Національний технічний університет України

“ Київський політехнічний інститут ”



Лабораторна робота №2

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

Тема: “ Передача даних інтерфейсом UART ”

Варіант 2

Виконав:Студент І курсу

Групи ДС-31мп

Київ 2024 р.

***Завдання***

**Передача-прийом в режимі DMA**

1. Налаштувати виводи порту РА на вихід в режимі “push-pull” відповідно до номеру (РА2) та подвоєного (РА4) номеру варіанту.
2. Налаштувати роботу передавача UART в режимі DMA з параметрами передачі:  
   Швидкість: 2 варіант – 300 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 (РB2) та подвоєного (РB4) 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(РА2). Якщо надходить інше значення – встановити значення логічної одиниці на вихід порту РА значення якого є подвоєним до номеру варіанту(РА4).
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

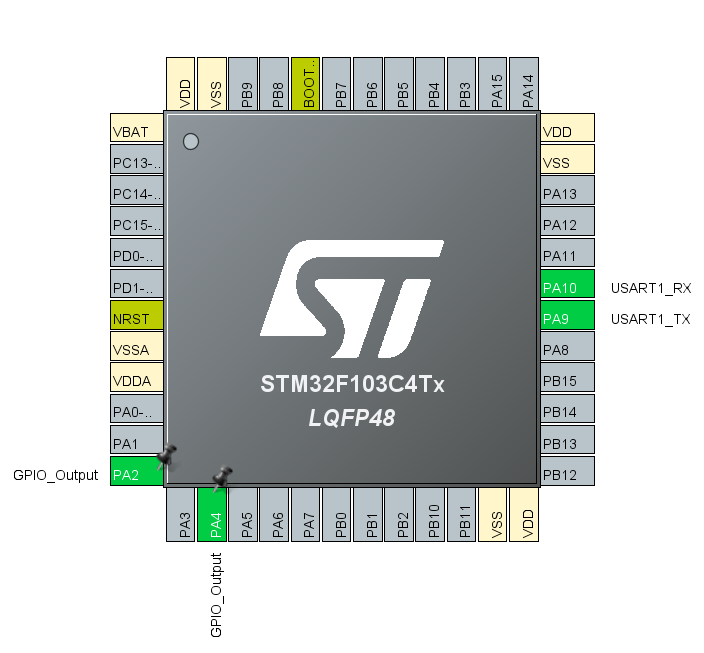


Рис. 1 Конфігурація пінів для передачі-прийому в режимі DMA

Лістинг програми:

/\* 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[0], **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] == '2') // Якщо отримано номер варіанту

{

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_2, *GPIO\_PIN\_SET*); // Встановити PA2

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_4, *GPIO\_PIN\_RESET*); // Вимкнуть PA4

}

**else**

{

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_4, *GPIO\_PIN\_SET*); // Інакше встановити PA4

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_2, *GPIO\_PIN\_RESET*); // Вимкнуть PA2

}

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

}

}

/\* USER CODE END 4 \*/

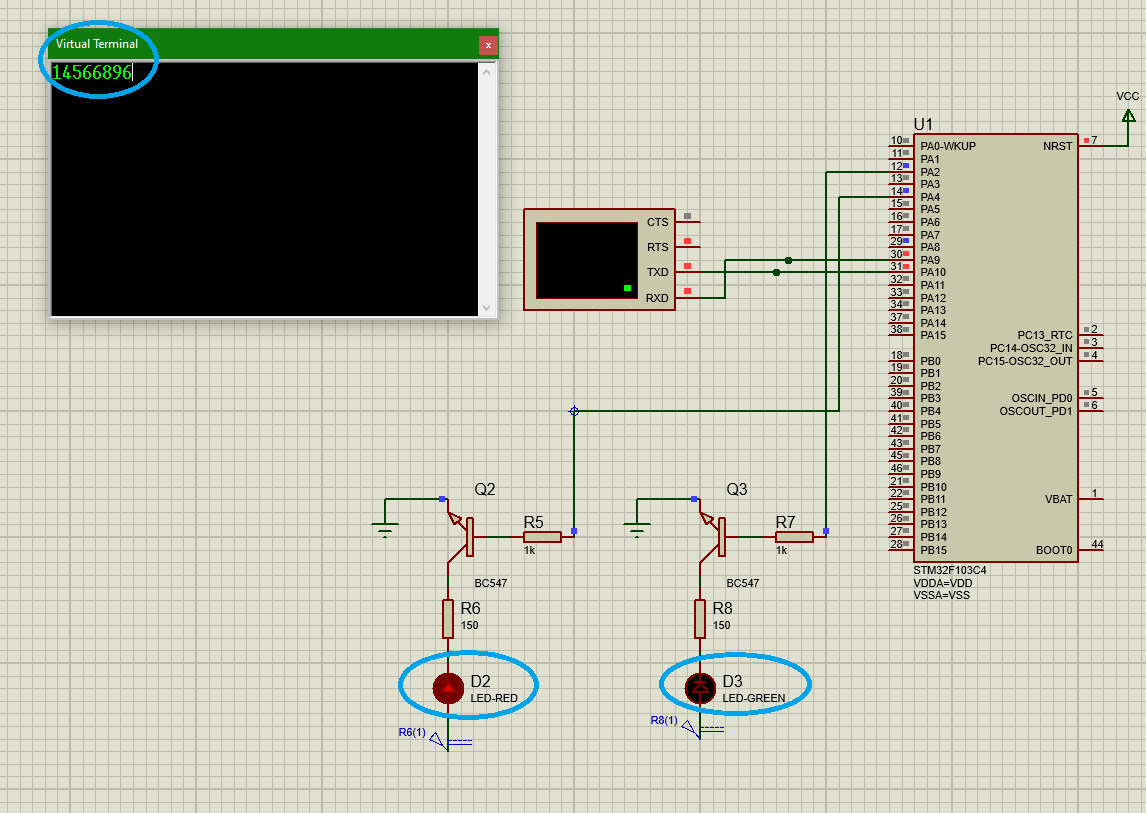


Рис. 2 Повідомлення в терміналі не відповідає номеру варіанту

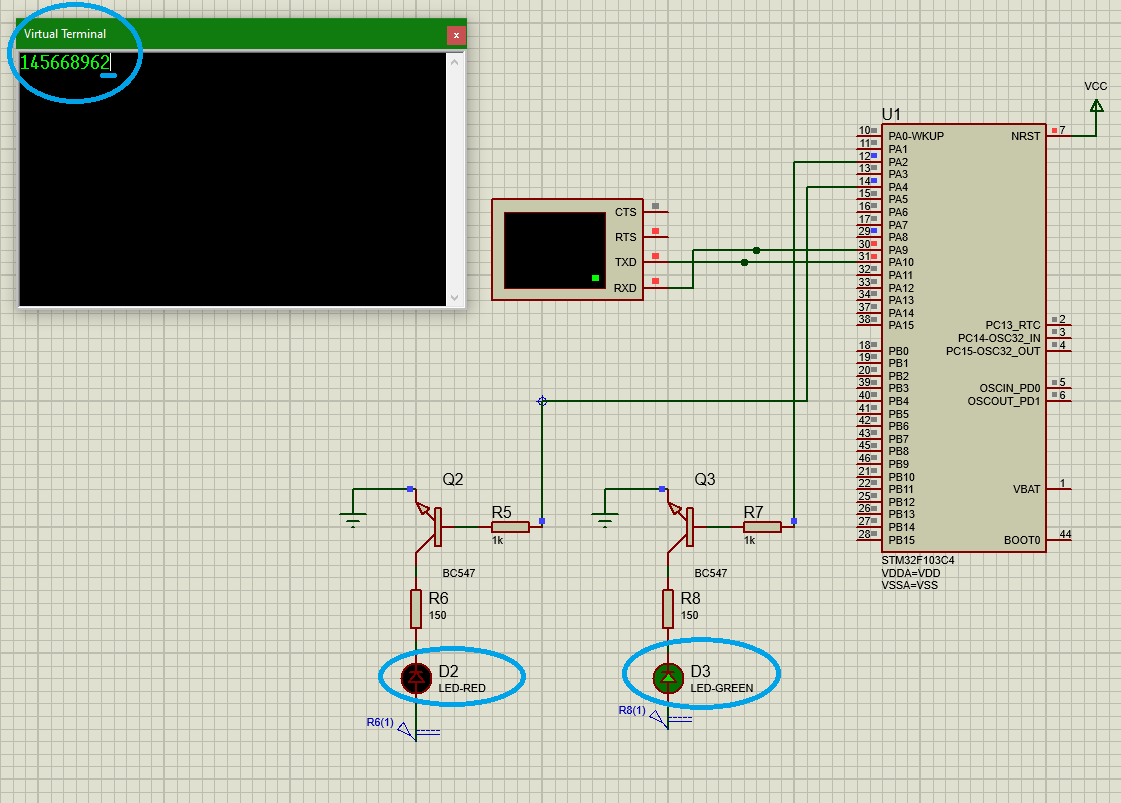


Рис. 3 Повідомлення в терміналі відповідає номеру варіанту

***Blocking***

Лістинг коду:

Частина коду приймача

**#include** <string.h>

/\* USER CODE BEGIN PV \*/

uint8\_t rx[7]; // буфер з інформацією ←

/\* USER CODE END PV \*/

/\* USER CODE BEGIN 2 \*/

HAL\_UART\_Receive\_DMA(&huart1, &rx[0], **sizeof**(rx)); // (адреса юарт інтерфейсу, адреса буфера, кількість даних в байтах)

/\* USER CODE END 2 \*/

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

{

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

{

**if** (**strncmp**((**char**\*)rx, "Hanchev", 7)) // перевірка транслітерації прізвища

{

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_2, *GPIO\_PIN\_RESET*); // Встановити PA2

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_4, *GPIO\_PIN\_SET*); // Вимкнуть PA4

}

**else**

{

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_4, *GPIO\_PIN\_RESET*); // Інакше встановити PA4

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_2, *GPIO\_PIN\_SET*); // Вимкнуть PA2

}

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

}

}

Частина коду передавача

/\* USER CODE BEGIN PV \*/

uint8\_t surname[7] = "Hanchev";

uint8\_t name[7] = "Pavlo";

/\* USER CODE END PV \*/

/\* USER CODE BEGIN WHILE \*/

**while** (1)

{

/\* USER CODE END WHILE \*/

HAL\_UART\_Transmit(&huart1, (uint8\_t\*)name, **sizeof**(name), 1000); // виведення імені в термінал

HAL\_Delay(500); // пауза на пів секунди

HAL\_UART\_Transmit(&huart1, (uint8\_t\*)surname, **sizeof**(surname), HAL\_MAX\_DELAY); // виведення прізвища

HAL\_Delay(500);

/\* USER CODE BEGIN 3 \*/

}

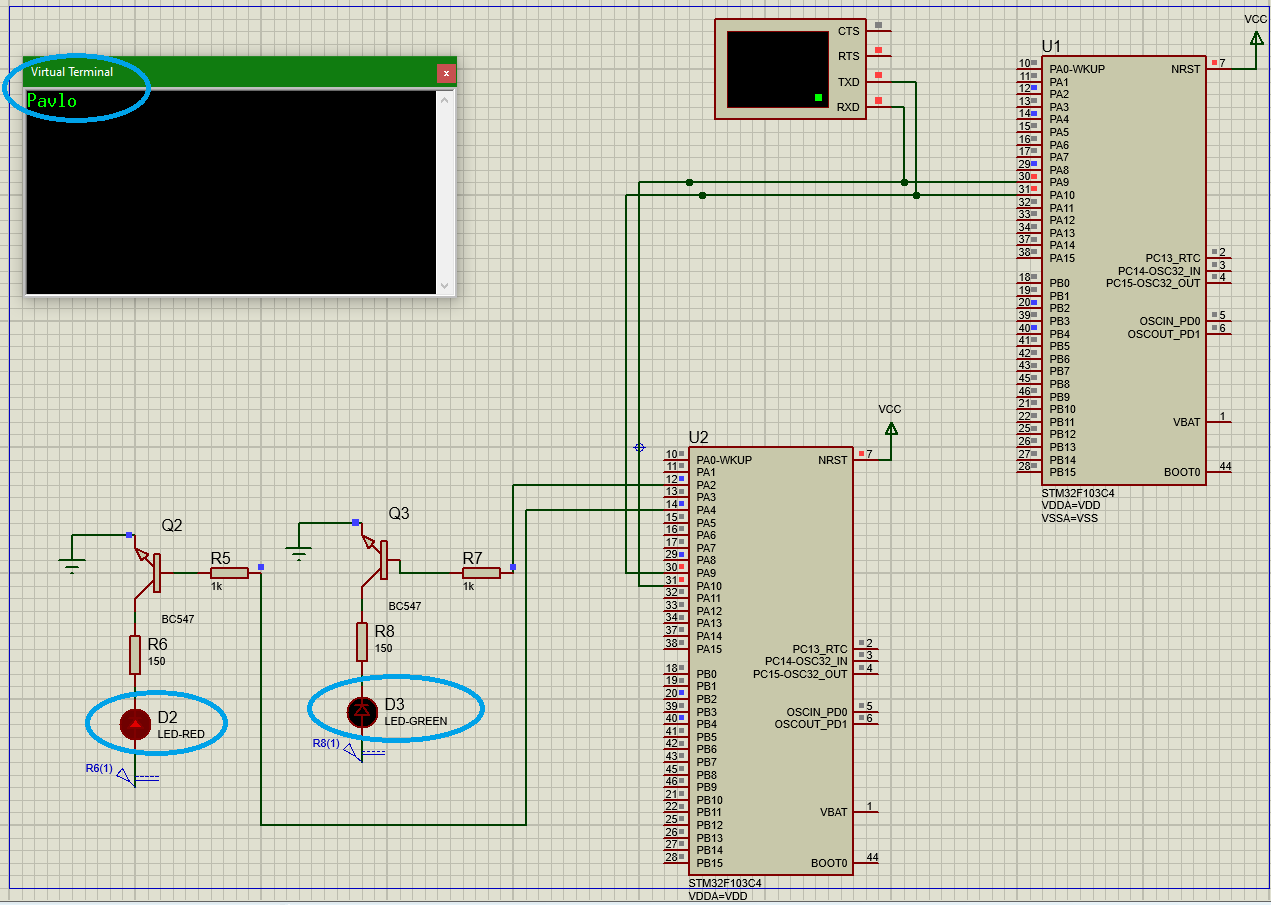


Рис. 4 Передано до терміналу ім’я

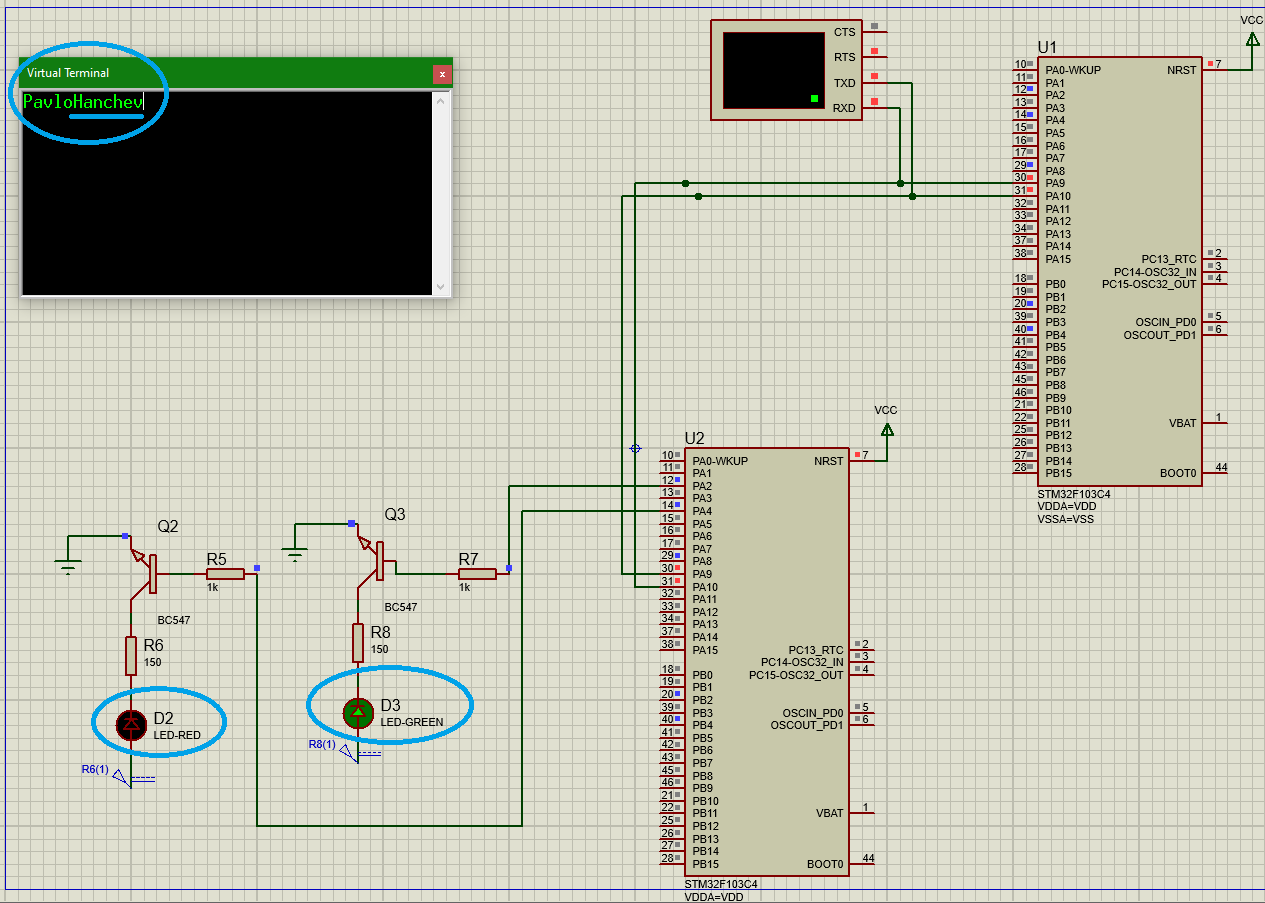


Рис. 5 Передана транслітерація прізвища

***Interrupt mode***

Лістинг коду:

Частина коду приймача

**#include** <string.h>

/\* USER CODE BEGIN PV \*/

uint8\_t rx[7]; // буфер з інформацією ←

/\* USER CODE END PV \*/

/\* USER CODE BEGIN 2 \*/

HAL\_UART\_Receive\_DMA(&huart1, &rx[0], **sizeof**(rx)); // (адреса юарт інтерфейсу, адреса буфера, кількість даних в байтах)

/\* USER CODE END 2 \*/

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

{

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

{

**if** (**strncmp**((**char**\*)rx, "Hanchev", 7)) // перевірка транслітерації прізвища

{

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_2, *GPIO\_PIN\_RESET*); // Встановити PA2

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_4, *GPIO\_PIN\_SET*); // Вимкнуть PA4

}

**else**

{

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_4, *GPIO\_PIN\_RESET*); // Інакше встановити PA4

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_2, *GPIO\_PIN\_SET*); // Вимкнуть PA2

}

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

}

}

Частина коду передавача

/\* USER CODE BEGIN 1 \*/

uint8\_t name[7] = "Pavlo";

uint8\_t surname[7] = "Hanchev";

/\* USER CODE END 1 \*/

**while** (1)

{

/\* USER CODE END WHILE \*/

**for**(**int** i=0; i<**sizeof**(name); i++)

{

HAL\_GPIO\_TogglePin(GPIOB, GPIO\_PIN\_2);

HAL\_UART\_Transmit(&huart1, &name[i], 1, 1000);

HAL\_Delay(200);

}

HAL\_Delay(500);

**for**(**int** i=0; i<**sizeof**(surname); i++)

{

HAL\_GPIO\_TogglePin(GPIOB, GPIO\_PIN\_2);

HAL\_UART\_Transmit(&huart1, &surname[i], 1, 1000);

HAL\_Delay(200);

}

HAL\_Delay(500);

/\* USER CODE BEGIN 3 \*/

}

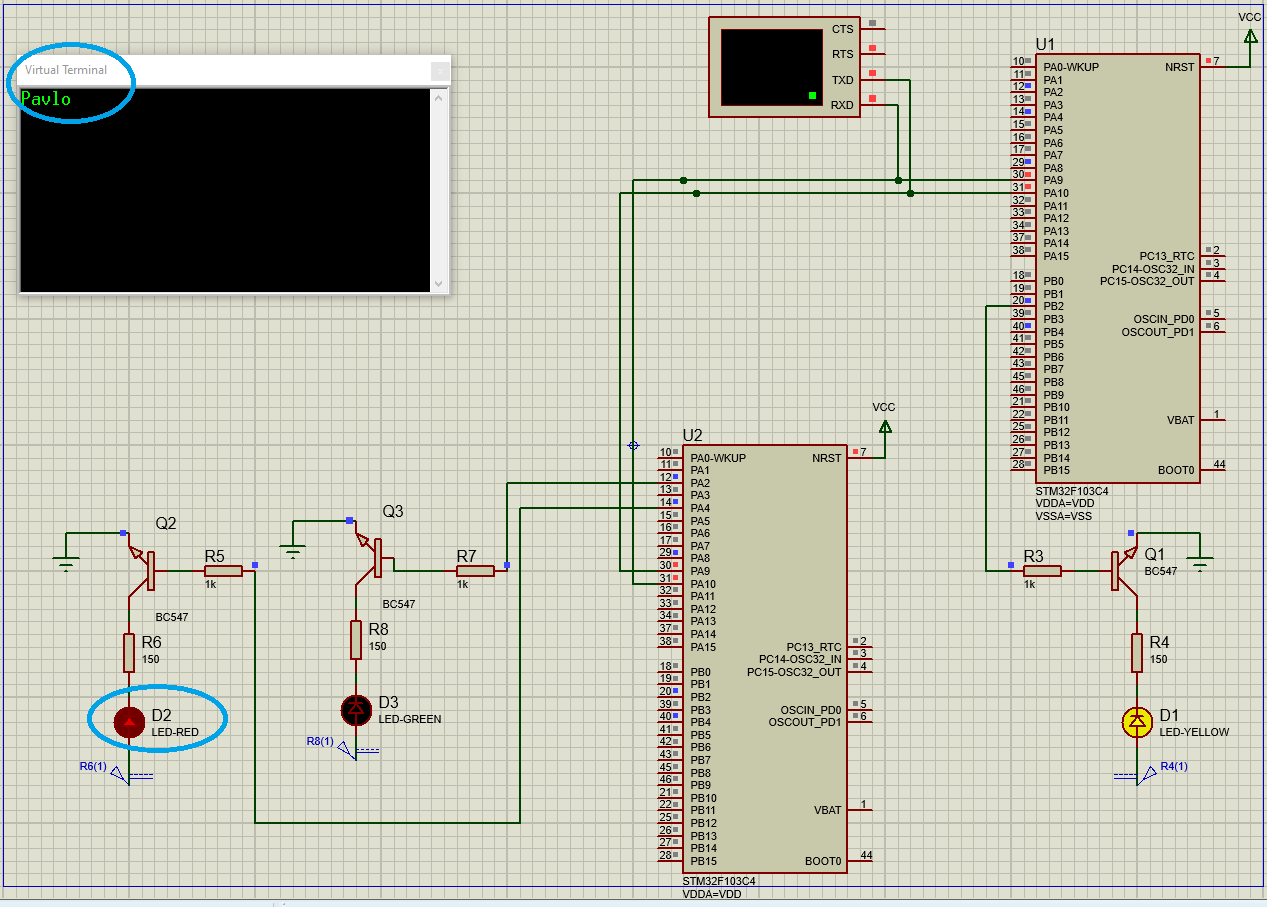


Рис. 6 Передано до терміналу ім’я

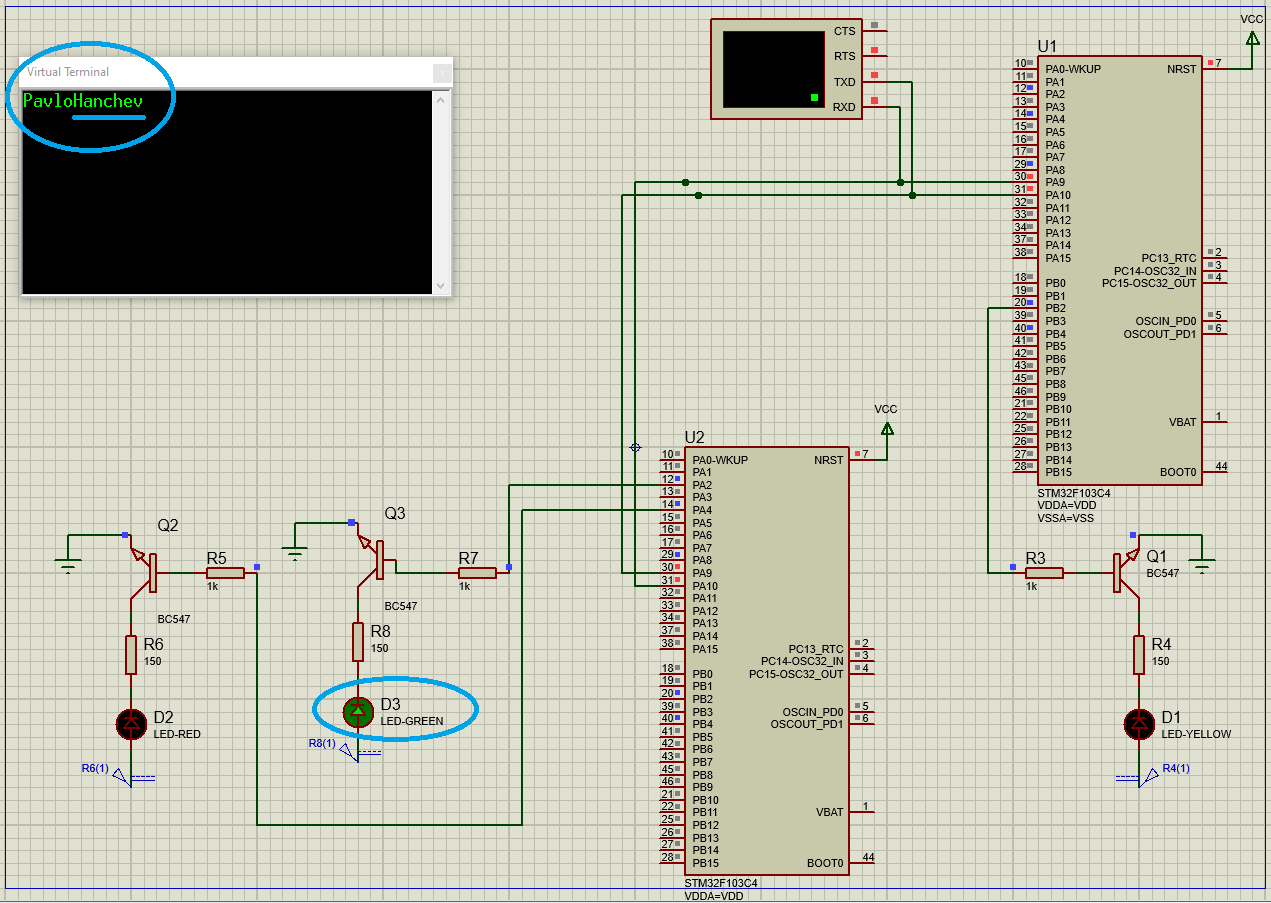


Рис. 7 Передана транслітерація прізвища

Висновок:

У ході лабораторної роботи було виконано налаштування та реалізація передачі та прийому даних за допомогою інтерфейсу UART в режимах DMA та з перериваннями. Зокрема, було налаштовано виводи порту PA на вихід в режимі "push-pull" та успішно впроваджено передачу і прийом повідомлень за заданими параметрами швидкості (300 бод), довжини повідомлення (8 біт), відсутності біта парності, та одного стоп-біта.

Програмний код, розроблений для цієї мети, ефективно обробляв прийняті повідомлення, перевіряючи їх відповідність до заданого критерію (номер варіанту або транслітерація прізвища), і відповідно встановлював логічні рівні на виводах GPIOA. Робота коду була демонстрована через використання терміналу в середовищі Proteus, де були отримані чіткі підтвердження коректної роботи системи згідно з вимогами завдання.