

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кафедра САП



Лабораторна робота №1

з дисципліни «Програмування інтелектуальних вбудованих систем»

на тему: “Процес розроблення програм на мові C на мікроконтролерах STM32 та інтегрованому середовищі розробки програмного забезпечення CubeIDE”

Виконав

ст. гр. ПП-31

Гаврилюк Назар

Прийняв:

Колесник К.К.

Мета роботи. Ознайомитися з платою розробника STM32F4-Discovery та середовищем розробки STM32CubeIDE. Також ознайомитися з можливостями STM32CubeMX і запустити простий проект на платі для розробки.

Завдання.

Повторити налаштування проекту описане в лабораторній роботі. Дівчата №1, хлопці №2.

Хід роботи

Я Завантажив STM32CubeIDE по силці, після чого запустив та вибрав потрібну схему (Рис. 1).

The screenshot displays the STM32CubeMX Board Selector interface. On the left, the 'Board Filters' sidebar is visible, showing filters for Commercial Part Number (STM32F407G-DISC1), PRODUCT INFO (Type, Supplier, MCU / MPU Series, Marketing Status, Price), MEMORY (Ext. Flash, Ext. EEPROM, Ext. RAM), and FEATURES (Embedded Sensor, User Button, Camera, CAN, Connector). The main area shows the 'STM32F4 Series' with the selected board 'STM32F407G-DISC1' (Discovery kit with STM32F407VG MCU). It includes a description, a photo of the board, and a table of features. Below the main area, a 'Boards List' table shows the selected board.

Commercial Part No.	Type	Marketing Status	Unit Price (US\$)	Mounted Device
STM32F407G-DISC1	Discovery Kit	Active	19.9	STM32F407VG

Рис. 1. Вибір Схеми STM32F407G-Disc1

Налаштував плату у розділі Pinout & Configuration (Рис.2).

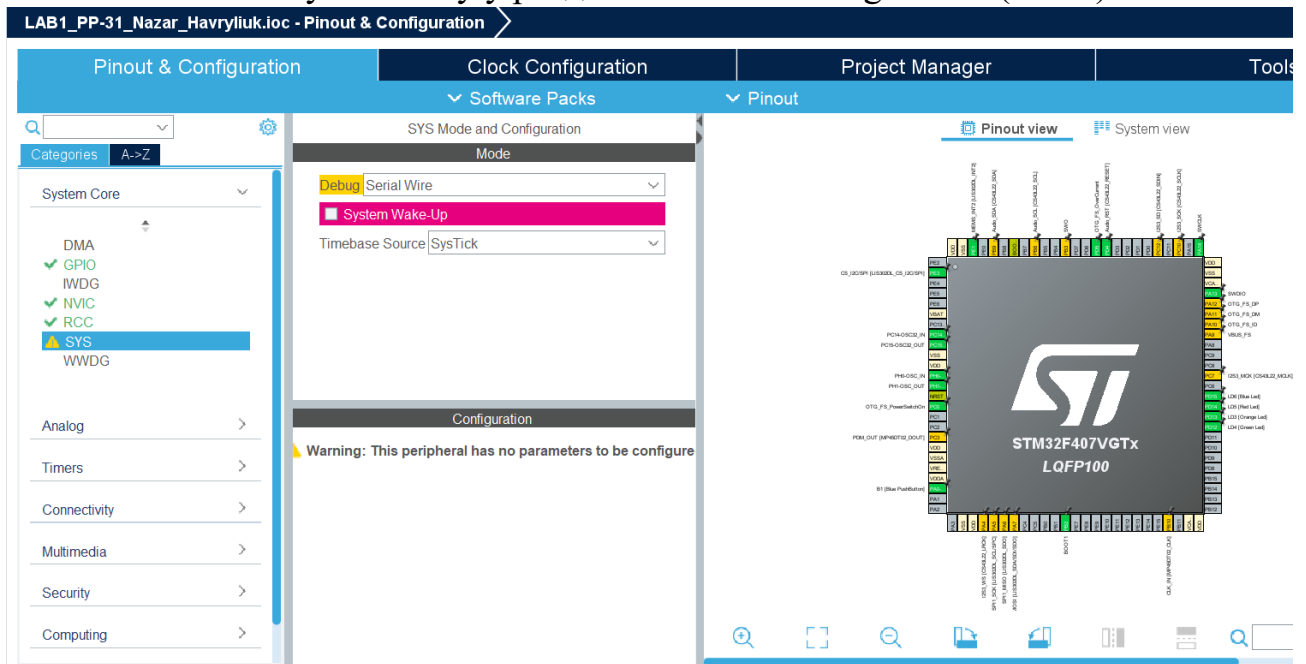


Рис. 2. Вибір джерела відладки

Налаштував конфігурацію (Рис.3).

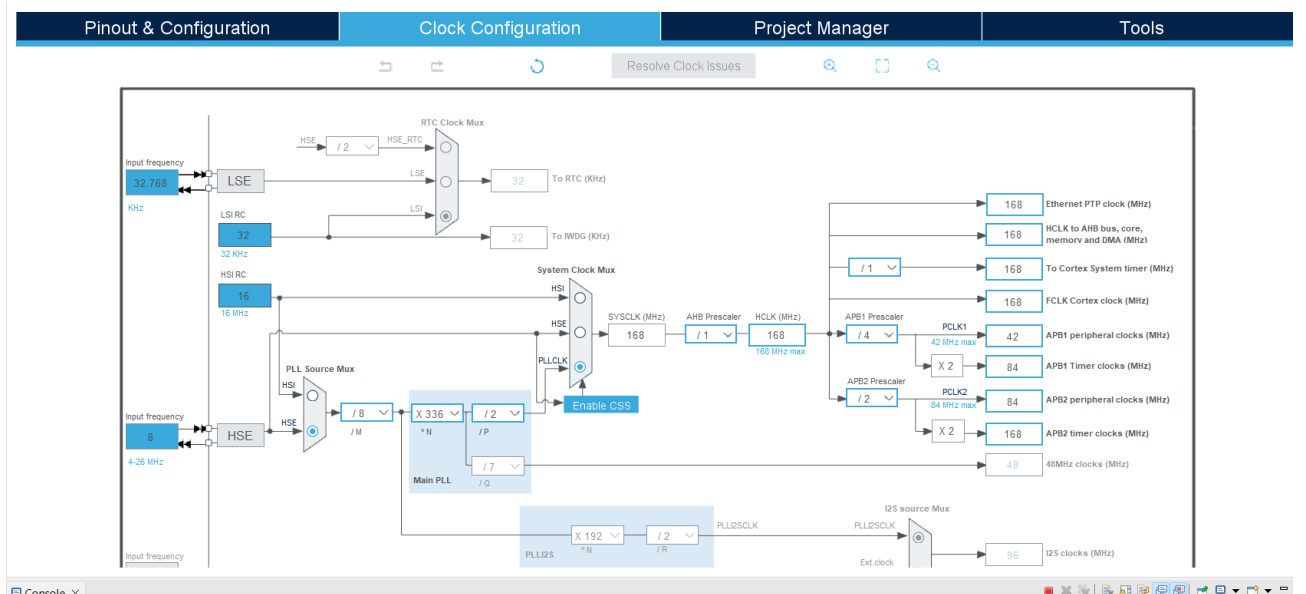


Рис. 3. Вкладка Clock Configuration

Написав код в програмному середовищі STM32CubeIDE (Рис.4) після чого запустив його.

```

94  /* USER CODE BEGIN WHILE */
95  while (1)
96  {
97      /* USER CODE END WHILE */
98
99      /* USER CODE BEGIN 3 */
100     if((HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_0) == GPIO_PIN_SET))
101     {
102         HAL_GPIO_WritePin(GPIOD, GPIO_PIN_12, GPIO_PIN_SET);
103         HAL_GPIO_WritePin(GPIOD, GPIO_PIN_13, GPIO_PIN_SET);
104         HAL_GPIO_WritePin(GPIOD, GPIO_PIN_14, GPIO_PIN_SET);
105         HAL_GPIO_WritePin(GPIOD, GPIO_PIN_15, GPIO_PIN_SET);
106     }
107     else
108     {
109         HAL_GPIO_WritePin(GPIOD, GPIO_PIN_12, GPIO_PIN_RESET);
110         HAL_GPIO_WritePin(GPIOD, GPIO_PIN_13, GPIO_PIN_RESET);
111         HAL_GPIO_WritePin(GPIOD, GPIO_PIN_14, GPIO_PIN_RESET);
112         HAL_GPIO_WritePin(GPIOD, GPIO_PIN_15, GPIO_PIN_RESET);
113     }
114 }
115 /* USER CODE END 3 */

```

Рис. 4. Код програми.

Виконав індивідуальне завдання видане лаборантом.

Код програми:

int i=0;

while (1)

{

/* USER CODE END WHILE */

/* USER CODE BEGIN 3 */

if((HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_0) == *GPIO_PIN_SET*))

{

switch (i){

case 0:

```

GPIO_PIN_SET);

GPIO_PIN_SET);

GPIO_PIN_RESET))

        HAL_GPIO_WritePin(GPIOD, GPIO_PIN_12,

        HAL_GPIO_WritePin(GPIOD, GPIO_PIN_13,

        if((HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_0) ==

                {i++;}

        break;

    case 1:

        HAL_GPIO_WritePin(GPIOD, GPIO_PIN_13,

        GPIO_PIN_SET);

        HAL_GPIO_WritePin(GPIOD, GPIO_PIN_14,

        GPIO_PIN_SET);

        if((HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_0) ==

                {i++;}

        break;

    case 2:

        HAL_GPIO_WritePin(GPIOD, GPIO_PIN_14,

        GPIO_PIN_SET);

        HAL_GPIO_WritePin(GPIOD, GPIO_PIN_15,

        GPIO_PIN_SET);

        if((HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_0) ==

                {i++;}

        break;

    case 3:

        HAL_GPIO_WritePin(GPIOD, GPIO_PIN_15,

        GPIO_PIN_SET);

        HAL_GPIO_WritePin(GPIOD, GPIO_PIN_12,

        GPIO_PIN_SET);

        if((HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_0) ==

                {i=i-3;}

```

```

        break;
    }

}

else
{
    HAL_GPIO_WritePin(GPIOD, GPIO_PIN_12, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOD, GPIO_PIN_13, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOD, GPIO_PIN_14, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOD, GPIO_PIN_15, GPIO_PIN_RESET);
}
}

/* USER CODE END 3 */
}i

```

Результат відео додав у файл.

Теоретичні питання на захист.

1. Що таке STM32CubeIDE?

STM32CubeIDE – це середовище розробки (IDE) для мікроконтролерів STM32, яке об'єднує редактор коду, компілятор, відлагоджувач і вбудований STM32CubeMX для генерації коду.

2. Що таке STM32CubeMX?

STM32CubeMX – це інструмент для графічної конфігурації мікроконтролера STM32: вибір моделі, налаштування пінів, тактовань, периферії та автоматична генерація коду.

3. В вікні CubeMX пояснити призначення вкладки Pinout & Configuration?

Pinout & Configuration – вкладка у CubeMX, де обираються функції для виводів мікроконтролера (GPIO, UART, SPI тощо) та вмикаються/вимикаються внутрішні периферійні модулі. Це «карта пінів» мікроконтролера.

4. В вікні CubeMX призначення вкладки Clock Configuration?

Clock Configuration – вкладка для налаштування джерел і частот тактування (HSE, HSI, PLL тощо). Тут ви задаєте, з якою швидкістю буде працювати ядро, шини й периферія.

5. Поясніть призначення функцій HAL_GPIO_WritePin, HAL_GPIO_ReadPin, HAL_GPIO_TogglePin та HAL_Delay.

HAL_GPIO_WritePin – встановлює рівень на виводі (логічний 0 або 1). Використовується для вмикання/вимикання світлодіодів, реле тощо.

HAL_GPIO_ReadPin – зчитує поточний стан виводу (0 або 1), наприклад, щоб дізнатися, чи натиснута кнопка.

HAL_GPIO_TogglePin – змінює стан виводу на протилежний (якщо був 0 → стане 1, і навпаки). Зручно для миготіння LED.

HAL_Delay – створює затримку у мілісекундах (наприклад, HAL_Delay(1000) = 1 секунда).

Висновок:

На лабораторній роботі я ознайомився з платою розробника STM32F4-Discovery та середовищем розробки STM32CubeIDE. Також ознайомився з можливостями STM32CubeMX і запустив простий проект на платі для розробки.

.