Національний технічний університет України

“ Київський політехнічний інститут ”



Лабораторна робота №2

По курсу: ” МІКРОПРОЦЕСОРНІ СИСТЕМИ НА БАЗІ ARM ПРОЦЕСОРІВ*”*

Тема: “ Передача даних інтерфейсом UART ”

Варіант 6

Виконав:Студент І курсу

Групи ДС-31мп

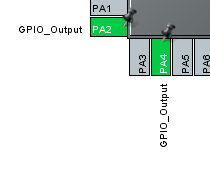
Рєзнік Олександр Сергійович

Київ 2024 р.

***Завдання***

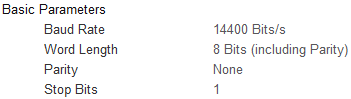
**Передача-прийом в режимі DMA**

1. Налаштувати виводи порту РА на вихід в режимі “push-pull” відповідно до номеру (РА6) та подвоєного (РА12) номеру варіанту.



1. Налаштувати роботу передавача UART в режимі DMA з параметрами передачі:  
   Швидкість: 6 варіант – 9600 бод; Довжина повідомлення – 8 біт;

Біт парності – відсутній; Стоп-біт – 1.



1. Створити програмний код, що реалізує передачу повідомлення з номером варіанту.
2. Налаштувати роботу приймача UART в режимі DMA з аналогічними параметрами.
3. Створити програмний код, що після прийому повідомлення порівнює отримане значення з номером варіанту, та при підтвердженні рівності встановлює значення логічної одиниці на виході порту РА визначеного номером варіанту. Якщо надходить інше значення – встановити значення логічної одиниці на вихід порту РА значення якого є подвоєним до номеру варіанту.
4. Довести роботоспроможність коду через застосування передавача з п. 2,3 та через застосування терміналу в середовищі Proteus.

**Передача в режимі Blocking-прийом в режимі Interruption**

1. Налаштувати виводи порту РB на вихід в режимі “push-pull” відповідно до номеру (РB6) та подвоєного (РB12) номеру варіанту.
2. Налаштувати роботу передавача UART в режимі Blocking з параметрами передачі аналогічними до п.2 цієї роботи.
3. Створити програмний код, що реалізує передачу повідомлення з англійською транслітерацією Вашого прізвища.
4. Налаштувати роботу приймача UART в режимі роботи з перериванням по прийому (Interrupt mode) з аналогічними параметрами.
5. Створити програмний код, що після прийому повідомлення порівнює отримане значення з транслітерацією Вашого прізвища, та при підтвердженні рівності встановлює значення логічної одиниці на виході порту РА визначеного номером варіанту(РА6). Якщо надходить інше значення – встановити значення логічної одиниці на вихід порту РА значення якого є подвоєним до номеру варіанту(РА12).
6. Довести роботоспроможність коду через застосування терміналу в середовищі Proteus.
7. Налаштувати приймач на роботу в блокуючому режимі. Виконати дії відповідно до п 11,12.

**Вимоги до звіту**

1. Титульна сторінка
2. Завдання варіанту, визначене через розрахункові формули
3. Лістинг програмного коду з коментарями
4. Скріншоти реалізації моделі в середовищі “Proteus”, що демонструють її роботоспроможність
5. Короткі висновки за роботою

uint8\_t rx[1];

HAL\_UART\_Receive\_DMA(&huart1, &rx[1], **sizeof**(rx));

**void** **HAL\_UART\_RxCpCallback**(UART\_HandleTypeDef \*huart)

{

HAL\_GPIO\_TogglePin(GPIOA, GPIO\_PIN\_6);

uint8\_t flag=rx[0];

**if** (flag=='1')

{

HAL\_GPIO\_TogglePin(GPIOA, GPIO\_PIN\_5);

}

**else**

{

HAL\_GPIO\_TogglePin(GPIOA, GPIO\_PIN\_7);

}

HAL\_UART\_Receive\_DMA(&huart1, &rx[0], **sizeof**(rx));

}

………………………………………………………………….

/\* USER CODE BEGIN PV \*/

uint8\_t array[1];

uint8\_t flag;

/\* USER CODE END PV \*/

…

/\* USER CODE END 2 \*/

HAL\_UART\_Transmit\_DMA(&huart1, &array[1], 1);

…

**void** **HAL\_UART\_TxCpltCallback**(UART\_HandleTypeDef \*huart)

{

HAL\_GPIO\_TogglePin(GPIOB, GPIO\_PIN\_1);

flag=array[0];

flag++;

array[0]=flag;

HAL\_UART\_Transmit\_DMA(&huart1, &flag, 1);

}

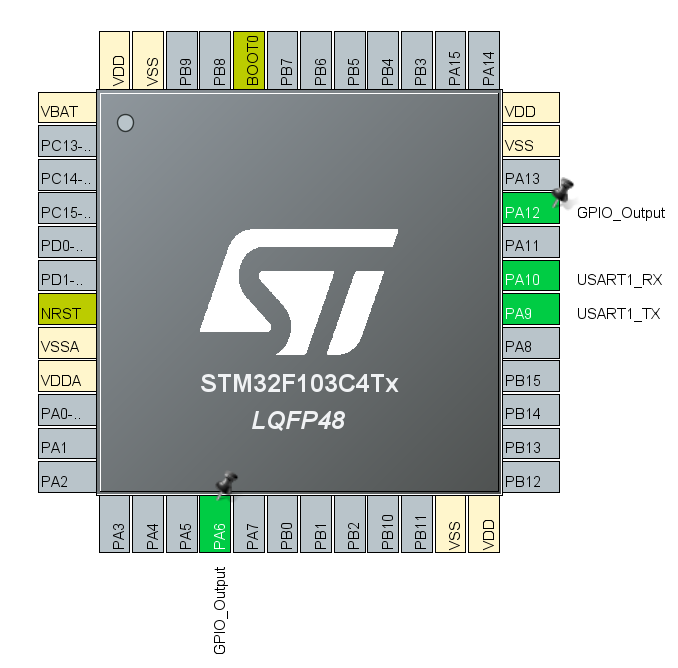


Рис. 1 Конфігурація пінів для Передачі-прийому в режимі DMA

Лістинг програми:

**#include** <stdbool.h>

…

/\* USER CODE BEGIN PV \*/

uint8\_t rx[1]; // буфер з інформацією

/\* USER CODE END PV \*/

…

**int** **main**(**void**)

{

/\* USER CODE BEGIN 2 \*/

HAL\_UART\_Receive\_DMA(&huart1, &rx[1], **sizeof**(rx)); // (адреса юарт інтерфейсу, адреса буфера, кількість даних в байтах)

/\* USER CODE END 2 \*/

}

…

/\* USER CODE BEGIN 4 \*/

// функція HAL\_UART\_RxCpltCallback

**void** **HAL\_UART\_RxCpltCallback**(UART\_HandleTypeDef \*huart)

{

**if** (huart->Instance == USART1) // Перевірка, чи callback викликано для правильного UART

{

**if** (rx[0] == '6') // Якщо отримано номер варіанту

{

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_6, *GPIO\_PIN\_SET*); // Встановити PA6

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_12, *GPIO\_PIN\_RESET*); // Вимкнуть PA12

}

**else**

{

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_12, *GPIO\_PIN\_SET*); // Інакше встановити PA12

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_6, *GPIO\_PIN\_RESET*); // Вимкнуть PA6

}

HAL\_UART\_Receive\_DMA(&huart1, &rx[0], **sizeof**(rx)); // Перезапуск прийому

}

}

/\* USER CODE END 4 \*/

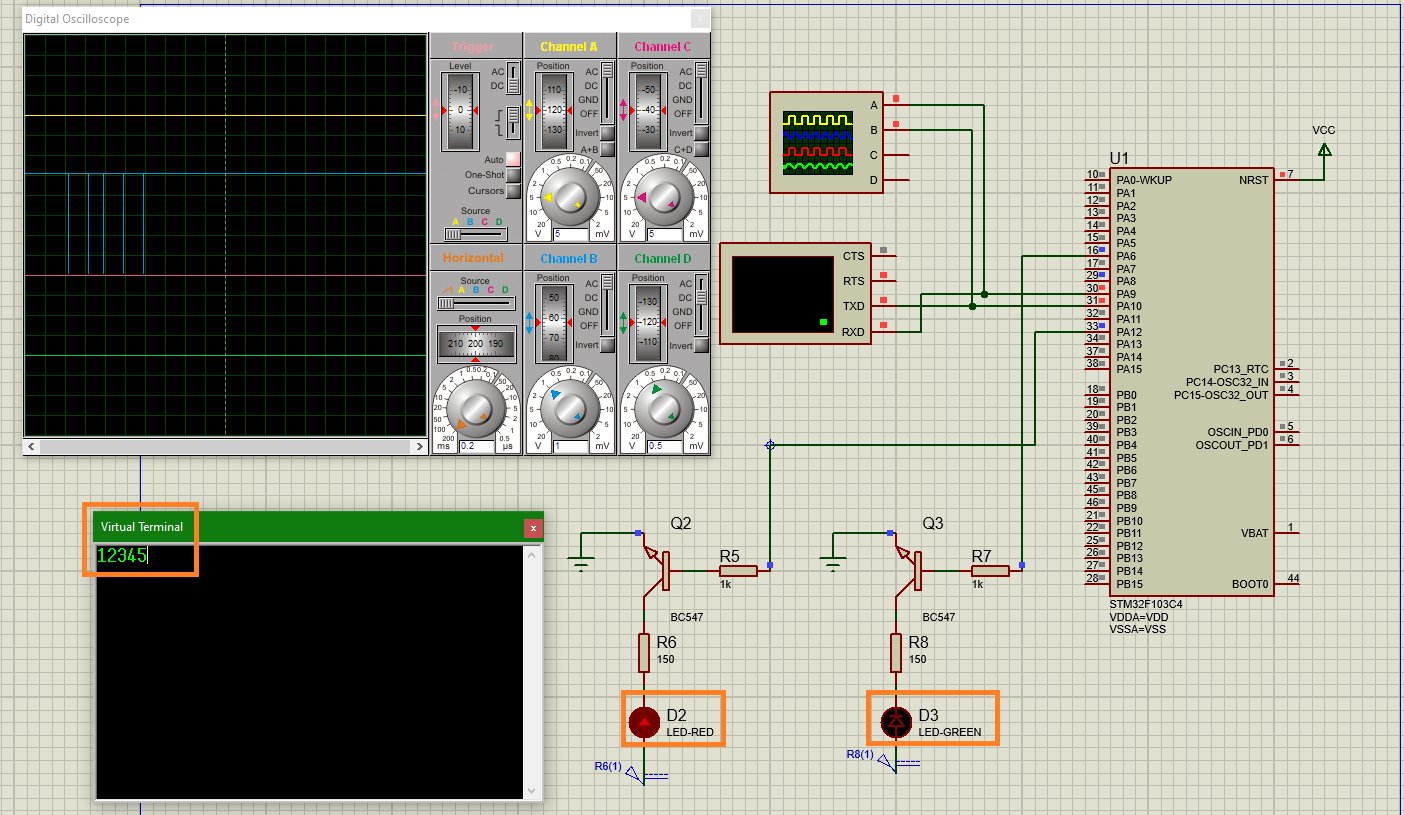


Рис. 2 Повідомлення в терміналі не відповідає номеру варіанту

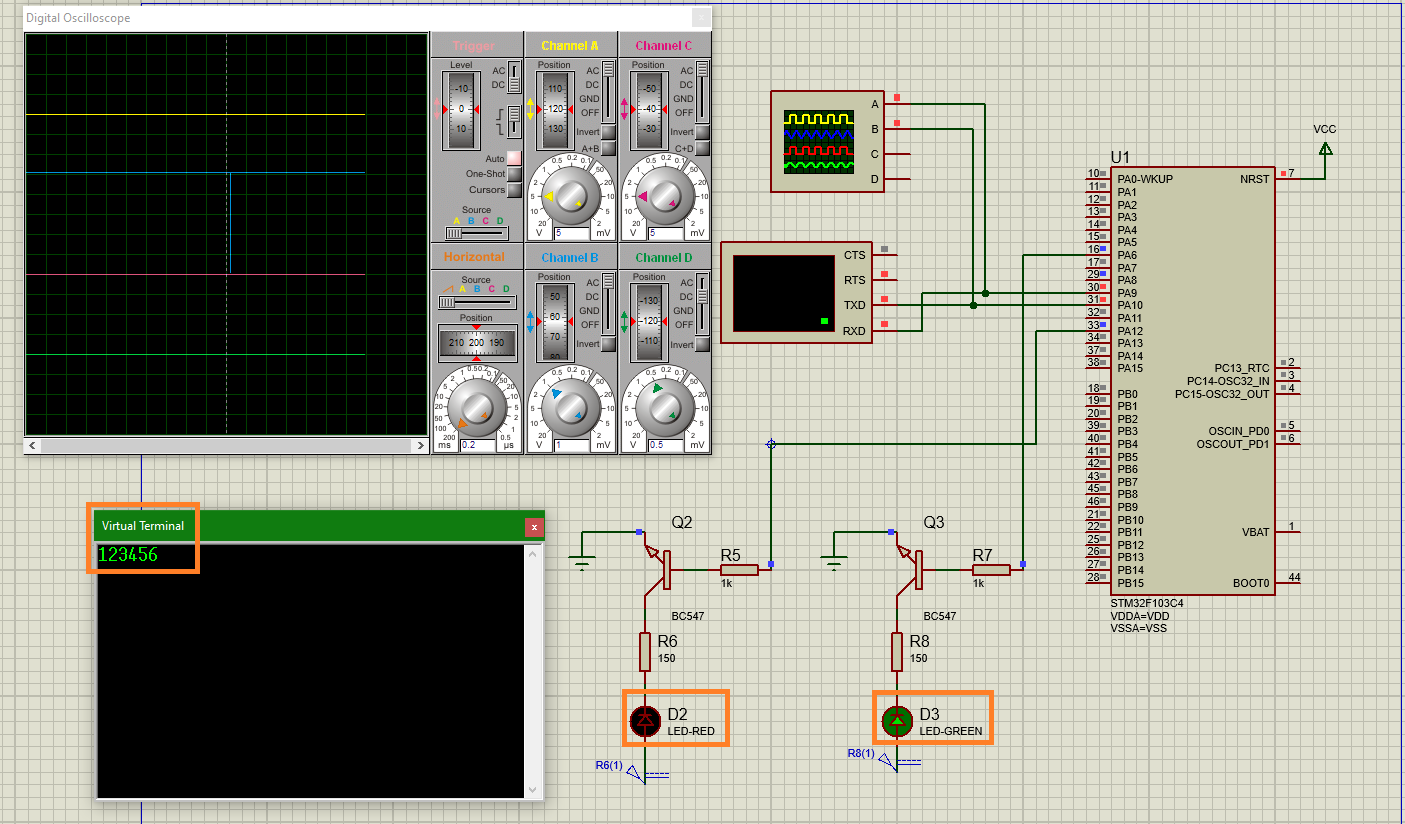


Рис. 3 Повідомлення в терміналі відповідає номеру варіанту

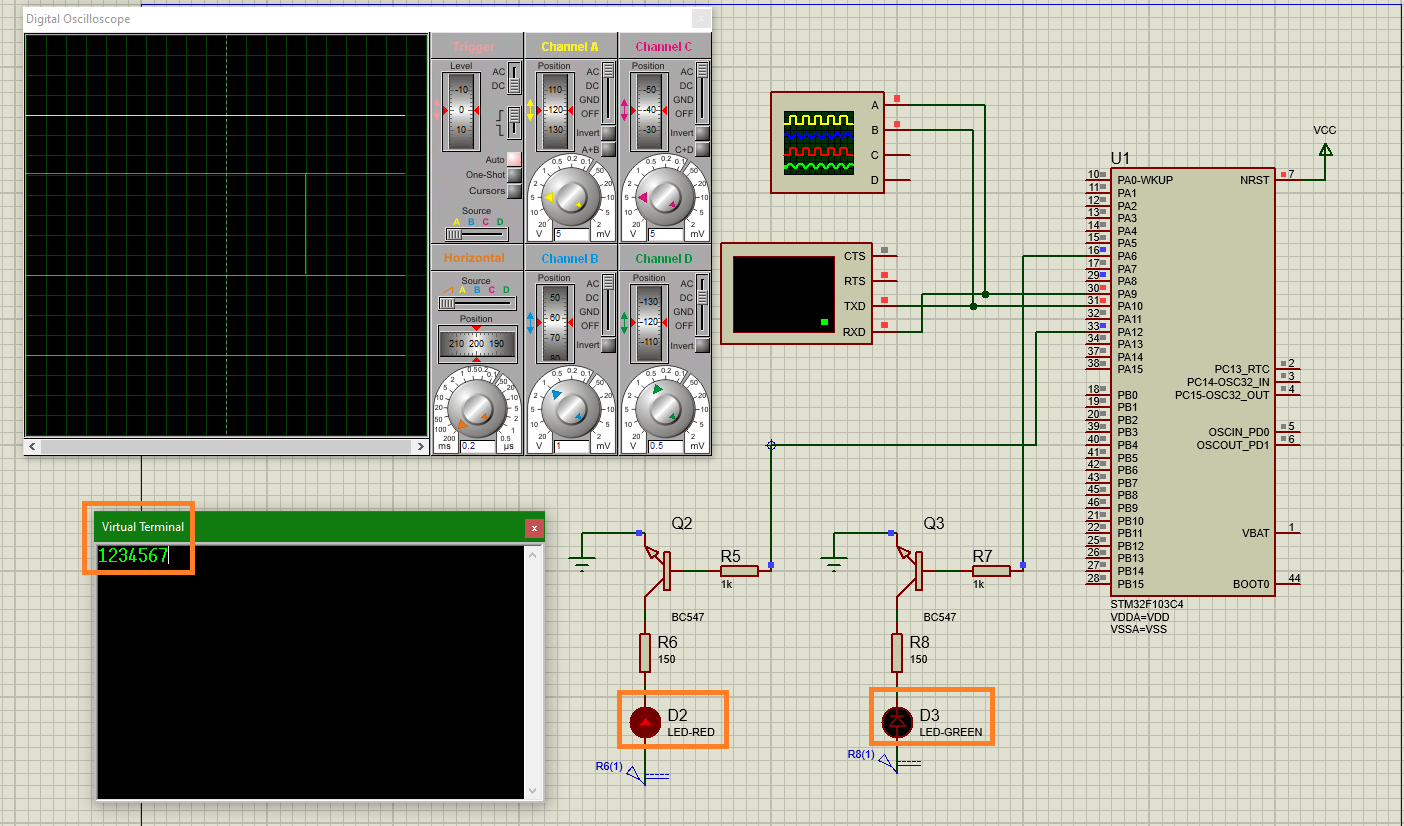


Рис. 4 Повідомлення в терміналі знову не відповідає номеру варіанту

Лістинг програми:

/\* USER CODE BEGIN PV \*/

**const** uint8\_t rx\_buffer[7] = {"rieznik"}; // Розмір має відповідати довжині транслітерації прізвища

uint8\_t array[1];

uint8\_t flag; // додаткова змінна

/\* USER CODE END PV \*/

…

/\* USER CODE BEGIN 2 \*/

HAL\_UART\_Transmit\_DMA(&huart1, &array[1], 1);

/\* USER CODE END 2 \*/

/\* USER CODE BEGIN 4 \*/

**void** **UART\_DMATransmitCplt**(DMA\_HandleTypeDef \*hdma)

{

}

**void** **HAL\_UART\_TxCpltCallback**(UART\_HandleTypeDef \*huart)

{

// Передає вміст rx\_buffer через UART, використовуючи блокуючу передачу.

HAL\_UART\_Transmit(&huart1, (uint8\_t\*)rx\_buffer, **sizeof**(rx\_buffer), HAL\_MAX\_DELAY);

// Читає перший байт з масиву array і зберігає його у змінній flag.

flag=array[0];

// Збільшує значення flag на одиницю.

flag++;

// Записує оновлене значення flag назад у перший елемент масиву array.

array[0]=flag;

// Передає оновлене значення flag через UART, використовуючи передачу через DMA.

HAL\_UART\_Transmit\_DMA(&huart1, &flag, 1);

}

/\* USER CODE END 4 \*/

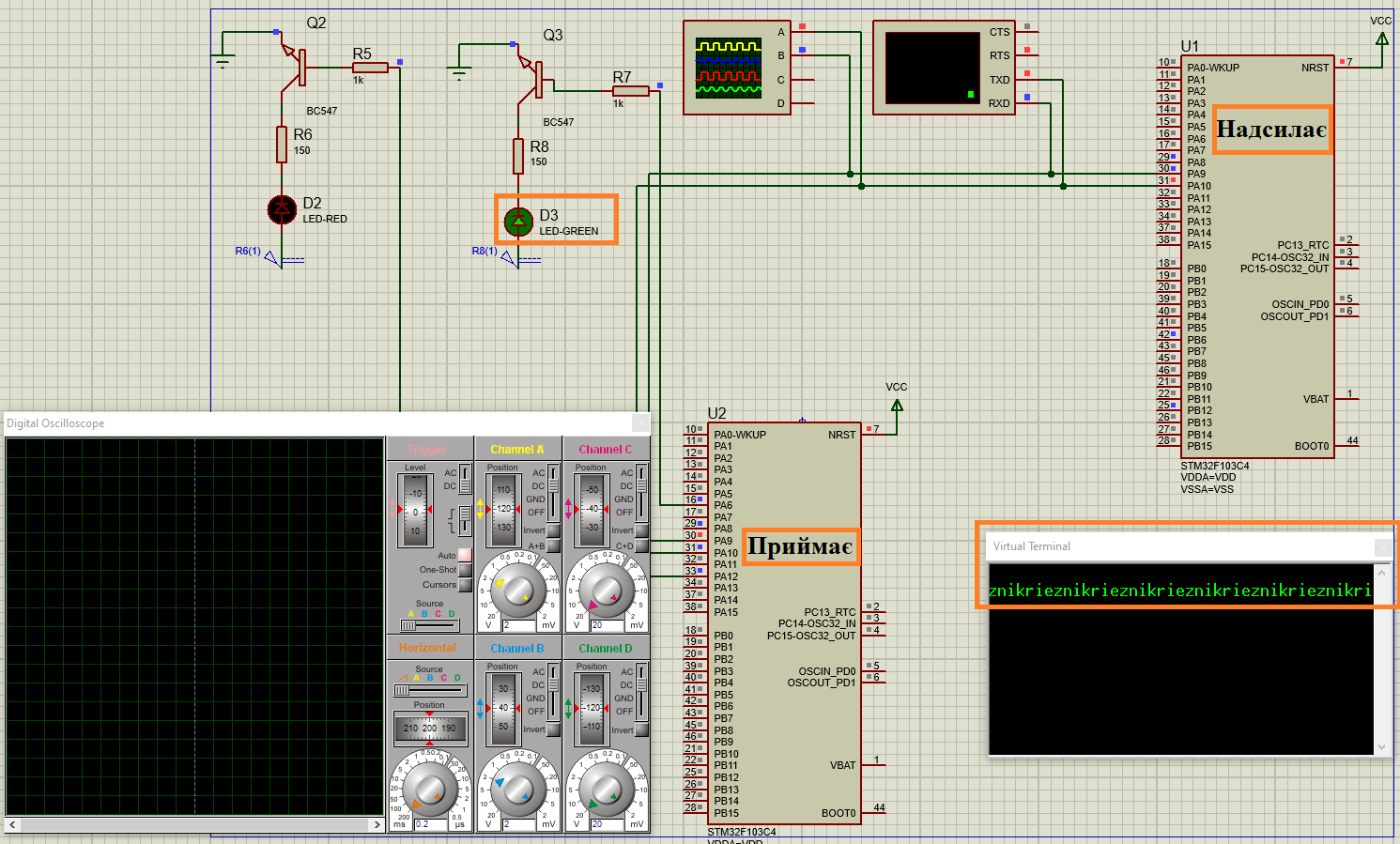


Рис. 5 Передано відповідне повідомлення

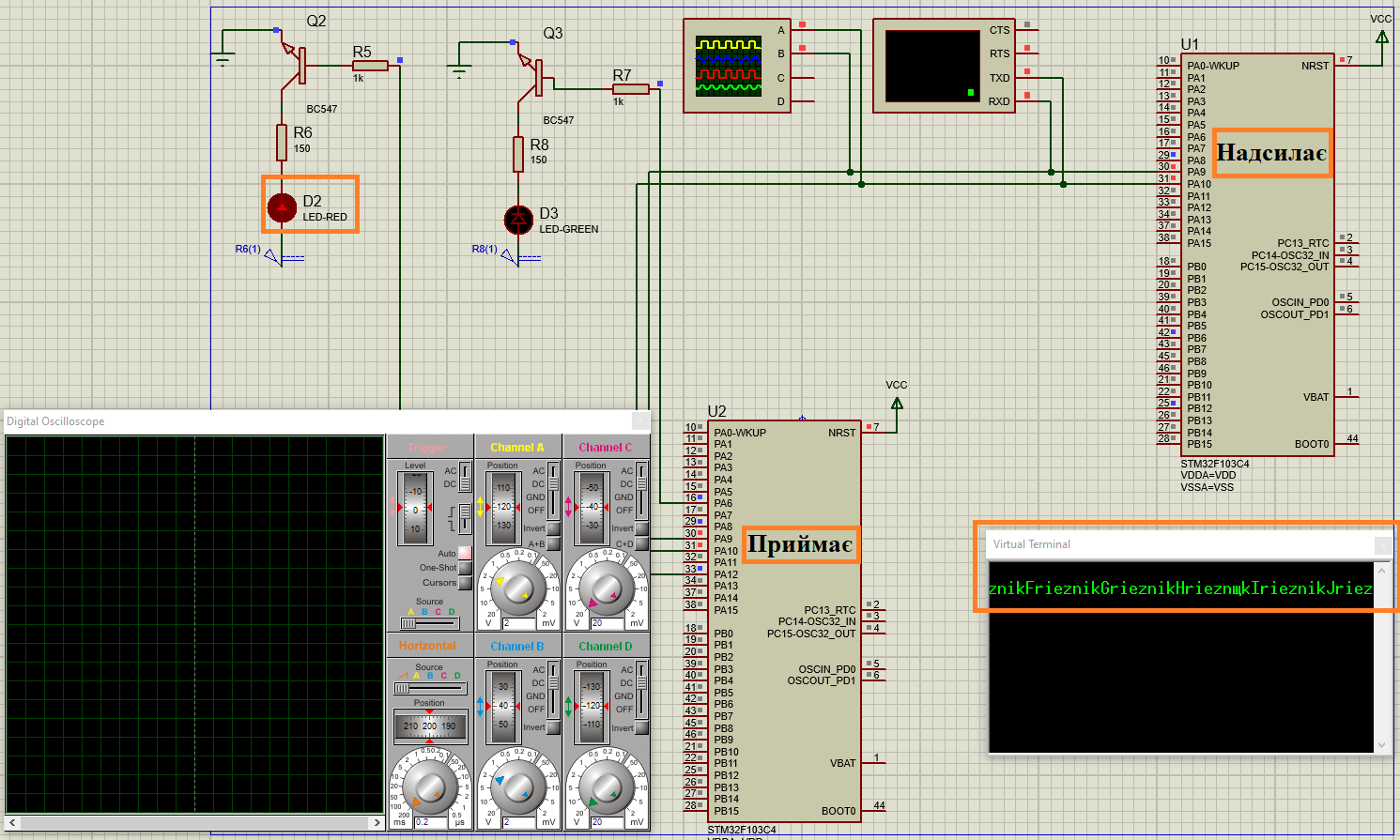


Рис. 6 Передано не відповідне повідомлення

Висновок:

У процесі виконання лабораторної роботи №2 з курсу "Мікропроцесорні системи на базі ARM процесорів" було успішно реалізовано задачу передачі та прийому даних за допомогою інтерфейсу UART у режимах DMA та Blocking/Interruption. Відповідно до завдання, налаштовано виводи порту PA для режиму "push-pull", виконано передачу повідомлень з номером варіанту та транслітерацією прізвища. Тестування у середовищі Proteus демонструє коректну роботу програми: при отриманні повідомлення, що відповідає номеру варіанту, встановлюється логічна одиниця на PA6, а при отриманні іншого повідомлення — на PA12.

Робота підтвердила важливість розуміння та вміння застосовувати різні режими роботи інтерфейсу UART для ефективної взаємодії мікропроцесорної системи з периферійними пристроями. Здобуті навички та знання будуть корисними для подальшої практичної роботи у галузі розробки та дослідження мікропроцесорних систем.