

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кафедра САП



Лабораторна робота №2

з дисципліни «Програмування інтелектуальних вбудованих систем»
на тему: “ Програмування GPIO та внутрішніх периферій мікроконтролера
STM32”

Виконав

ст. гр. ПП-31

Гаврилюк Назар

Прийняв:

Колесник К.К.

Львів – 2025

Мета роботи. ознайомитися з GPIO, перериваннями, таймерами та ШІМ.
Практично реалізувати проект в групі з їх застосуванням.

Завдання.

Прив'язати кнопки на платі розширення (SWT1-5) до світлодіодів на платі розробника. На початку всі діоди не світяться. Кнопки і світлодіоди розміщені у вигляді стрілок, відповідно кожна кнопка має включати свій світлодіод. Умовно права кнопка включає правий діод, повторне натиснення – вимикає. Середня кнопка вимикає всі діоди (якщо хоч один включений), а повторне – вимикає.

Код програми (функція main):

```
int main(void)
```

```
{
```

```
/* USER CODE BEGIN 1 */
```

```
/* USER CODE END 1 */
```

```
/* MCU Configuration-----*/
```

```
/* Reset of all peripherals, Initializes the Flash interface and the Systick. */
```

```
HAL_Init();
```

```
/* USER CODE BEGIN Init */
```

```
/* USER CODE END Init */
```

```
/* Configure the system clock */
```

```
HAL_Init();
```

```
SystemClock_Config();
```

```
MX_GPIO_Init();
```

```
/* USER CODE BEGIN SysInit */
```

```
/* USER CODE END SysInit */
```

```
/* Initialize all configured peripherals */
```

```
/* USER CODE BEGIN 2 */
```

```
/* USER CODE END 2 */
```

```

/* Infinite loop */
/* USER CODE BEGIN WHILE */

bool flagCenter=false;
bool sw1Flag=true;
bool sw2Flag=true;
bool sw3Flag=true;
bool sw4Flag=true;
bool CenterFlag=true;

uint32_t lastCenterTime = 0;
uint32_t lastSw1Time = 0, lastSw2Time = 0, lastSw3Time = 0, lastSw4Time = 0;
const uint32_t debounceDelay = 50; // MC

while (1)
{
    uint32_t currentTime = HAL_GetTick();

    if(HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_15) == GPIO_PIN_RESET && flagCenter == false && CenterFlag == false)
    {
        if(currentTime - lastCenterTime >= debounceDelay)
        {
            HAL_GPIO_WritePin(GPIOD, GPIO_PIN_12, GPIO_PIN_SET);
            HAL_GPIO_WritePin(GPIOD, GPIO_PIN_13, GPIO_PIN_SET);
            HAL_GPIO_WritePin(GPIOD, GPIO_PIN_14, GPIO_PIN_SET);
            HAL_GPIO_WritePin(GPIOD, GPIO_PIN_15, GPIO_PIN_SET);
            flagCenter = true;
            lastCenterTime=currentTime;
            CenterFlag=true;
        }
    }

    else if(HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_15) == GPIO_PIN_RESET && flagCenter == true && CenterFlag == false)
    {
        if(currentTime - lastCenterTime >= debounceDelay)
        {
            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);
    flagCenter = false;
    lastCenterTime = currentTime;
    CenterFlag = true;
}

}

else if(HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_15) == GPIO_PIN_SET)
{
    CenterFlag = false;
}

// ===== SW1 (ліва кнопка, PC9 -> LED3 PD13) =====
if(HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_9) == GPIO_PIN_SET && sw1Flag == false)
{
    if(currentTime - lastSw1Time >= debounceDelay)
    {
        HAL_GPIO_TogglePin(GPIOD, GPIO_PIN_13);
        sw1Flag = true;
        lastSw1Time = currentTime; // <- ОБОВ'ЯЗКОВО оновлюємо час
        lastSw1Time = currentTime;
    }
}

else if(HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_9) == GPIO_PIN_RESET)
{
    sw1Flag = false;
}

}

// ===== SW2 (верхня кнопка, PC6 -> LED4 PD12) =====
if(HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_6) == GPIO_PIN_SET && sw2Flag == false)
{
    if(currentTime - lastSw2Time >= debounceDelay)
    {
        HAL_GPIO_TogglePin(GPIOD, GPIO_PIN_14);
        sw2Flag = true;
    }
}

```

```

lastSw2Time = currentTime; // <- ОБОВ'ЯЗКОВО оновлюємо час
}

}

else if(HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_6) == GPIO_PIN_RESET)
{
    sw2Flag = false;
}

// ===== SW3 (права кнопка, PC8 -> LED5 PD14) =====
if(HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_8) == GPIO_PIN_SET && sw3Flag == false)
{
    if(currentTime - lastSw3Time >= debounceDelay)
    {
        HAL_GPIO_TogglePin(GPIOD, GPIO_PIN_12);
        sw3Flag = true;
        lastSw3Time = currentTime;
    }
}

else if(HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_8) == GPIO_PIN_RESET)
{
    sw3Flag = false;
}

if(HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_11) == GPIO_PIN_SET && sw4Flag == false)
{
    if(currentTime - lastSw4Time >= debounceDelay)
    {
        HAL_GPIO_TogglePin(GPIOD, GPIO_PIN_15);
        sw4Flag = true;
        lastSw4Time = currentTime;
    }
}

else if(HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_11) == GPIO_PIN_RESET)
{
    sw4Flag = false;
}

```

```
/* USER CODE BEGIN 3 */  
}  
/* USER CODE END 3 */  
}
```

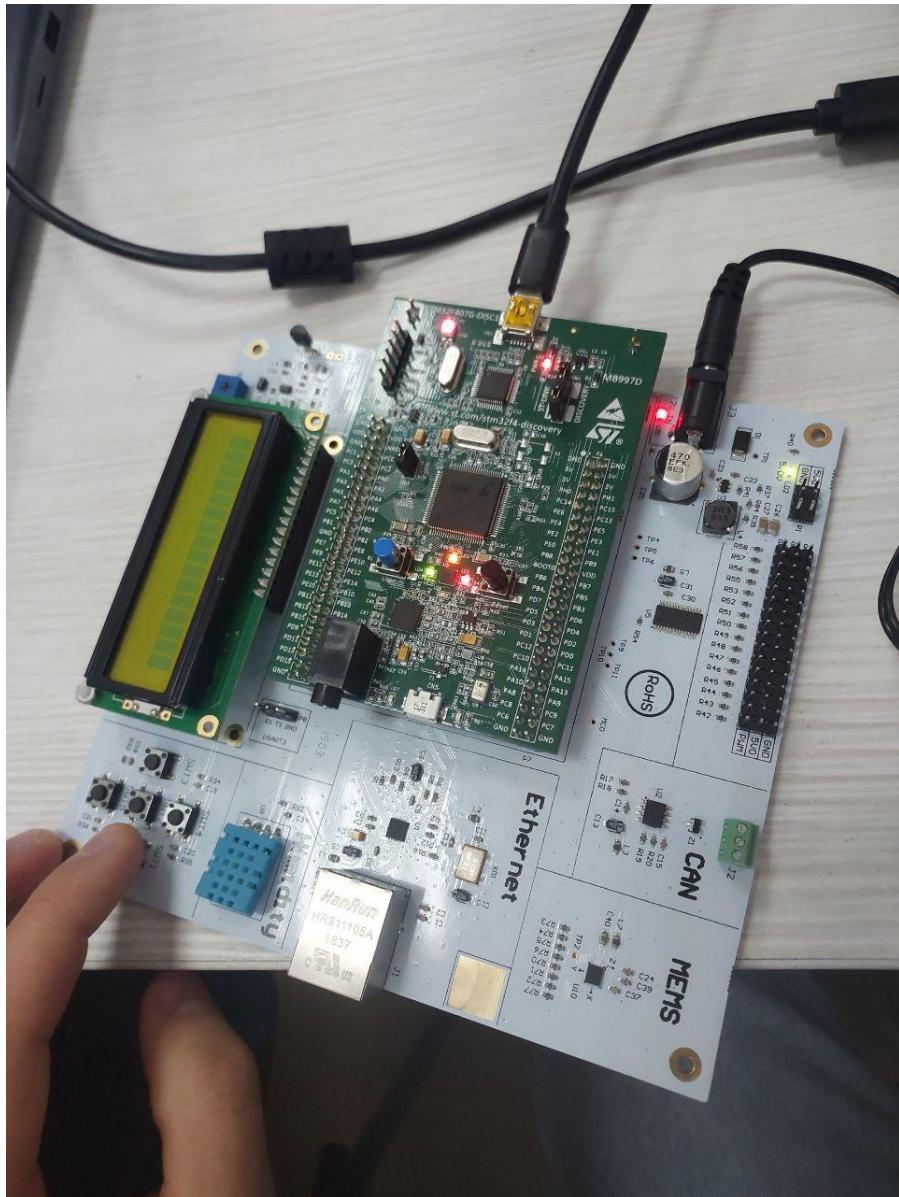


Рис. 1. Результат роботи

Контрольні запитання:

1. Пояснити як працює ваш код.

Код опитує кнопки на портах GPIO з антидребезговою затримкою (debounceDelay). При натисканні центральної кнопки всі світлодіоди вмикаються або вимикаються, а інші кнопки (SW1–SW4) керують окремими світлодіодами через HAL_GPIO_TogglePin. Для відстеження натискань використані працпорці та час останнього натискання.

2. Що таке GPIO? Якщо була вимога використати в варіанті:

GPIO (General Purpose Input/Output) — загального призначення вхід/вихід мікроконтролера, який дозволяє читати стани кнопок або керувати світлодіодами, реле тощо.

3. Що таке переривання?

Переривання — механізм, коли мікроконтролер миттєво припиняє основний код, щоб виконати спеціальну функцію (обробник), наприклад, при натисканні кнопки або закінченні таймера.

4. Що таке таймер в мікроконтролері?

Таймер у мікроконтролері — внутрішній лічильник, який генерує події через заданий проміжок часу, використовується для затримок, створення періодичних сигналів чи вимірювання часу.

5. Що таке ШІМ?

ШІМ (PWM, Pulse-Width Modulation) — широтно-імпульсна модуляція, дозволяє керувати яскравістю LED або швидкістю двигунів шляхом зміни співвідношення часу увімкнено/вимкнено.

6. Принци роботи LCD WH1602.

LCD WH1602 — текстовий рідкокристалічний дисплей на 2 рядки по 16 символів; працює через 4- або 8-бітний інтерфейс, приймає команди для керування курсором, очищення дисплея і відображення символів.

Висновок:

На лабораторній роботі я ознайомився з GPIO, перериваннями, таймерами та ШІМ. Практично реалізував проект з їх застосуванням.