Міністерство освіти і науки України

Харківський національний університет ім. В. Н. Каразіна

Факультет комп’ютерних наук

**КУРСОВА РОБОТА**

з дисципліни «Комп’ютерна схемотехніка та архітектура комп’ютерів»

Тема «Периферія USART. Робота через переривання і DMA»

|  |  |
| --- | --- |
| Оцінка \_\_\_\_\_ / \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Члени комісії:  \_\_\_\_\_\_\_\_\_\_\_ Рало О.М.  \_\_\_\_\_\_\_\_\_\_\_Осипчук А.В.  \_\_\_\_\_\_\_\_\_\_\_ Стервоєдов М. Г. | Виконав:  студент 3 курсу, групи КС-32 |
| Спеціальності:  122 «Комп’ютерні науки»  ЯСмирнов Вадим |
|  |

Харків – 2020

ЗМІСТ

[ВСТУП 3](#_Toc58773894)

[1 ПОСТАНОВКА ЗАДАЧІ 5](#_Toc58773895)

[2 ЗНАЙОМСТВО З USART 6](#_Toc58773896)

[2.1 Що таке USART? 6](#_Toc58773897)

[2.2 Функціональний опис USART 6](#_Toc58773898)

[2.3 Використання DMA 9](#_Toc58773899)

[2.4 Переривання USART 11](#_Toc58773900)

[3 ЗАСТОСУВАННЯ НА ПРАКТИЦІ 13](#_Toc58773901)

[ВИСНОВКИ 16](#_Toc58773902)

[ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ 18](#_Toc58773903)

# ВСТУП

USART - це модуль універсального синхронно-асинхронного приймально-передавача. Всього в контролерах stm32 може бути до 3-х модулів USART (USART1, USART2, USART3) в залежності від конкретної моделі.

Коли відправляємо дані по USART'у, то беремо дані з якогось нашого масиву, і послідовно перекладаємо їх в регістр (USART\_DR), а з нього вони відлітають по проводках. При прийомі все відбувається в зворотному порядку - байти прилітають в приймальний регістр (він так само називається USART\_DR, але для прийому і відправки вони різні), а ми зобов'язані швиденько їх звідти забирати і кудись зберігати (якщо забрати не встигли, то отримаємо помилку). Все ці дії, запис в регістр, читання, перекладання з масиву в регістр і назад, виробляються за допомогою ЦПУ. Тобто ЦПУ витрачає свій час на ці операції і само собою програма нічого іншого в цей час робити не може. Для цього нам і потрібен DMA.

DMA (Direct Memory Access) - технологія прямого доступу до пам'яті. Ця технологія дозволяє швидко і без використання центрального процесора пересилати дані з однієї області пам'яті в іншу. При цьому для такої пересилки замість ЦП використовується свій спеціальний контролер, який називається контролером DMA. У мікроконтролери stm32 може бути вбудовано до 2-х контролерів DMA - DMA1 і DMA2.

Тобто ми можемо сказати DMA, - «ось масив байт в ОЗУ, візьми їх і відправ по USART'у». Або навпаки, - «сиди і чекай коли прийдуть дані по USART'у, і склади їх в такий-то масив». А тепер найголовніше - вся ця робота буде відбуватися без будь-якого участі ЦПУ, все буде робити саме DMA. Тобто, ми даємо команду DMA, і можемо далі виконувати код з програми. Контролер DMA працює незалежно від ЦП і паралельно з ним. Таким чином використання DMA дозволяє економити ресурси ЦП, який під час операцій пересилки даних за допомогою DMA може займатися якимись іншими корисними справами.

Незважаючи на те, що ЦП і DMA працюють незалежно один від одного, DMA може призупиняти доступ ЦП до системної шини на кілька тактів у випадках коли вони обидва намагаються звернутися до одним і тим же адресами пам'яті.

# 1 ПОСТАНОВКА ЗАДАЧІ

**Постановка задачі:** ознайомитися з універсальним синхронним/ асинхронним приймачем та його функціональністю.

**Об’єкт дослідження**: USART.

**Предмет дослідження**: методи та засоби використання USART на практиці та роботи з перериваннями та DMA.

**Мета:** покращення навичок роботи з мікроконтролером сімейства STM32 за допомогою універсального синхронного/ асинхронного приймача та його роботи з перериваннями та DMA.

1. **ЗНАЙОМСТВО З USART**

## **2.1 Що таке USART?**

Універсальний синхронний/асинхронний трансивер (universal synchronous asynchronous receiver transmitter, USART) надає гнучкий спосіб полнодуплексного обміну з зовнішнім обладнанням, що вимагають промислові стандарти послідовної асинхронної передачі даних в форматі NRZ. USART можна налаштувати на швидкість обміну в широкому діапазоні завдяки використання дрібного генератора швидкості (fractional baud rate generator).

Підтримується односпрямований обмін у будь-яку сторону і напівдуплексний обмін в обидві сторони по одному дроту (half-duplex single wire). Також підтримуються протоколи LIN (local interconnection network), Smartcard Protocol і IrDA (infrared data association) специфікацій SIR ENDEC і операції модему (CTS / RTS). Це дозволяє реалізувати мультипроцесорний обмін даними. Висока швидкість обміну підтримується конфігурацією DMA для декількох буферів.

* 1. **Функціональний опис USART**

Інтерфейс зазвичай підключається до зовнішніх систем трьома ніжками (рис. 1). Будь-двонаправлений обмін USART вимагає як мінімум двох сигнальних висновків: вхідні дані, що приймаються (Receive Data In, RX) і вихідні дані, що передаються (Transmit Data Out, TX):

* RX: вхід послідовних отриманих даних. Використовуються техніки передискретизации для відновлення даних, щоб відокремити корисні приходять дані від шуму.
* TX: вихід даних, що передаються. Коли передавач заборонений, ніжка виходу поверне свою конфігурацію порту введення / виведення (GPIO). Коли передавач дозволений і нічого не передається, рівень виходу ніжки TX знаходиться в лог. 1. У режимах single-wire та smartcard, цей висновок I / O використовується і для передачі, і для прийому даних (на рівні USART дані потім приймаються на SW\_RX).

Через ці висновки послідовні дані приймаються і передаються в нормальному режимі USART як фрейми. У цьому процесі використовується наступне:

* Стан очікування лінії (Idle Line) до передачі або прийому.
* Start-біт.
* Слово даних (8 або 9 біт), наймолодший біт слова (LSB) йде першим.
* 0.5,1, 1.5, 2 Stop-стоп-біт, що показують завершення фрейма.
* Використовується дробовий генератор швидкості - з 12-розрядної мантиси і 4-бітної дробової частиною.
* Регістр статусу (USART\_SR).
* Регістр даних (USART\_DR).
* Регістр генератора швидкості (USART\_BRR) з 12-розрядної мантиси і 4-бітної дробової частиною.
* Регістр захисного часу, Guardtime Register (USART\_GTPR) в разі використання режиму Smartcard.

Для роботи в синхронному режимі потрібен додатковий висновок тактів:

* CK: вихід тактів передавача. На цей висновок видаються такти даних для синхронної передачі, відповідної режиму SPI master (немає тактовихімпульсів на бітах start і stop, і програмно обрана опція відправки тактового імпульсу на останньому бите даних). Паралельно і синхронно можуть прийматися дані через ніжку RX. Це можна використовувати для управління зовнішніми периферійними пристроями, у яких є регістри зсуву (наприклад драйвери LCD). Фаза і полярність тактів вибирається програмно. У режимі smartcard CK може надавати такти для смарт-карти.

У режимі апаратного управління потоком потрібні ще 2 висновки:

1. CTS: сигнал Clear To Send, блокуючий передачу даних по закінченні поточної передачі (коли CTS = 1).
2. RTS: сигнал Request To Send, що показує, що USART готовий приймати дані (коли RTS = 0).

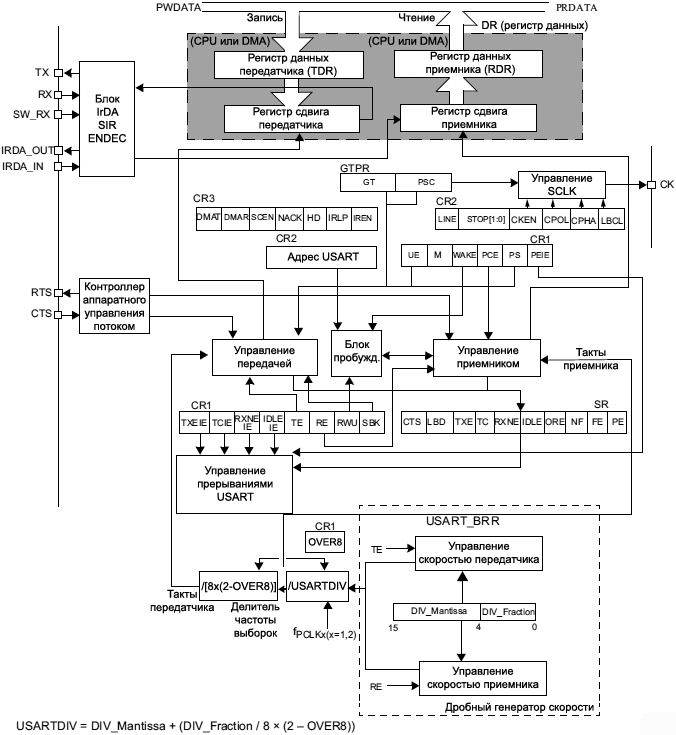


Рисунок 1 – Блок-схема USART

## **2.3 Використання DMA**

USART може реалізувати безперервний обмін за допомогою DMA. Запити DMA для буферів прийому і передачі генеруються незалежно.

Передача з використанням DMA. Режим DMA може бути дозволений для передачі шляхом установки біта DMAT в регістрі USART\_CR3. Дані завантажуються з області SRAM, сконфигурированной з використанням периферійного пристрою DMA для регістра USART\_DR, кожен раз, коли встановлений біт TXE. Для відображення каналу DMA на передачу USART використовуйте наступну процедуру (x позначає номер каналу DMA):

1. Запишіть адресу регістра USART\_DR в регістр управління DMA, щоб настроїти його як місце призначення передачі. Дані будуть переміщатися за цією адресою з пам'яті після кожної події TXE.
2. Запишіть адресу пам'яті в регістр управління DMA, щоб настроїти його як джерело передачі. Ці дані будуть завантажуватися в регістр USART\_DR з цієї пам'яті після кожної події TXE.
3. Налаштуйте загальна кількість переданих байт в регістр управління DMA.
4. Налаштуйте пріоритет каналу в регістрі DMA.
5. Налаштуйте генерацію переривання після половини/повної передачі, як цього вимагає додаток.
6. Очистіть біт TC в регістрі SR записом туди 0.
7. Активуйте канал в регістрі DMA.

Коли досягнуто запрограмоване в контролер DMA кількість переданих байт, контролер DMA згенерує переривання по вектору каналу DMA.

У режимі передачі як тільки DMA записав всі дані для передачі (в регістрі DMA\_ISR встановився прапор TCIF), прапор TC може відслідковуватися, щоб упевнитися в завершенні обміну USART. Це необхідно перед забороною USART або входом в режим Stop, щоб не ушкодилася остання передача. Програма повинна почекати, щоб TC перейшов в 1. Прапор TC залишається очищеним під час усіх передач даних, і встановиться апаратно після закінчення передачі останнього кадру.

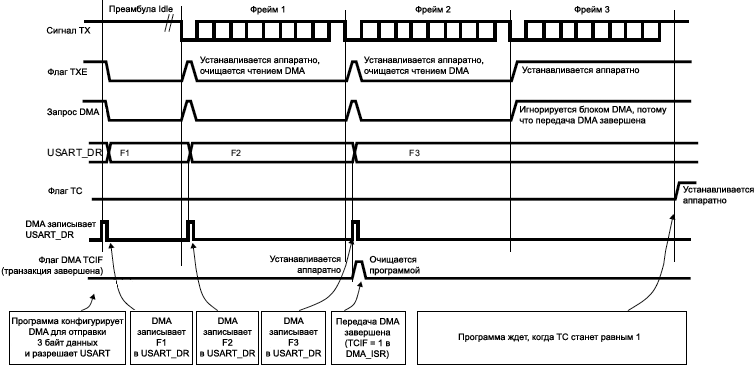


Рисунок 2 – Передача з використанням DMA

Прийом з використанням DMA. Режим DMA може бути дозволений для прийому шляхом установки біта DMAR в регістрі USART\_CR3. Дані завантажуються з регістра USART\_DR в область SRAM, налаштований з використанням периферійного пристрою DMA, всякий раз, коли приймається байт даних. Щоб відобразити канал DMA для прийому USART, використовуйте наступну процедуру:

1. Запишіть адресу регістра USART\_DR в регістр управління DMA, щоб настроїти його як джерело прийому. Дані будуть переміщатися з цієї адреси пам'яті після кожної події RXNE.
2. Запишіть адресу пам'яті в регістр управління DMA, щоб настроїти його як місце призначення прийому. Ці дані будуть завантажуватися з регістра USART\_DR в цю пам'ять після кожного події RXNE.
3. Налаштуйте загальна кількість прийнятих байт в регістр управління DMA.
4. Налаштуйте пріоритет каналу в регістрі DMA.
5. Налаштуйте генерацію переривання після половини/повної передачі, як цього вимагає додаток.
6. Увімкніть канал в регістрі DMA.

Коли досягнуто кількість даних, запрограмоване в контролері DMA, контролер DMA генерує переривання на векторі переривання каналу DMA. Біт DMAR в регістрі USART\_CR3 повинен бути очищений програмою в обробнику переривання.

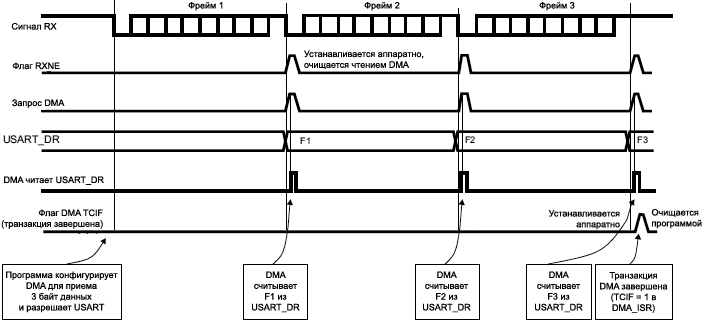


Рисунок 3 – Прийом з використанням DMA

## **2.4 Переривання USART**

Події переривань USART під'єднані до одного і тим самим вектором переривання (рис. 4).

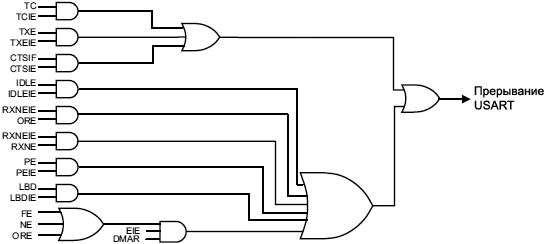


Рисунок 4 – Відображення подій USART на переривання

Під час передачі можуть бути переривання: Transmission Complete (передача завершена), Clear to Send (лінія чиста для початку передачі) або Transmit Data Register empty (реєстр даних передачі порожній).

Під час прийому можуть бути переривання: Idle Line detection (визначено стан очікування лінії), Overrun error (помилка переповнення, програма не встигла прочитати прийняті дані), Receive Data register not empty (реєстр даних прийому не порожній), Parity error (помилка перевірки парності ), LIN break detection (визначення сигналу припинення передачі LIN), Noise Flag (прапор шуму, тільки при багатобуферний обміні) і Framing Error (помилка фрейма, тільки при багатобуферний обміні).

Всі ці події генерують переривання, якщо встановлений відповідний біт дозволу переривання.

# 3 ЗАСТОСУВАННЯ НА ПРАКТИЦІ

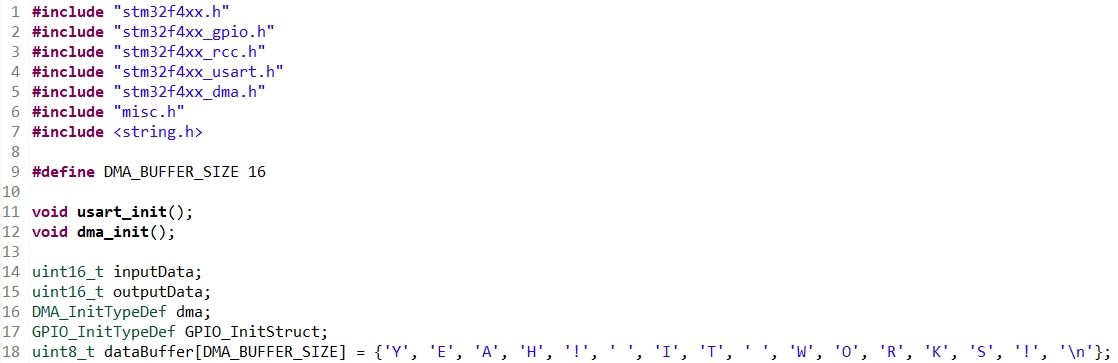


Рисунок 5 – Ініціалізація змінних та функцій, підключення бібліотек

Спочатку підключаємо необхідні бібліотеки. Далі ініціалізуємо змінні та функції.

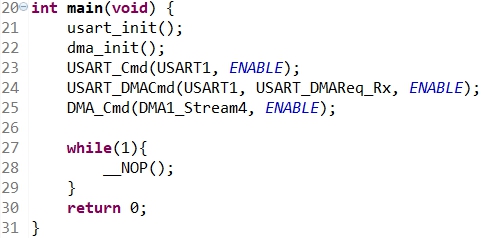


Рисунок 6 – Головна функція main для виконання програми

В функції main відбувається ініціалізація USART- та DMA-портів, ввімкнення USART-порту, підключення USART до DMA, та ввімкнення DMA-потоку. Функцію \_\_NOP() можна використати, адже маємо невеликі затримки.

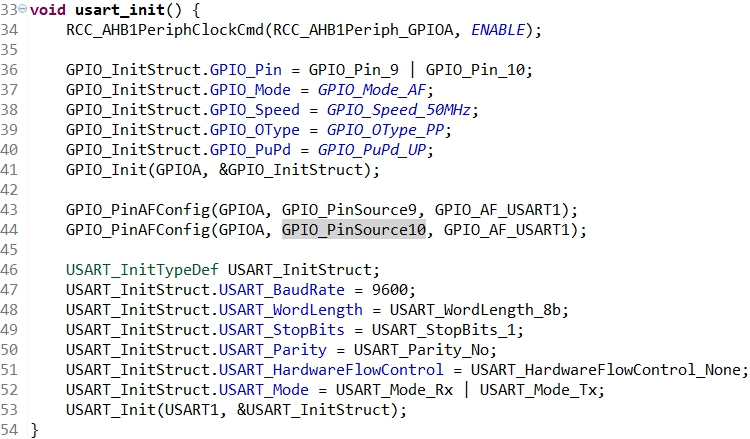


Рисунок 7 – Функція для ініціалізації порту USART.

В функції usart\_init підключаємося до периферії GPIO на матриці AHB1, на якій підключаємо 2 піни, підключаємо до них альтернативну функцію для того, щоб над ними працював не процесор, а USART, встановлюємо частоту сигналів. Далі налаштовуємо USART, а саме: встановлюємо швидкість зчитування та запису інформації, розмір одного символу, стоп-біт, вимикаємо перевірку на парність та контроль апаратної системи, встановлюємо режим передачі та отримання даних, ініціюємо налаштування.

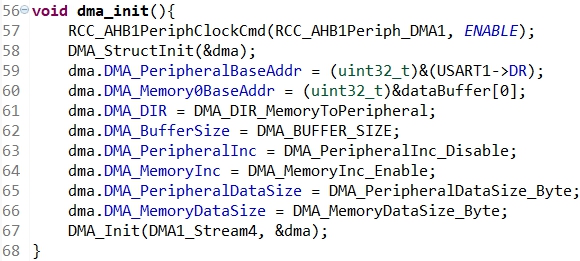


Рисунок 8 – Функція для ініціалізації DMA.

В функції dma\_init підключаємо периферію DMA на матриці AHB1, встановлюємо адресу регістра USART\_DR в регістр управління DMA, щоб настроїти його як джерело прийому, адресу пам'яті в регістр управління DMA, щоб настроїти його як джерело передачі режим запису з пам’яті в периферію, встановлюємо розмір інформації, яку потрібно передати, збільшення лічильника периферії вимикаємо, вмикаємо збільшення лічильника пам’яті, встановлюємо розмір даних на периферії та в пам’яті на 1 байт. Ініціюємо налаштування на потоці DMA.

# ВИСНОВКИ

Отже, в даній курсовій роботі ми ознайомилися з універсальним синхронним/асинхронним приймачем та його функціональністю, дізналися, для чого і коли використовують його, про роботу з перериваннями та DMA. Також була розроблена програма, яка демонструє роботу USART та DMA.

Можна зробити висновок про те, що USART надає можливість гнучкого налаштування конфігурації: можна в широкому діапазоні змінювати швидкість передачі; змінювати кількість бітів в переданому одним фреймом слові; використовувати один з декількох можливих варіантів контролю парності або відключати контроль; налаштовувати кількість стоп-бітів; можна використовувати або не використовувати лінії CTS / RTS. Безумовно, багатство можливостей по налаштуванню надає USART велику гнучкість і універсальність, але з іншого боку, створює певні труднощі при обміні даними. Адже для того, щоб обмін даними був можливим, підключені між собою пристрої повинні мати однакові настройки за всіма параметрами.

Після конфігурації USART можна приступати до обміну даними. У мікроконтролера є два варіанти роботи з USART: побайтное передача і прийом або з використанням DMA. У першому випадку, для прийому і передачі кожного байта процесор повинен виконати послідовність певних дій. Так, він може обробляти переривання, що виникають, коли спустошується регістр переданих даних або заповнюється регістр даних, що приймаються; встановивши причину переривання, записати в регістр даних черговий байт для передачі або вважати з регістра прийнятий байт і обробити його. Якщо використовується режим роботи з DMA, то процесору необхідно задати розташування буферів для прийому і передачі, їх розмір. USART і DMA далі будуть здійснювати передачу і прийом самостійно, не вимагаючи витрат обчислювальних ресурсів процесора.

Використання DMA розвантажує процесор і дозволяє досягти дуже високої швидкості передачі. Але якщо не планується передавати великі обсяги даних на високій швидкості, то можна використовувати побайтное передачу. Цей варіант дещо простіше в програмуванні.

# ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Программирование STM32F4. USART. Пример программы. [Електронний ресурс] – Режим доступу: http://microtechnics.ru/programmirovanie-stm32f4-usart-primer-programmy/.
2. STM32F4: INTERRUPT TIMER. [Електронний ресурс] – Режим доступу: http://amarkham.com/?p=29.
3. STM32F4xx: порты UART и USART. [Електронний ресурс] – Режим доступу: http://microsin.net/programming/arm/stm32f4xx-uart-and-usart.html.
4. STM32 с нуля. Использование модулей DMA и USART. [Електронний ресурс] – Режим доступу: https://microtechnics.ru/stm32-uchebnyj-kurs-dma.
5. Reference manual. STM32F40xxx, STM32F41xxx, STM32F42xxx, STM32F43xxx advanced ARM-based 32-bit MCUs. [Електронний ресурс] – Режим доступу: http://www.st.com/web/en/resource/technical/document/reference\_manual/DM00031020.pdf.