

УДК 681.3

Бабаш А. В., Квашнін В. О.**ОСОБЛИВОСТІ ВИКОРИСТАННЯ ПРЯМОГО ДОСТУПУ ДО ПАМ'ЯТІ DMA МІКРОКОНТРОЛЕРА STM32F4**

Для вирішення певних фонових завдань, таких як безперервна передача масивів даних, формування певної форми сигналу з використанням цифро-аналогового перетворювача, генерація сигналу за допомогою широтно-імпульсної модуляції (ШИМ) [1], передача даних через перетворювач USB-UART [2] дуже ефективно використовувати прямий доступ до пам'яті мікропроцесорного пристрою (DMA). DMA (Direct memory access) – прямий доступ до пам'яті, минаючи процесор. Цей механізм дозволяє виконувати будь-які дії (наприклад генерацію сигналу на виході ЦАП) без участі процесора. Такий механізм істотно прискорює роботу проектів, реалізованих з використанням мікроконтролерів. Він також дозволяє розгрузити процесор та виконувати ці завдання напряму, обходячи центральний процесор. Дана технологія дозволяє значно збільшити продуктивність мікропроцесорної системи в цілому. Таким чином, режим прямого доступу до пам'яті може бути використаний для певних монотонних фонових задач.

Мета роботи – визначення можливостей DMA мікроконтролера STM32F4, особливостей його використання та програмування, а також розробка методики та алгоритму налаштування роботи DMA.

Задачі дослідження:

- проведення літературного огляду існуючих рішень;
- ознайомлення з документацією мікроконтролера STM32F4;
- налаштування DMA для роботи з масивом чисел, який використовується для генерації сигналу на виході цифро-аналогового перетворювача з використанням прямого доступу до пам'яті.

Сучасні мікроконтролери сімейства STM32 містять в собі можливості використання прямого доступу до пам'яті. Даний режим роботи може бути ефективно застосований для генерації сигналів певної форми на виході цифро-аналогового перетворювача, для передачі даних на персональний комп'ютер з використанням USB UART та ін.

Для дослідження можливостей прямого доступу до пам'яті (DMA) сучасних мікроконтролерів обраний мікроконтролер STM32F4 [3–5], який є складовою частиною налагоджувальної плати STM32F4Discovery (рис. 1).



Рис. 1. Налагоджувальна плата STM32F4discovery

Характеристики мікроконтролера STM32F4 168 МГц, 1 Мб Flash, 192 Кб ОЗУ, ST-LINK / V2, 3-х осьовий акселерометр, цифровий мікрофон, USB-OTG, 24-розрядний аудіо ЦАП з підсилювачем класу D, 8 світлодіодів, 2 кнопки.

У якості прикладу буде розглянута генерація сигналу на виході ЦАП з використанням DMA. Генерація сигналу без використання DMA дуже рідко застосовується, так як виконувана програма постійно переривається. Все це впливає на швидкодію системи та її продуктивність в цілому.

Без використання DMA процесор постійно переривається від виконання основної програми і вона працює повільніше, причому уповільнення тим більше, чим більша частота генерації переривань від таймера ТІМ6 (у даному прикладі).

Для використання DMA потрібно розібратися з принципом генерації сигналу з використанням цієї технології.

Робота ЦАП достатньо проста, і фактично все, що він робить, це заносить нове значення в регістр даних для трансформації його в вихідну напругу на одному з виводів. Для організації передачі даних в модуль ЦАП введений параметр DAC_Trigger, який вказує подію, за якою модуль DAC запрошуватиме дані у DMA. Те, що може виступати в ролі такого тригера можна дізнатися з керівництва по програмуванню RM00090 Reference Manual [6] (рис. 2).

11.3.6 DAC trigger selection

If the TENx control bit is set, conversion can then be triggered by an external event (timer counter, external interrupt line). The TSELx[2:0] control bits determine which out of 8 possible events will trigger conversion as shown in [Table 42](#).

Table 42. External triggers

Source	Type	TSEL[2:0]
Timer 6 TRGO event	Internal signal from on-chip timers	000
Timer 8 TRGO event		001
Timer 7 TRGO event		010
Timer 5 TRGO event		011
Timer 2 TRGO event		100
Timer 4 TRGO event	101	
EXTI line9	External pin	110
SWTRIG	Software control bit	111

Each time a DAC interface detects a rising edge on the selected timer TRGO output, or on the selected external interrupt line 9, the last data stored into the DAC_DHRx register are transferred into the DAC_DORx register. The DAC_DORx register is updated three APB1 cycles after the trigger occurs.

If the software trigger is selected, the conversion starts once the SWTRIG bit is set. SWTRIG is reset by hardware once the DAC_DORx register has been loaded with the DAC_DHRx register contents.

- Note: 1 TSELx[2:0] bit cannot be changed when the ENx bit is set.
2 When software trigger is selected, the transfer from the DAC_DHRx register to the DAC_DORx register takes only one APB1 clock cycle.

Рис. 2. Події, тригери, що викликають передачу даних з DMA в ЦАП

На рис. 2 перерахований перелік всіх можливих подій, і серед них вже знайомі таймери, обраний ТІМ6 – цей таймер в даному прикладі буде генерувати подію, за якою DAC буде отримувати нове значення від модуля DMA та видавати на вихід (два виводи PA4 і PA5 для DAC1 та DAC2 відповідно) [3].

Потрібно налаштувати контролер прямого доступу до пам'яті (DMA), на сторінці керівництва по програмуванню (рис. 3), де описані потоки та канали для DMA1 (всього у контролера 2 блоку DMA) [7–9].

Table 20. DMA1 request mapping

Peripheral requests	Stream 0	Stream 1	Stream 2	Stream 3	Stream 4	Stream 5	Stream 6	Stream 7
Channel 0	SPI3_RX		SPI3_RX	SPI2_RX	SPI2_TX	SPI3_TX		SPI3_TX
Channel 1	I2C1_RX		TIM7_UP		TIM7_UP	I2C1_RX	I2C1_TX	I2C1_TX
Channel 2	TIM4_CH1		I2S2_EXT_RX	TIM4_CH2	I2S2_EXT_TX	I2S3_EXT_TX	TIM4_UP	TIM4_CH3
Channel 3	I2S3_EXT_RX	TIM2_UP TIM2_CH3	I2C3_RX	I2S2_EXT_RX	I2C3_TX	TIM2_CH1	TIM2_CH2 TIM2_CH4	TIM2_UP TIM2_CH4
Channel 4	UART5_RX	USART3_RX	UART4_RX	USART3_TX	UART4_TX	USART2_RX	USART2_TX	UART5_TX
Channel 5			TIM3_CH4 TIM3_UP		TIM3_CH1 TIM3_TRIG	TIM3_CH2		TIM3_CH3
Channel 6	TIM5_CH3 TIM5_UP	TIM5_CH4 TIM5_TRIG	TIM5_CH1	TIM5_CH4 TIM5_TRIG	TIM5_CH2		TIM5_UP	
Channel 7		TIM6_UP	I2C2_RX	I2C2_RX	USART3_TX	DAC1	DAC2	I2C2_TX

Рис. 3. Канали і потоки DMA

Для подальшої роботи використовуватиметься Channel 7, потік 6 (DAC2 Channel 7 / Stream 6). Для роботи з цифроаналоговим перетворювачем та DMA необхідно виконати наступні налаштування (рис. 4).

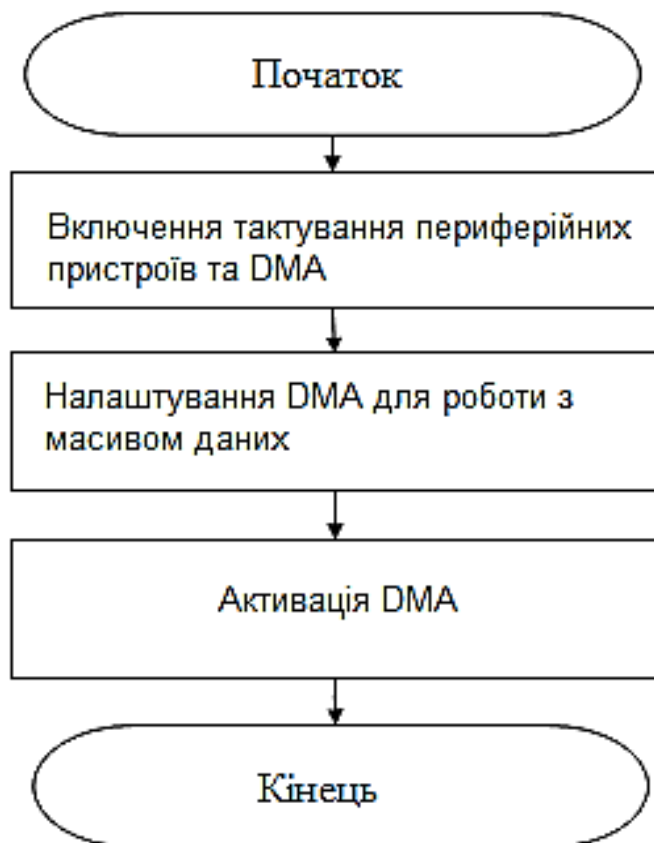


Рис. 4. Алгоритм налаштування DMA

За замовчуванням у мікроконтролері STM32F4 всі периферійні пристрої в цілях зниження енергоспоживання не активні. Щоб задіяти (активувати) певний периферійний пристрій (порт введення-виведення, ЦАП, DMA) необхідно подати на нього тактові імпульси від основної шини мікроконтролера. Дуже часто використовується термін «затакувати». Фрагмент програмного коду, який активує периферійні пристрої, наведений на рис. 5.

```

24 int main(void)
25 {
26     SystemInit(); // настройка тактирования
27
28
29     // включаем DMA1 GPIOA DAC TIM6
30     RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_DMA1 | RCC_AHB1Periph_GPIOA, ENABLE);
31     RCC_APB1PeriphClockCmd(RCC_APB1Periph_DAC, ENABLE);
32     RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM6, ENABLE);
33
34

```

Рис. 5. Активація периферійних пристроїв

Формування сигналу було виконано на основі масиву цілих чисел. Так як розрядність ЦАП (цифро-аналогового перетворювача) 12 біт, максимальне число, яке можна записати в регістр ЦАП буде 4095. Це число відповідає максимальній вихідній напрузі 3 В. Таким чином, можна встановлювати рівні вихідних напруг в діапазоні 0...3 В.

Значення регістра ЦАП визначається наступним чином [10]:

$$DAC_{вих} = \frac{U_{вих} \cdot 4095}{U_{max}}, \quad (1)$$

де $U_{вих}$ – необхідне значення вихідної напруги;

U_{max} – максимальне значення напруги;

$DAC_{вих}$ – значення регістра ЦАП.

Фрагмент програмного коду масиву даних для генерації вихідного сигналу ЦАП з використанням DMA наведений на рис. 6.

```

9 // амплитудные значения ЦАП для построения ступенчатого сигнала
10 const uint16_t step[64]= {
11     4095, 4095, 4095, 3000, 3000, 3000, 3000, 3000,
12     3000, 3000, 2000, 2000, 2000, 2000, 2000, 1000,
13     1000, 1000, 1000, 1000, 1000, 1000, 1500, 1500,
14     1500, 1500, 1500, 1500, 3000, 3000, 3000, 3000,
15     3000, 3000, 2000, 2000, 2000, 2000, 2000, 2000,
16     4095, 4095, 4095, 4095, 4095, 4095, 1000, 1000,
17     1000, 1000, 1000, 1000, 1000, 1000, 1000, 2000,
18     2000, 2000, 2000, 2000, 2000, 2000, 2000, 2000 };
19

```

Рис. 6. Масив значень для побудови ступінчастого вихідного сигналу на виході ЦАП

Для формування сигналу на виході ЦАП використовується масив цілих чисел в діапазоні 0..4095 (0...3 В). Даний масив містить 64 елемента.

Програмний код налаштувань DMA для роботи з масивом даних та подальшою його передачею ЦАП для генерації вихідного сигналу представлений на рис. 7.

```

87 // Для DAC1 НУЖНО ИСПОЛЬЗОВАТЬ СВЯЗКУ Stream5/Channel7
88 // Для DAC2 ИСПОЛЬЗУЕТСЯ Stream6/Channel7
89 DMA_DeInit(DMA1_Stream6);
90 DMA_InitTypeDef DMA_InitStructure;
91 DMA_InitStructure.DMA_Channel = DMA_Channel_7;
92 DMA_InitStructure.DMA_DIR = DMA_DIR_MemoryToPeripheral;
93 DMA_InitStructure.DMA_BufferSize = 64; // РАЗМЕР
94 DMA_InitStructure.DMA_Mode = DMA_Mode_Circular; // ЦИКЛИЧЕ
95 DMA_InitStructure.DMA_Priority = DMA_Priority_High;
96 DMA_InitStructure.DMA_MemoryBurst = DMA_MemoryBurst_Single;
97 DMA_InitStructure.DMA_PeripheralBurst = DMA_PeripheralBurst_Single;
98
99
100 DMA_InitStructure.DMA_PeripheralBaseAddr = (uint32_t)&DAC->DHR12R2; // Это пер
101 DMA_InitStructure.DMA_PeripheralInc = DMA_PeripheralInc_Disable;
102 DMA_InitStructure.DMA_PeripheralDataSize = DMA_PeripheralDataSize_HalfWord;
103
104 DMA_InitStructure.DMA_Memory0BaseAddr = (uint32_t) &step[0]; // Указате
105 DMA_InitStructure.DMA_MemoryInc = DMA_MemoryInc_Enable;
106 DMA_InitStructure.DMA_MemoryDataSize = DMA_MemoryDataSize_HalfWord;
107
108 DMA_Init(DMA1_Stream6, &DMA_InitStructure); // применим настройки DMA
109
110 DAC_DMACmd(DAC_Channel_2, ENABLE); // Включение связи DAC-DMA - (при генера
111 DAC_Cmd(DAC_Channel_2, ENABLE); // Включение DAC
112 DMA_Cmd(DMA1_Stream6, ENABLE); // Включение DMA - (при генера
113

```

Рис. 7. Налаштування прямого доступу до пам'яті DMA

Таким чином, необхідно вказати початком адрес масиву даних та задати розмір масиву (буфера `DMA_BufferSize`). Далі необхідно активувати режим прямого доступу до пам'яті та інші необхідні периферійні пристрої мікроконтролера (ЦАП).

Генерований сигнал ступінчастої форми на виході ЦАП згідно масиву даних (рис. 6), представлений на рис. 8.

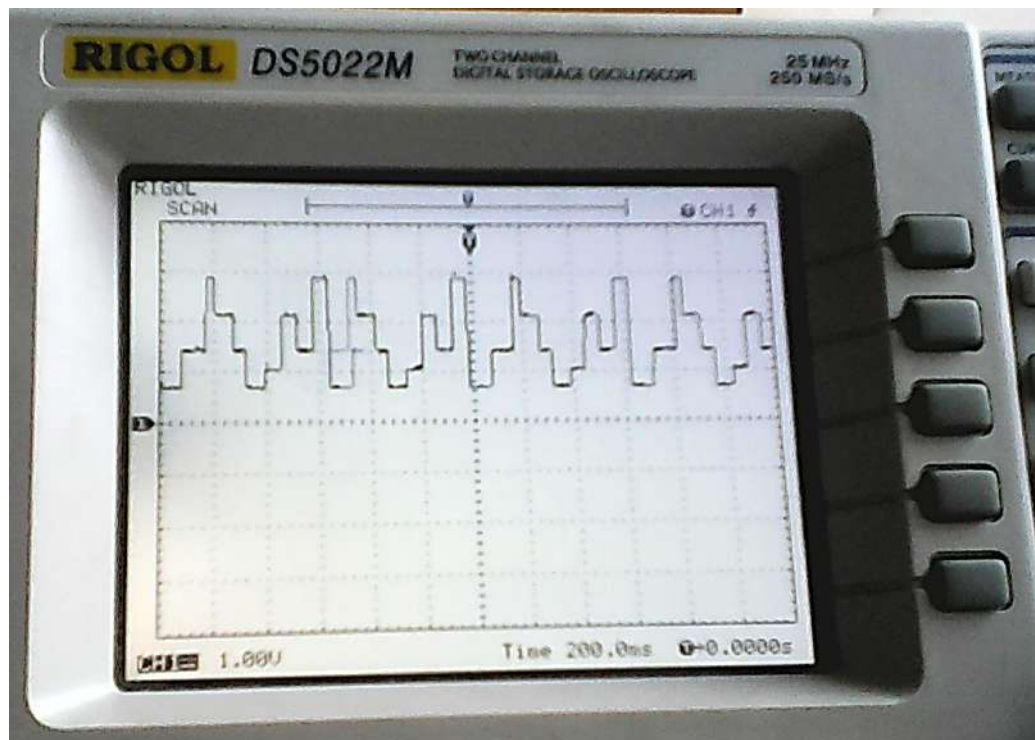


Рис. 8. Сигнал ступінчастої форми на виході ЦАП

ВИСНОВКИ

Були досліджені можливості використання прямого доступу до пам'яті DMA, визначено особливості його налаштування та програмування на мові високого рівня C. В загальному вигляді представлений алгоритм налаштування мікроконтролера для використання засобу прямого доступу до пам'яті. Для оцінки можливостей отримані програмні коди для налаштування та використання режиму DMA для генерації ступінчастого сигналу на виході ЦАП.

Детально розглянуто процес налаштування DMA. Використання даного режиму дозволяє проводити генерацію сигналу напряму, обминаючи центральний процесор. Це дозволяє підвищити швидкодію та продуктивність системи в цілому. Тобто генерація сигналу на виході ЦАП відбувається без участі центрального процесора.

Таким чином, отримані результати можуть бути використані для програмування мікроконтролерів STM32F4 при побудові систем контролю, вимірювальних систем та керування.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Kvashnin V. *Generating PWM by using microcontroller Stm32F4discovery* / V. Kvashnin, A. Babash // *Electrotechnic and computer systems*. – 2016. – № 22 (98). – P. 277–283.
2. Andrey Babash *Proprietary data transfer protocol between personal computers and microcontrollers STM32F4DISCOVERY development method* / Andrey Babash, Valeriy Kvashnin, Alexander Tarasov // *Proceedings of the International Symposium on Embedded Systems and Trends in Teaching Engineering Nitra, 2016*. – P. 30–35.
3. Квашинін В. О. *Методологія програмування мікроконтролерів Stm32F4Discovery та практичного їх застосування для вирішення наукових та інженерних задач* / В. О. Квашинін, А. В. Бабаши, В. В. Квашинін // *Сучасна освіта – доступність, якість, визнання : збірник наукових праць / під заг. ред. С. В. Ковалевського, д-ра техн. наук, проф.* – Краматорськ : ДДМА, 2016. – 209 с.
4. *Matlab Simulink application to program microcontrollers stm32f4discovery* / Vladyslav Kvashnin, Babash Andrey, Valeriy Kvashnin, Yuliya Lazutkina, Galina Klimenko // *Proceedings of the International Symposium on Embedded Systems and Trends in Teaching Engineering*. – Nitra, 2016. – P. 134–140.
5. Квашинін В. В. *Програмування мікроконтролерів STM32f4discovery в Matlab Simulink* / В. В. Квашинін, А. В. Бабаши, Г. П. Клименко // *Сучасні проблеми і досягнення в галузі радіотехніки, телекомунікацій та інформаційних технологій : тези доповідей VIII Міжнародної науково-практичної конференції (21–23 вересня 2016 р., м. Запоріжжя)*. – Запоріжжя : ЗНТУ, 2016. – С. 287–289.
6. *Обитель RC-инженера [Электронный ресурс]*. – Режим доступа: http://vg.ucoz.ru/publ/programirovanie_mikrokontrollerov_stm32/stm32f4_cap_dac_po_vzrosloму/9-1-0-28.
7. Бугаев В. И. / *Лабораторный практикум для изучения микроконтроллеров архитектуры ARM Cortex-M4 на базе оценочного модуля STM32F4Discovery* / В. И. Бугаев, М. П. Мусиенко, Я. М. Крайних. – Москва – Николаев : МФТИ, ЧГУ, 2013. – 71 с.
8. *DM00031020 Reference manual [Электронный ресурс]*. – Режим доступа: URL: http://www.st.com/st-web/ui/static/active/en/resource/technical/document/reference_manual/DM00031020.pdf.
9. *DM00039084 User manual [Электронный ресурс]*. – Режим доступа: URL: http://www.st.com/st-web/static/active/en/resource/technical/document/user_manual/DM00039084.pdf.
10. Квашинін В. О. *Використання вбудованих цифро-аналогового і аналого-цифрового перетворювачів мікроконтролера Stm32f4discovery [Електронний ресурс]* / В. О. Квашинін, А. В. Бабаши // *Научный Вестник ДГМА. – № 1 (19Е). – 2016. – С. 47–58. – Режим доступа: [http://www.dgma.donetsk.ua/science_public/science_vesnik/%E2%84%961\(19%D0%95\)_2016/article/9.pdf](http://www.dgma.donetsk.ua/science_public/science_vesnik/%E2%84%961(19%D0%95)_2016/article/9.pdf)*.