Національний технічний університет України

“ Київський політехнічний інститут ”



Лабораторна робота №3

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

Тема: “ Передача даних інтерфейсом SPI”

Варіант 6

Виконав:Студент І курсу

Групи ДС-31мп

Рєзнік Олександр Сергійович

Київ 2024 р.

**I. Передача-прийом в блокуючому режимі**

Дані пункти роботи виконуються для мікроконтролера STM32F103C6. Тому при виборі пристрою, для якого створюється програмне забезпечення в момент генерації проекту в середовищі STM32CubeIDE слід вибрати саме цей мікроконтролер.

1. Завантажити модель для симуляції в програмному середовищі Proteus з матеріалів прикладених до лабораторної роботи.

2. Налаштувати роботу першого мікроконтролера в режимі “Full-duplex master” для роботи в режимі передавача, параметри сигналу наступні: Формат повідомення:

Motorola;

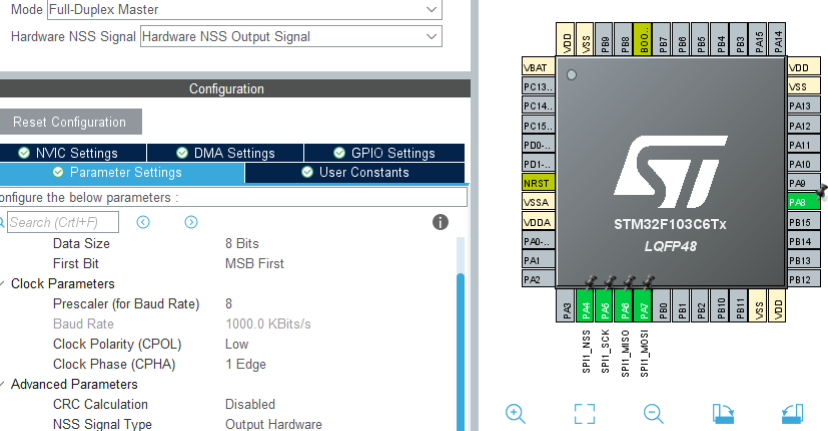
Довжина посилки: 8 біт;

Режим передачі: Старшим бітом вперед (MSB First);

CPOL=0 (Low);

CPHA=0 (1st Edge);

Hardware output SS signal.



3. Налаштувати роботу другого мікроконтролера в режимі “Full-duplex slave” або “Receive only slave”, для роботи в режимі приймача, з параметрами сигналу Формат повідомення:

Motorola;

Довжина посилки: 8 біт;

Режим передачі: Старшим бітом вперед (MSB First);

CPOL=0 (Low);

CPHA=0 (1st Edge);

Hardware input SS signal.

4. У ньому ж налаштувати передачу інформації інтерфейсом UART з параметрами сигналу:

Швидкість: 9600 бод;

Довжина повідомлення – 8 біт;

Біт парності – відсутній;

Стоп-біт – 1.

5. Створити програмний код, що, на боці передавача передаватиме інтефейсом SPI першу літеру з транслітерації Вашого прізвища англійською мовою. Після її прийому на боці приймача, вона має бути відправлена інтерфейсом UART до зовнішнього терміналу підключеного в моделі для відлагоджування в середовищі Proteus.

**II. Передача в блокуючому режимі-прийом в режимі з генерацією переривання Ці пункти роботи виконуються на звичному мікроконтролері STM32F103C4**

6. Налаштувати роботу першого мікроконтролера в режимі “Full-duplex master” для роботи у якості передавача, з параметрами повідомлення аналогічними п. 2 даної роботи.

7. Налаштувати роботу другого мікроконтролера в режимі роботи “Full-duplex slave” або “Receive only slave” з генерацією переривання при прийомі даних (при конфігурації підключити переривання «SPI1 global interrupt “ в підменю “NVIC Settings”). Параметри сигналу аналогічні п.3 даної роботи.

8. Налаштувати виводи порту РА того ж мікроконтролера на вихід в режимі “push-pull” відповідно до номеру та подвоєного номеру варіанту.

9. Створити програмний код, що на боці передавача циклічно передаватиме цілі числа від 0 до 99. На боці приймача створити програмний код, що за прийому парного числа підключатиме периферію підключену до виводу порту РА за номером варіанту, у іншому випадку – периферію підключену до виводу порту РА з подвоєним номером варіанту.

**I. Передача-прийом в блокуючому режимі**

Лістинг коду:

Частина коду передавача

**#include** "main.h"

uint8\_t dataToSend[1] = {"R"};

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

{

/\* USER CODE BEGIN 2 \*/

HAL\_Delay(1000);

/\* USER CODE END 2 \*/

/\* Infinite loop \*/

/\* USER CODE BEGIN WHILE \*/

**while** (1)

{

/\* USER CODE END WHILE \*/

HAL\_SPI\_Transmit(&hspi1, &dataToSend[0], 1, HAL\_MAX\_DELAY);

HAL\_Delay(100); // Відправляємо дані кожну секунду HAL\_MAX\_DELAY

/\* USER CODE BEGIN 3 \*/

}

/\* USER CODE END 3 \*/

}

Частина коду приймача

**#include** "main.h"

**#include** <string.h>

uint8\_t rD[1] = "R";

uint8\_t received\_data[1];

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

{

/\* USER CODE BEGIN 2 \*/

HAL\_Delay(1000);

/\* USER CODE END 2 \*/

/\* Infinite loop \*/

/\* USER CODE BEGIN WHILE \*/

**while** (1)

{

/\* USER CODE END WHILE \*/

HAL\_SPI\_Receive(&hspi1, &receivedData[0], 1, 100);

//HAL\_Delay(1000);

**if** (rD[0] == receivedData[0]){

HAL\_UART\_Transmit(&huart1, &receivedData[0], 1, 100);

} // Відправляємо отримані дані через UART

HAL\_Delay(100);

/\* USER CODE BEGIN 3 \*/

}

/\* USER CODE END 3 \*/

}

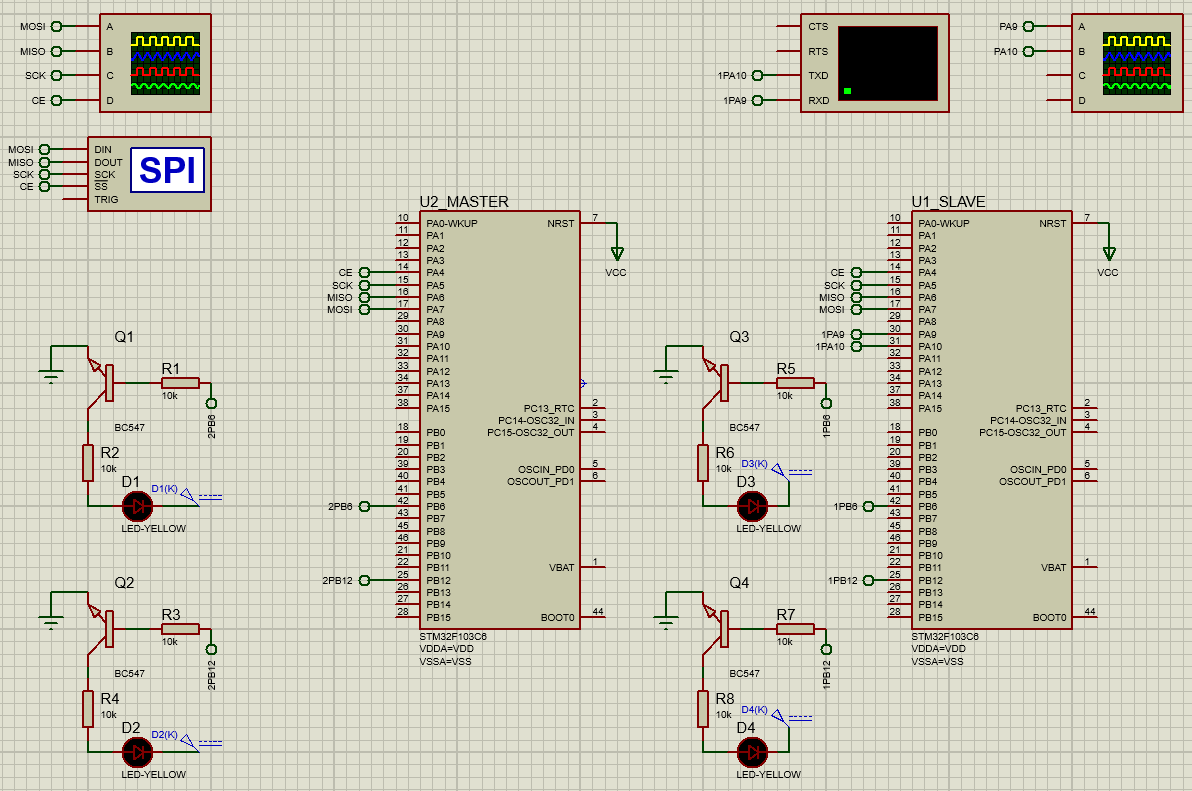


Рис. 1 Загальна схема роботи

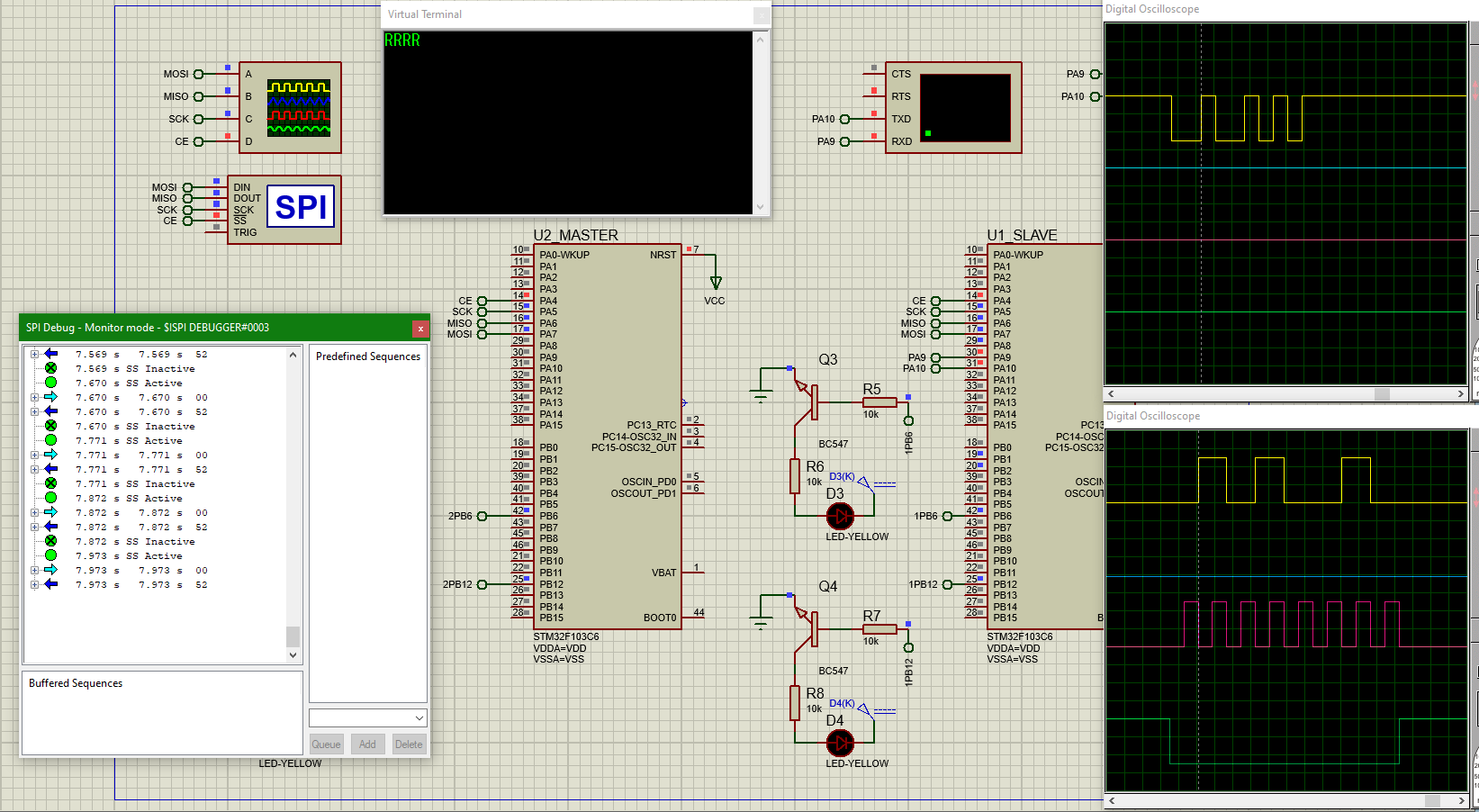


Рис. 2 Отримання символу «R» на терміналі, після передачі SPI I UART

**II. Передача в блокуючому режимі-прийом в режимі з генерацією переривання**

Лістинг коду:

Частина коду передавача

**#include** "main.h"

/\* USER CODE BEGIN 0 \*/

uint8\_t data = 0;

/\* USER CODE END 0 \*/

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

{

/\* Infinite loop \*/

/\* USER CODE BEGIN WHILE \*/

**while** (1)

{

/\* USER CODE END WHILE \*/

HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_6, *GPIO\_PIN\_RESET*);

HAL\_SPI\_Transmit(&hspi1, &data, 1, HAL\_MAX\_DELAY);

data = (data + 1) % 100; // Збільшуємо число до 99

HAL\_Delay(100); // Відправляємо дані кожну секунду HAL\_MAX\_DELAY

HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_6, *GPIO\_PIN\_SET*);

HAL\_Delay(100);

/\* USER CODE BEGIN 3 \*/

}

/\* USER CODE END 3 \*/

}

Частина коду периймача

**#include** "main.h"

/\* USER CODE BEGIN 0 \*/

uint8\_t receivedData[1];

/\* USER CODE END 0 \*/

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

{

/\* USER CODE BEGIN 2 \*/

HAL\_Delay(1000);

/\* USER CODE END 2 \*/

/\* Infinite loop \*/

/\* USER CODE BEGIN WHILE \*/

**while** (1)

{

/\* USER CODE END WHILE \*/

/\* USER CODE BEGIN 3 \*/

HAL\_SPI\_Receive(&hspi1, receivedData, 1, 1000);

**if** ((**int**)receivedData % 2 == 0) {

// Підключення периферії до виводу порту PA з номером варіанту

HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_6, *GPIO\_PIN\_SET*);

HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_12, *GPIO\_PIN\_RESET*);

} **else** {

// Підключення периферії до виводу порту PA з подвоєним номером варіанту

HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_6, *GPIO\_PIN\_RESET*);

HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_12, *GPIO\_PIN\_SET*);

}

HAL\_UART\_Transmit(&huart1, receivedData, 1, 1000);

HAL\_Delay(100);

}

/\* USER CODE END 3 \*/

}

***Висновок:***

Виконані експерименти підтвердили правильність налаштувань мікроконтролерів та працездатність програмного коду.

Реалізація передачі та прийому даних через SPI інтерфейс продемонструвала ефективність використання мікроконтролерів STM32 для таких задач.

Було налаштовано генерацію переривання при прийомі даних. Реалізовано передачу чисел від 0 до 99 з передавача до приймача. На боці приймача здійснено перевірку парності отриманого числа та підключення периферії відповідно до результату перевірки.

Отримані навички налаштування та програмування інтерфейсу SPI, а також роботи з UART для відлагодження, є важливими для подальшого використання в проектах з мікроконтролерами.