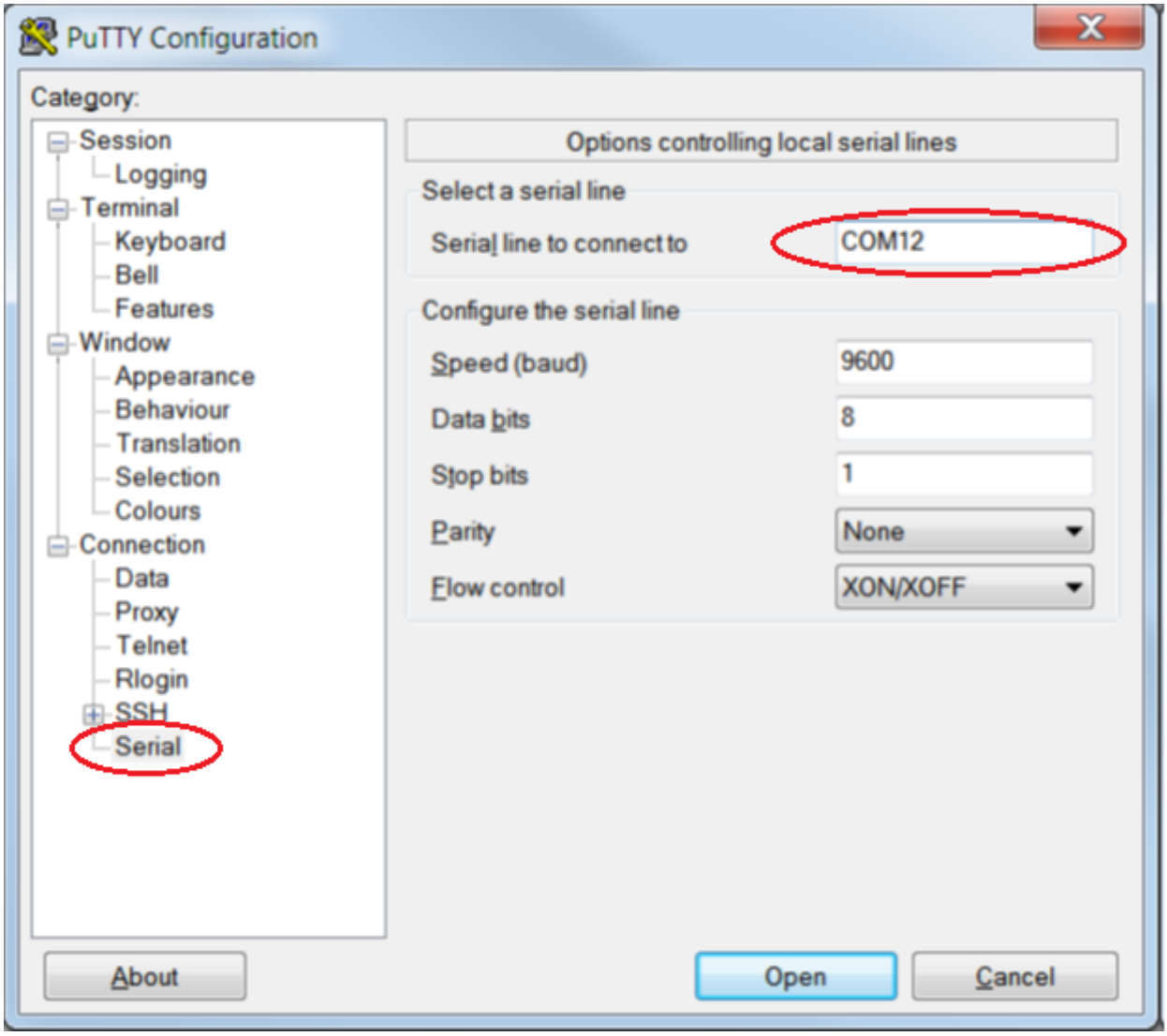


Рис. 3. Вигляд головного вікна програми Putty

В даному вікні в блоці “Category” вибираємо категорію “Serial” як вказано на рисунку, а в блоці “Serial line to connect to” вказуємо відповідний номер COM-роз’єму до якого підключений стенд PSoC 6. Якщо номер роз’єму не відомий, то його можна взнати зайшовши в меню «Диспетчер пристроїв». Вказавши відповідне значення COM-порта, до якого приєднано PSoC 6, нижче в пункті “Speed (baud)” вказуємо значення швидкості передаxs даних. Це значення повинно співпадати із значенням? яке вказується при налаштування модуля UART.

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

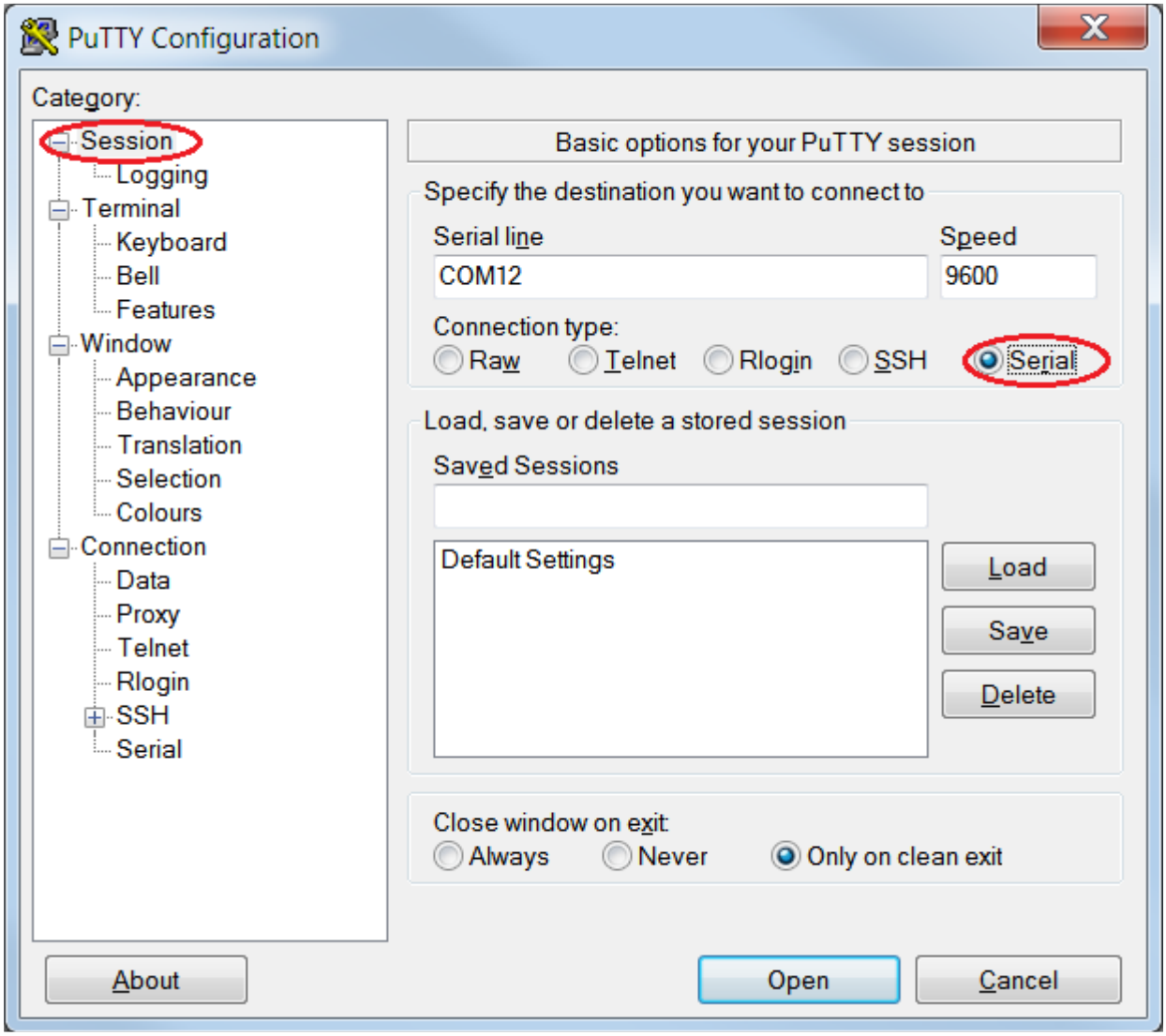


Рис. 4. Налаштування даних вкладки Session програми Putty

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

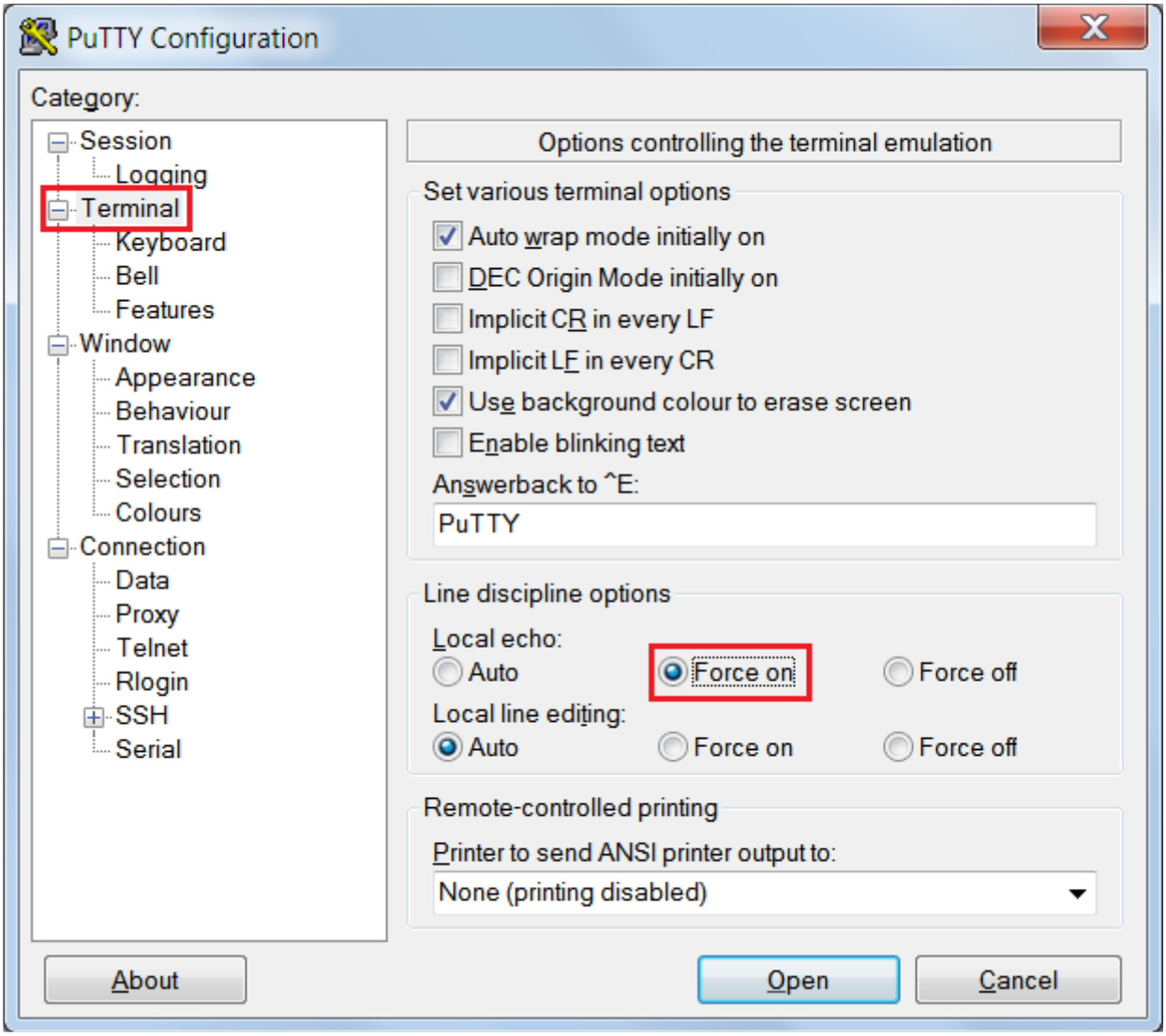


Рис. 5. Налаштування вкладки Terminal програми Putty

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

Опис літератури:

Завдання.