

ZASTOSOWANIE DMA W UKŁADACH SZEREGOWEJ TRANSMISJI DANYCH

Dariusz SOWIŃSKI¹, Marek KCIUK²

1. Politechnika Śląska, Wydział Elektryczny, Katedra Mechatroniki
tel.: 32 237 28 03 e-mail: dariusz.sowinski91@gmail.com ,
2. Politechnika Śląska, Wydział Elektryczny, Katedra Mechatroniki
tel.: 32 237 28 03 e-mail: marek.kciuk@polsl.pl,

Streszczenie: W artykule przedstawiono projekt układu wykorzystującego w transmisji danych kontroler DMA. Głównym zadaniem kontrolera jest odciążenie jednostki centralnej użytego mikrokontrolera przez bezpośredni dostęp do jego pamięci. Badanie transmisji przeprowadzono za pomocą analizatora stanów logicznych LOGIC8 wraz z dedykowanym do niego oprogramowaniem producenta. Analiza otrzymanych wyników potwierdza, iż uzyskana szybkość transmisji danych przy zastosowaniu kontrolera DMA jest o 16% większa niż w przypadku klasycznej obsługi procesu komunikacji z zastosowaniem procesora.

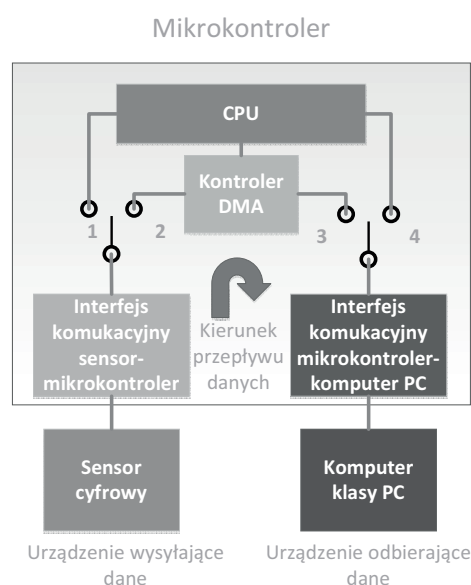
Słowa kluczowe: szeregowa transmisja danych, DMA, testowanie komunikacji.

1. WPROWADZENIE

W dzisiejszym świecie postęp technologiczny dokonuje się szybciej niż kiedykolwiek, między innymi poprzez zastosowanie półprzewodnikowych, programowalnych układów scalonych w przedmiotach codziennego użytku.

Jako jeden z istotnych trendów rozwoju można zaobserwować bardzo gwałtowny postęp technologii IoT – tzw. Internet rzeczy (Internet of Things). W ramach tej koncepcji wymagane jest zbieranie, przetwarzanie, przesył oraz archiwizacja znacznych ilości danych na niespotykaną dotąd skalę. Bardzo intensywnie rozwijane są technologie przesyłu tak dużych ilości danych z dużą szybkością. Metoda przesyłu danych z pominięciem jednostki centralnej mikrokontrolera z zastosowaniem kontrolera DMA w znacznym stopniu usprawnia i przyspiesza komunikację. Tematyka ta jest poruszana w pracach badawczych [1].

Opracowany układ umożliwi pomiar szybkości przepływu informacji między urządzeniem wysyłającym i odbierającym dane. Cały proces jest sterowany za pomocą mikrokontrolera. Głównym celem pracy jest zbadanie współpracy interfejsów komunikacyjnych z mechanizmem DMA. Zaimplementowany algorytm umożliwi odciążenie jednostki centralnej mikrokontrolera. Takie rozwiązanie zmniejsza czas transmisji. Na rysunku 1 została przedstawiona zasada działania układu komunikacyjnego, w którym możliwa jest komunikacja z zastosowaniem kontrolera DMA oraz bez niego. Możliwe są cztery tryby pracy: komunikacja bez DMA, pełna komunikacja DMA (z pominięciem jednostki centralnej), komunikacja DMA tylko po stronie czujnika oraz komunikacja DMA tylko po stronie komputera.



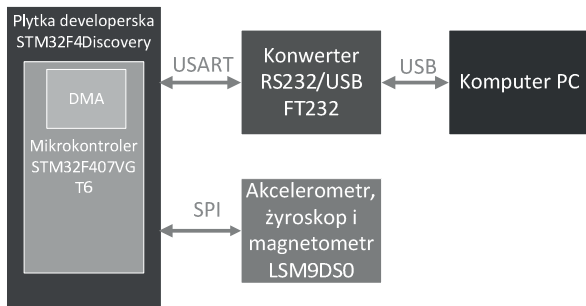
Rys. 1. Schemat poglądowy przedstawiający istotę działania układu wykorzystującego DMA w układach transmisji danych

2. KONSTRUKCJA STANOWISKA

Opracowany układ składa się z następujących elementów:

- Zintegrowanego dziewięcioosiowego (9DOF) czujnika o symbolu LSM9DS0 (generator danych) [2],
- Płytki developerskiej STM32F4-DISCOVERY (przetwarzanie danych) [3],
- Konwertera RS232/USB (dopasowanie interfejsów komunikacyjnych między mikrokontrolerem, a komputerem PC),
- Komputera PC z uruchomioną aplikacją (archiwizacja danych).

Mikroprocesor STM pobiera dane z czujnika za pośrednictwem magistrali SPI, a następnie przesyła je do komputera PC za pośrednictwem magistrali UART/USB. Proces komunikacji można przedstawić na rysunku 2. Jego analiza zostanie przedstawiona zgodnie z kierunkiem transferu danych między blokami przedstawionymi na schemacie.



Rys. 2. Schemat sposobu połączenia poszczególnych bloków urządzenia wykorzystującego DMA

W stanowisku zastosowano czujnik 9DOF (ang. 9 Degrees of Freedom) LSM9DS0, który umożliwia komunikację za pośrednictwem magistral I2C (ang. Inter-Integrated Circuit) oraz SPI (ang. Serial Peripheral Interface). Na potrzeby projektu zaimplementowano obsługę magistrali SPI. Powodem jest maksymalna częstotliwość sygnału zegarowego, która wg danych podanych przez producenta dla I2C wynosi 400 kHz (tryb fast), natomiast dla SPI jest to 10 MHz [4][5]. Komunikacja między tym modulem, a płytką developerską jest dwukierunkowa, ponieważ w pierwszym etapie transmisji następuje konfiguracja czujnika. Model komunikacji zrealizowany jest w trybie master-slave.

Układem zarządzającym wymianą danych jest 32-bitowy mikrokontroler STM32f407VGT6 (rdzeń chipa to ARM Cortex-M4F) wyprodukowany przez firmę STMicroelectronics. Zawiera on moduły umożliwiające komunikację z urządzeniami za pośrednictwem różnych interfejsów szeregowych. Wśród dostępnych opcji znajdują się wspomniany wcześniej SPI, a także UART (ang. Universal Asynchronous Receiver and Transmitter) oraz zmodernizowana wersja USART (ang. Universal Synchronous and Asynchronous Receiver and Transmitter). Zastosowany mikrokontroler posiada dwa wbudowane kontrolery DMA, z których każdy może obsłużyć do ośmiu strumieni. Kontrolery są niezbędne do skrócenia czasu transmisji danych oraz odciążenia jednostki centralnej.

Dane pobrane z sensora są gromadzone w pamięci RAM a następnie wysłane do komputera PC. W tym celu zastosowano dwukierunkowy interfejs transmisji szeregowy USART, który w warstwie łącza danych jest kompatybilny z interfejsem RS232, jednak jest coraz rzadziej stosowany. Obecnie, w sprzęcie komputerowym standardowo stosuje się uniwersalny interfejs USB (ang. Universal Serial Bus), który zastąpił większość stosowanych wcześniej interfejsów komunikacyjnych, zarówno szeregowych jak i równoległych [6]. Stąd też wynika potrzeba zastosowania sprzętowego adaptera RS232/USB.

3. PRZYKŁADOWE POMIARY

Przeprowadzone badania polegały na wykonaniu serii transmisji przy różnych ustawieniach szybkości, liczby bajtów w ramce danych oraz powtórzeń. W trakcie transmisji wymuszano obciążenie procesora pętlami opóźniającymi będącymi symulacją obciążenia procesora innymi zadaniami. Dokonano pomiaru czasów trwania transmisji przy różnych konfiguracjach z aktywnym i nieaktywnym modulem DMA.

W celu łatwej konfiguracji opracowano strukturę danych w postaci zdefiniowanego typu (typedef) LSM9DS0_Trans_x, zawierającego parametry komunikacji.

Umożliwia to szybkie przekonfigurowanie programu między kolejnymi testami, bez konieczności zagłębiania się w strukturę kodu mikrokontrolera [7][8].

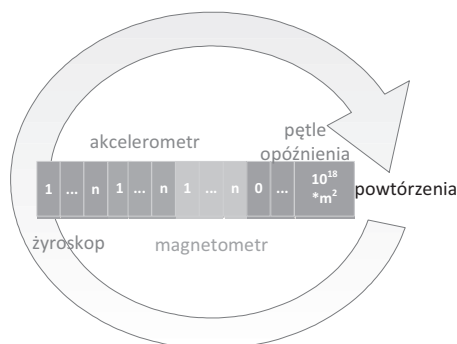
Tablica 1. Struktura typedef zawierająca parametry transmisji

L.P.	LSM9DS0_Trans_x
1	mode_of_transmission_SPI_x
2	mode_of_transmission_UART_x
3	number_of_measurements_x
4	number_of_used_blocks_x
5	number_of_operation_in_delay_loop_x

Dwa pierwsze elementy struktury (1) i (2) pozwalają wybrać dowolny dostępny tryb pracy interfejsów SPI oraz USART. Dostępne tryby to: polling – transmisja jest w pełni kontrolowana przez procesor, jest to tryb blokujący, IT – transmisja jest realizowana przez procesor w trybie przerwanowym, DMA – transmisja jest realizowana z pominięciem jednostki centralnej natomiast jest realizowana przez kontroler DMA. Trzy pozostałe elementy umożliwiają sterowanie innymi opcjami konfiguracyjnymi transmisji. Element (3) definiuje liczbę odczytów wszystkich rejestrów wyjściowych (6 rejestrów) wybranego urządzenia tj. żyroskopu, akcelerometru lub magnetometru. Kolejny element (4) określa ilość powtórzeń całego procesu. Ostatni element: (5) precyzuje liczbę operacji mnożenia liczb zmiennoprzecinkowych w pętli opóźnienia.

W celu dokładniejszego wyjaśnienia cały proces został zobrazowany na rysunku 3 (na rysunku oznaczono elementy struktury typedef następującymi literami:

- n* - number_of_measurements_x (3),
- o* - number_of_used_block_x (4),
- m* - number_of_operation_in_delay_loop_x (5).



Rys.3. Koncepcja procesu cyklicznej transmisji danych

Aby lepiej zrozumieć rysunek należy zapoznać się z poniższymi zależnościami:

$$n_{byt} = (6 \cdot n) \quad (1)$$

$$n_{buf} = (3 \cdot o) \quad (2)$$

$$n_{rec} = n_{byt} \cdot n_{buf} \quad (3)$$

gdzie:

n_{byt} – liczba bajtów w ramce danych,

n_{buf} – liczba ramek danych,

n_{rec} – liczba odebranych bajtów.

W badaniach dobrano wartości parametrów n oraz o tak, aby zbadać czas przesyłu $n_{rec} = 18000$ B, co odpowiada wartości prawie 18 kB, dla wszystkich konfiguracji obydwu interfejsów. Jednak każdy pomiar będzie się różnił liczbą przestanych ramek danych oraz liczbą bajtów znajdujących w pojedynczym buforze.

Badania zostały przeprowadzone dla pięciu przypadków nazwanych wariantami, tj.

1. $(n_{byt} \cdot 1) \cdot (n_{buf1}) = 3000$ ramek 6 bajtowych,
2. $(n_{byt} \cdot 10) \cdot (n_{buf2}) = 300$ ramek 60 bajtowych,
3. $(n_{byt} \cdot 100) \cdot (n_{buf3}) = 30$ ramek 600 bajtowych,
4. $(n_{byt} \cdot 250) \cdot (n_{buf4}) = 12$ ramek 1500 bajtowych,
5. $(n_{byt} \cdot 500) \cdot (n_{buf5}) = 6$ ramek 3000 bajtowych.

Rozwiązanie ma na celu sprawdzenie, czy efektywniejsze okaże się wysyłanie większej ilości niewielkich porcji danych, czy może lepszym rozwiązaniem jest wysyłanie danych zgromadzonych w większych buforach. Dodatkowo, dla każdego z powyższych wariantów uwzględniono obciążenie procesora wynikające z realizacji innych zadań. Obciążenie to zasymulowano generując pętlę opóźniająca. Uwzględniono trzy przypadki obciążenia procesora pętlami wykonującymi 10^{18} , 10^{20} oraz 10^{22} iteracji mnożenia liczb zmiennoprzecinkowych równocześnie z realizacją procesu transmisji danych. Możliwe jest również przeprowadzenie pomiarów bez symulacji obciążenia procesora – ten przypadek nie był jednak w ramach pracy analizowany.

Teoretyczna szybkość transmisji dla SPI i USART wyniosła 5,25 Mb/s. Czasy transmisji, podane w milisekundach, dla przypadku 10^{18} wykonanych pętli opóźnienia zostały zamieszczone w tabeli 2.

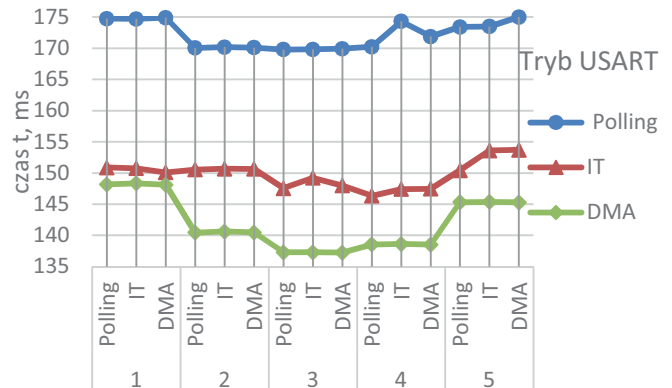
Tablica 2. Wyniki pomiarów wykonanych analizatorem stanów logicznych

Numer wariantu	Tryb USART			
	Tryb SPI	m=1, 10^{18} pętli opóźnienia		
		Polling	IT	DMA
1	Polling	174,737	150,881	148,162
	IT	174,694	150,763	148,341
	DMA	174,885	150,087	148,111
2	Polling	170,050	150,556	140,453
	IT	170,160	150,672	140,619
	DMA	170,084	150,666	140,462
3	Polling	169,795	147,565	137,296
	IT	169,805	149,222	137,284
	DMA	169,918	147,988	137,232
4	Polling	170,217	146,341	138,560
	IT	174,305	147,420	138,623
	DMA	171,874	147,468	138,526
5	Polling	173,404	150,413	145,317
	IT	173,488	153,611	145,364
	DMA	174,990	153,701	145,301

4. ANALIZA WYNIKÓW

Dane zawarte w tabeli 2 przedstawiono graficznie na wykresie znajdującym się na rysunku 4. Wyraźnie widać, iż

najdłuższe czasy występują w przypadku transmisji USART w trybie polling, natomiast najkrótsze czasy zostały uzyskane w trybie DMA (po stronie USART). W przypadku transmisji z zastosowaniem magistrali SPI zmiana trybów przynosi mniej widoczne efekty. Najprawdopodobniej jest to spowodowane faktem, iż komunikacja z zastosowaniem SPI jest ściśle określona czasowo, tzn. po wysłaniu bajtu zawierającego adres komórki przechowującej interesujące nas dane w odpowiedzi otrzymujemy ich wartość.



Rys.4. Wykres zależności czasowych podczas przeprowadzenia transmisji – 1018 pętli opóźnienia

Komunikację w trybie polling przyjęto jako referencyjną. Wyniki pozostałych dwóch trybów porównano w odniesieniu do tego trybu w celu sprawdzenia zwiększenia szybkości transmisji przy zastosowaniu trybów nieblokujących w porównaniu do transmisji blokującej jednostkę centralną. Dodatkowo porównano obydwa tryby nieblokujące, w tym przypadku wyniki odniesiono do trybu przerwaniowego, który również angażuje procesor. Opracowano trzy wskaźniki względnej zmiany szybkości transmisji dla analizowanych trybów:

- W_{PI} – wskaźnik względnej zmiany szybkości trybu przerwaniowego względem trybu polling,
- W_{PD} – wskaźnik względnej zmiany szybkości trybu DMA względem trybu polling,
- W_{ID} – wskaźnik względnej zmiany szybkości trybu DMA względem trybu przerwaniowego.

Powyższe wskaźniki uwzględniają wyłącznie ten sam rodzaj transmisji w obydwu magistralach SPI i USART. Proponowane wskaźniki mają następującą postać:

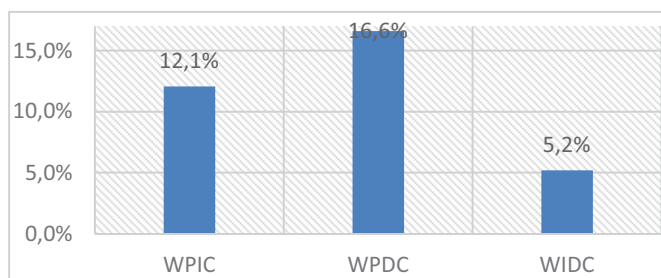
$$W_{PI} = \frac{\sum_{i=1}^k \left(\frac{t_{Polling} - t_{IT}}{t_{Polling}} \right)_i}{k} * 100\%, \quad (4)$$

$$W_{PD} = \frac{\sum_{i=1}^k \left(\frac{t_{Polling} - t_{DMA}}{t_{Polling}} \right)_i}{k} * 100\%, \quad (5)$$

$$W_{ID} = \frac{\sum_{i=1}^k \left(\frac{t_{IT} - t_{DMA}}{t_{IT}} \right)_i}{k} * 100\%, \quad (6)$$

gdzie: W_{PI} , W_{PD} , W_{ID} to wyznaczone wskaźniki, k – liczba pomiarów w serii, $t_{polling}$, t_{IT} , t_{DMA} są to czasy trwania transmisji zawarte w tabeli 2.

Na rysunku 5 przedstawiono wartości współczynników W_{PI} , W_{PD} oraz W_{ID} , wyznaczonych jako średnie wartości współczynników W_{PI} , W_{PD} , W_{ID} obliczone dla wszystkich przypadków obciążenia procesora oraz wszystkich pięciu wariantów.



Rys.5. Przedstawienie współczynników WPIC, WPDC, WIDC

Analizując uzyskane wyniki można sformułować następujące spostrzeżenia: obydwie przypadki nieblokujące (TI oraz DMA) w znacznym stopniu poprawiły szybkość transmisji względem trybu blokującego, odpowiednio o 12,1% oraz 16,6%. Niewielka różnica na korzyść trybu DMA została zweryfikowana w trzecim porównaniu, w którym w trybie DMA uzyskano o 5,2% większą szybkość transmisji niż trybie TI.

Najkrótszy czas przesyłania testowego bloku 18 kB danych wyniósł 137 ms dla konfiguracji połączeń DMA-DMA w wariancie trzecim struktury danych (30 buforów 600 bitowych), natomiast w najwolniejszym przypadku proces ten trwał 538 ms przy konfiguracji pooling-DMA oraz maksymalnej liczbie pętli opóźniających wynoszącej 10^{22} w wariancie pierwszym (3000 buforów 6 bajtowych). Wynik ten nie został ujęty części tabeli zaprezentowanej w artykule. Stosunek najkrótszego czasu do najdłuższego uzyskany w czasie badań wynosi prawie 4, przy przesłaniu tej samej porcji danych. Po określeniu tych wartości można obliczyć całkowitą szybkość transmisji danych:

$$V_{\text{całkowita}} = n_{\text{rec}}/t \quad (7)$$

Korzystając ze wzoru (7) obliczono minimalną oraz maksymalną uzyskaną szybkość transmisji uwzględniając konieczność sterowania czujnikiem oraz obciążenie procesora innymi zadaniami (symulowane pętle opóźnienia).

- szybkość minimalna:

$$V_{\text{całkowita min}} = \frac{18\,000 \text{ bajtów}}{0,537707 \text{ s}} = 33475,48 \approx 33,5 \text{ kB/s} \quad (8)$$

- szybkość maksymalna:

$$V_{\text{całkowita max}} = \frac{18\,000 \text{ bajtów}}{0,137232 \text{ s}} = 131164,58 \approx 131,2 \text{ kB/s} \quad (9)$$

Jak widać, obliczone wartości różnią się znacznie. Jednak, aby możliwe było porównanie ich z szybkością skonfigurowaną w interfejsie należy sobie zdać sprawę, iż do odbioru jednego bajtu na komputerze należy przetworzyć 3 bajty na mikrokontrolerze (adres bajtu rejestru odczytanego sensora przez SPI, dane pobrane z rejestru sensora interfejsem SPI, wysłany bajt magistralą USART). Szybkość skonfigurowaną należy więc przekształcić wyliczając rzeczywistą szybkość uwzględniającą rozmiar transmitowanej ramki danych:

$$\frac{5,25 \text{ Mb/s}}{8 \text{ bit}} = 0,65625 \text{ MB/s} = 656,25 \text{ kB/s} \quad (10)$$

$$\frac{656,25 \text{ kB/s}}{3} = 218,75 \text{ kB/s} \quad (11)$$

W ten sposób została wyznaczona maksymalna teoretyczna przepustowość. Jednak jej wartość nie uwzględnia przełączania źródła danych za pomocą sygnału Chip Select (CS) zmieniającego sygnał źródłowy w czujniku LSM9DS0 oraz zmiany funkcji zapis/odczyt w przypadku interfejsu SPI, co jest do zaobserwowania na przebiegach uzyskanych z analizatora stanów logicznych.

5. PODSUMOWANIE

W artykule potwierdzono, że zastosowanie kontrolera DMA poprawiło prędkość transmisji o ponad 16%. Przedstawiono przyjętą metodologię badań, uzyskane wyniki pomiarowe czasu transmisji dla jednej serii pomiarowej (obciążenie procesora 10^{18} pętłami opóźniającymi) oraz analizę otrzymanych wyników.

Obliczone w ostatnim rozdziale wskaźniki uwidaczniają różnice szybkości transmisji wynikające z konfiguracji sprzętowej. Najlepszy efekt uzyskano przy pełnym przejściu transmisji przez DMA. Różnica wyniosła 16,6% w porównaniu do sytuacji, w której jednostka centralna musiała obsługiwać proces komunikacji. Warto także dodać, że transfer z zastosowaniem DMA w porównaniu do transferu za pomocą przerwań jest również efektywniejszy, w tym przypadku o 5,2%.

Uzyskano efektywną przepustowość równą 60% przepustowości teoretycznej, co jest dobrym wynikiem uwzględniając fakt, że w trakcie przesyłu danych musiała odbywać się także rekonfiguracja czujnika za pomocą sygnału CS. Można przypuszczać, że dobranie innego typu czujnika, nie wymagającego przełączania sygnałów oraz zastosowanie zoptymalizowanej struktury danych spowoduje dalsze poprawienie wyników.

6. BIBLIOGRAFIA

1. Palumbo F.; Pani D., Pilia A., Raffo L.: Impact of Half-Duplex and Full-Duplex DMA Implementations on NoC Performance. 4 ACM/IEEE International Symposium on Networks-on-Chip (NOCS), Grenoble, Francja, 2010.
2. Nota katalogowa czujnika LSM9DS0 <http://www.st.com/content/ccc/resource/technical/document/datasheet/ab/2a/3b/45/f0/92/41/73/DM00087365.pdf/files/DM00087365.pdf/jcr:content/translations/en.DM00087365.pdf> (dostęp 18.05.2017),
3. Podręcznik użytkownika zestawu STMdiscovery F4 http://www.st.com/content/ccc/resource/technical/document/user_manual/70/fe/4a/3f/e7/e1/4f/7d/DM00039084.pdf/files/DM00039084.pdf/jcr:content/translations/en.DM00039084.pdf (dostęp 18.05.2017),
4. Bogusz J.: Lokalne interfejsy szeregowy w systemach cyfrowych, BTC, Warszawa, 2004,
5. Mielczarek W.: Szeregowy interfejsy cyfrowe, Helion, Gliwice, 1993,
6. Metzger P.; Siemieniacki M.: Anatomia PC. Architektura komputerów zgodnych z IBM PC, Helion, Gliwice, 2003,
7. Paprocki K.: Mikrokontrolery STM32 w praktyce, BTC, Legionowo, 2009,
8. Kardaś M.: Mikrokontrolery AVR. Język C. Podstawy programowania, ATNEL, Szczecin, 2013,

THE USE OF DMA IN SERIAL DATA TRANSMISSION SYSTEMS

In this paper authors proved that DMA controller have positive effect on the serial transmission process. Firstly, was presented a scheme with principles of the device using DMA controller. The stream of data sent from sensor can use one of four modes, it means: without DMA, with DMA on sensor side, with DMA on PC side and with DMA on both sides. The main device was assembled with following modules: LSM9DS0 sensor as a data generator, STM32F4 ARM microcontroller for data transmsion, USART-USB converter and PC computer as a data reader. The main module is the ARM microcontroller which controls whole process. Additionally, authors present used transmission interfaces working between the modules. In research 18 kB of data is send from the sensor to the computer and time of data transmission is measured. There are different structures of data frame in range of 3000 buffers containing 6b of data to 6 buffers containing 3000b. The speed for the SPI and UART amounted to 5.25 Mb/s. The times for the case: 10^{18} executed delay loops are presented in Table 2 and are given in milliseconds. In result transmission fully controlled by DMA is 16,6% faster than CPU controlled one (pulling mode) and 5,2% faster than CPU utilizing interrupts. Transmission controlled by CPU utilizing interrupts is 12,1% faster than CPU controlled one (in pulling mode). The fastest transmission achieved 60% of theoretical transmission speed.

Keywords: serial duplex transmission, direct memory access, communication testing, performance analysis.

