

# Aplikácia radiča DMA v sériovej komunikácii SPI

<sup>1</sup>Patrik JACKO, <sup>2</sup>Dobroslav KOVÁČ

<sup>1,2</sup> Katedra teoretickej a priemyselnej elektrotechniky, Fakulta elektrotechniky a informatiky, Technická univerzita v Košiciach, Slovenská republika

<sup>1</sup> patrik.jacko.2@tuke.sk, <sup>2</sup> dobroslav.kovac@tuke.sk

**Abstrakt** — Rýchle konvertovanie analógových signálov ešte nemusí zaručovať jeho následné rýchle spracovanie. Keďže rýchlosť je v súčasnosti nevyhnutnou časťou spracovania signálov, je potrebné zabezpečiť čo najrýchlejšie ukladanie dát do pamäte. Sériová komunikácia SPI patrí medzi najrýchlejšie a najvyužívanejšie rozhrania na prenos dát medzi mikrokontrolérmi alebo medzi mikrokontrolérom a inými perifériami. Aj napriek tomu, že SPI dosahuje vysoké rýchlosti prenosu dát, je možné túto rýchlosť ešte viac zvýšiť využitím radiča DMA, teda priameho prístupu k pamäti. Tento článok hovorí o aplikácii DMA v sériovej komunikácii SPI, kde bude možné porovnať čas odosielania dát bez DMA a s použitím radiča DMA.

**Kľúčové slová:** — radič DMA, rozhranie SPI, STM32, STM32F446RE

## The DMA controller application in the SPI serial communication

**Abstract** — Fast conversion of analog signals may not warrant its fast signal processing. The speed is nowadays essential part of signal processing, therefore it needs ensure the fastest data saving to the storage. The SPI serial communication is one of the fastest and most used interfaces for data transfer between microcontrollers or between microcontroller and other peripherals. Even through SPI achieves high data rates, this speed can be further enhanced by using the DMA controller, that is, direct access to memory. This article describes DMA in the SPI communication, where it will be possible to compare time to send data without DMA and using the DMA controller.

**Keywords** —DMA controller, SPI serial communication, STM32, STM32F446RE

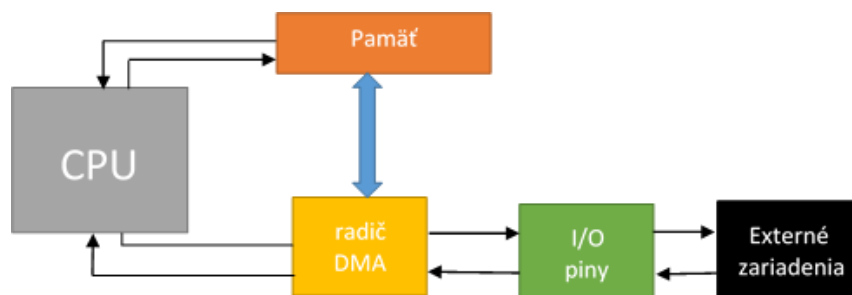
### I. ÚVOD

V niektorých situáciách, kedy potrebujeme v čo najrýchlejšom čase odosielať veľké množstvo dát medzi mikrokontrolérmi môže rozhodujúcu úlohu zohrávať spôsob komunikácie. V prvom rade je vhodné zvoliť správne rozhranie, a teda či dáta budeme odosielať synchronne pomocou pravidelných hodinových impulzov, alebo asynchronne ako je to v prípade rozhrania UART. Je jasné, že rýchlejší spôsob odosielania dát je synchronný mód, preto v tomto článku budeme hovoriť o synchronnej komunikácii SPI. Nakoľko klasickému odosielaniu dát sériovou linkou SPI sme sa venovali v predchádzajúcom článku[3], v tomto článku sa budeme venovať aplikácii radiča DMA pri sériovej komunikácii SPI. Zároveň budeme merať čas odosielania/príjmu pri 200 hodnotách. Rýchlosť prenosu dát porovnáme s prenosom rovnakého objemu dát bez použitia DMA radiča. Následne vyhodnotíme pomer rýchlosti prenosu bez DMA a s DMA.

## II. MIKROKONTROLÉRY STM32F4 A RADIČ DMA

### Radič DMA:

Radič DMA alebo *Direct memory access* sprostredkúva, ako už z názvu je zrejme, priamy prístup k pamäti. Jeho úlohou je uskutočniť prenos dát z pamäte k periférii, z periférie do pamäte alebo medzi dvomi pamäťami. Svojou činnosťou dokáže „odbrejniť“ procesor, ktorý za iných okolností musí dáta spracovať, ukladať do pamäte alebo ich posúvať z pamäte k periférii ktorá ich následne odošle. Použitím DMA je možné tento proces obísť, čím sa zvýši rýchlosť odosielania/prijmu dát, navyše procesor počas spomínaných akcií môže vykonávať ďalšie operácie.



Obr. 1 Principiálna bloková schéma ukladania dát pomocou DMA

Mikrokontroléry STM32F446RE obsahujú dva radiče DMA z čoho každý radič disponuje ôsmimi streamami. K jednotlivým streamom sú privedené kanály na ktorých sú priamo pripojené periférie mikrokontroléra. Mikrokontrolér STM32F446RE obsahuje štyri zbernice SPI, ktoré sú pripojené k DMA1 a DMA2. preto je dôležité pre vybrané SPI zvoliť správny radič DMA, stream ku ktorému je periféria pripojená a taktiež správne kanály ku ktorým sú pripojené komunikačné piny (MISO a MOSI).

## III. SÉRIOVÁ KOMUNIKÁCIA SPI RIADENÁ RADIČOM DMA

V predchádzajúcom článku sme sa venovali nastaveniu periférie SPI, odosielaniu dát cez túto zbernicu a meraniu času prenášania dát. Uviedli sme, že čas za ktorý prebehlo prenesenie 200 hodnôt z master zariadenia do slave zariadenia bol 182,322  $\mu$ s, čo predstavuje reálnu rýchlosť 17,552 MHz. Pozorovaním odosielania/prijmu dát sme zistili, že jednotlivé bitové slová (16 bitov) sa pri nastavenej prenosovej rýchlosti 45 Mbit/s naozaj prenášajú touto rýchlosťou, no výsledné oneskorenie je spôsobené spracovaním dát, ukladaním do pamäte a podobne. Aplikácia radiča DMA tento problém úplne odstraňuje a garantuje, že pri zvolenej prenosovej rýchlosti 45Mbit/s budú dáta v ľubovoľnom množstve odosielané touto rýchlosťou bez akéhokoľvek oneskorenia.

```
226 void dma_spi_RX_config()
227 {
228     DMA2_Stream2->CR |= (3<<25);
229     DMA2_Stream2->PAR = (uint32_t)&SPI1->DR;
230     DMA2_Stream2->M0AR = (uint32_t)&spi_received_data_DMA;
231     DMA2_Stream2->NDTR = 1;
232     DMA2_Stream2->CR |= DMA_SxCR_CIRC;
233     DMA2_Stream2->CR &= ~DMA_SxCR_CT;
234     DMA2_Stream2->CR &= ~DMA_SxCR_DIR;
235     DMA2_Stream2->CR |= DMA_SxCR_MSIZE_0;
236     DMA2_Stream2->CR |= DMA_SxCR_PSIZE_0;
237     DMA2_Stream2->CR |= DMA_SxCR_PL_1;
238     DMA2_Stream2->CR |= DMA_SxCR_TCIE;
239     DMA2_Stream2->CR |= DMA_SxCR_EN;
240
241     NVIC_EnableIRQ(DMA2_Stream2_IRQn);
242     NVIC_SetPriority(DMA2_Stream2_IRQn,1);
243 }
244
```

```
245 void dma_spi_TX_config()
246 {
247     RCC->AHB1ENR |= RCC_AHB1ENR_DMA2EN;
248     DMA2_Stream5->CR |= (3<<25);
249     DMA2_Stream5->CR &= ~DMA_SxCR_CT;
250     DMA2_Stream5->CR |= DMA_SxCR_DIR_0;
251     DMA2_Stream5->CR |= DMA_SxCR_MINC;
252     DMA2_Stream5->CR |= DMA_SxCR_MSIZE_0;
253     DMA2_Stream5->CR |= DMA_SxCR_PSIZE_0;
254     DMA2_Stream5->CR |= DMA_SxCR_PL_1;
255     DMA2_Stream5->CR |= DMA_SxCR_TCIE;
256
257     NVIC_EnableIRQ(DMA2_Stream5_IRQn);
258     NVIC_SetPriority(DMA2_Stream5_IRQn,1);
259 }
```

Obr. 2 Nastavenie radiča DMA pre perifériu SPI. (vľavo pre kanál prijmu, vpravo pre odosielanie dát)

Na Obr. 2 môžeme vidieť konfiguráciu radiča DMA2 pre perifériu SPI1. Okrem povolenia taktovacích hodín pre radič DMA2 je potrebné zvoliť správny stream a kanál na ktorých je periféria

definovaná. Môžeme vidieť, že linka na odosielanie dát (pin MOSI) je pripojená k streamu 5 a linka na príjem dát je pripojená k stream 2. V nasledujúcich príkazoch sú definované pamäťové miesta kde sa majú prijaté dáta uložiť. Určenie pamäťovej premennej je definované v registri M0AR. Register PAR slúži ako pinter určujúci perifériu z ktorej sa budú dáta prijímať. Register NDTR určuje počet opakovaní po prijatí dát. To znamená, že ak register nastavíme na hodnotu 200, radič DMA prijme 200 hodnôt a následne vyvolá prerušenie. Ďalšie príkazy slúžia na nastavenie veľkosti dátového slova, smer presunu dát (odosielanie/príjem), prípadne povolenie prerušení a podobne. Popis konkrétnych bitov ktoré sme nastavili je možné nájsť v referenčnom manuáli [2], alebo v publikácii ktorá tieto bity popisuje [4].

Pri nastavovaní periférie SPI je potrebné pripomenúť, že okrem základného nastavenia, ktoré sme popísali v predchádzajúcej časti článku [3], je potrebné povoliť bity na riadenie periférie radičom DMA.

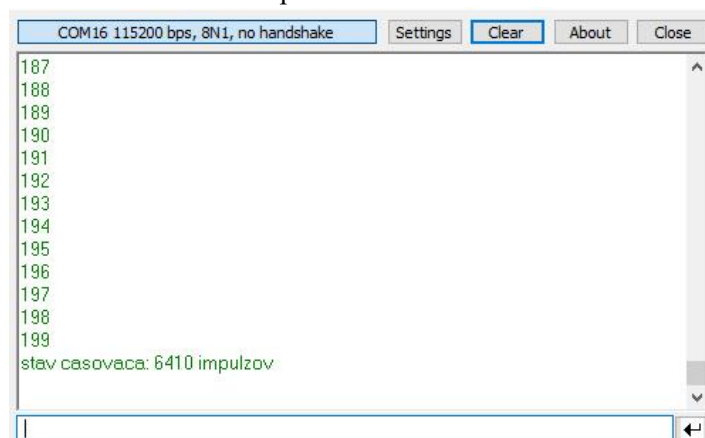
```
211 void spil_config()
212 {
213     RCC->APB2ENR |= RCC_APB2ENR_SPI1EN;    //povolenie hodin na SPI
214
215     SPI1->CR1 &= ~SPI_CR1_BR;                //baud rate f/2
216     SPI1->CR1 &= ~SPI_CR1_CPOL;             //polarita hodin v necinnom stave nastavena na LOW
217     SPI1->CR1 |= SPI_CR1_CPHA;              //data sa odosielaju na druhu nabeznu hranu
218     SPI1->CR1 &= ~SPI_CR1_BIDIMODE;        //plne duplexny mod
219     SPI1->CR1 &= ~SPI_CR1_LSBFIRST;        //most significant bit first
220     SPI1->CR1 |= SPI_CR1_SSM | SPI_CR1_SSI; //hardverovo nastaveny NSS pin pre master
221     SPI1->CR1 |= SPI_CR1_MSTR;             //nastavenie mastra
222
223     SPI1->CR2 |= SPI_CR2_DS;                //data lenght = 16 bit
224     SPI1->CR2 |= SPI_CR2_TXDMAEN;          //povolenie DMA na odosielanie dat
225     SPI1->CR2 |= SPI_CR2_RXDMAEN;          //povolenie DMA na prijem dat
226
227     SPI1->CR1 |= SPI_CR1_SPE;              //zapnutie SPI
228 }
```

Obr. 3 Nastavenie periférie SPI1 pri použití radiča DMA

Nasledujúca kapitola bude venovaná meraniu rýchlosti prenosu dát perifériou SPI1 riadenou radičom DMA. Preto poukazujeme na druhý príkaz v Obr.3, nastavenie bitu BR, na 45 MHz, čo predstavuje prenosovú rýchlosť 45 Mbit/s.

#### IV. MERANIA ČASU PRENOSU DÁT

Meranie času prenášania dát sériovou zbernicou SPI riadenou cez DMA sme realizovali pomocou 32-bitového časovača TIM2. Časovač sa spustil v momente, kedy radič DMA začal odosielať dáta zo zariadenia master do zariadenia slave. Časovač počítal do okamihu odoslania posledného 16-bitového slova. Po odoslaní dát radič DMA vygeneroval prerušenie, v ktorom sa zastavil časovač. Následne uložil svoju aktuálnu nameranú hodnotu do premennej *time* a prijaté dáta spolu s výsledným počtom hodinových impulzov odoslal na terminál do počítača.

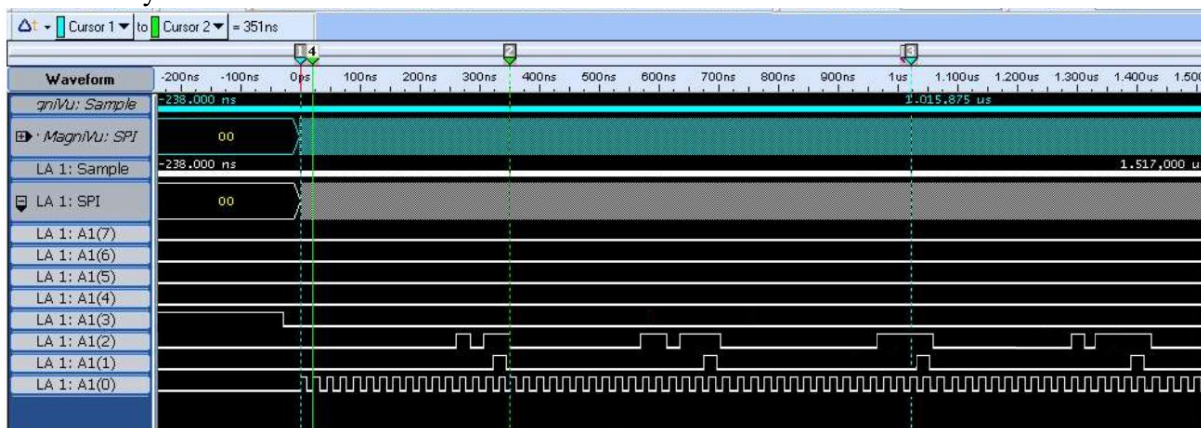


Obr. 4 Výpis prenesených dát sériovou linkou SPI a dĺžka prenášania dát

Ako môžeme vidieť na Obr. 4, časovač taktovaný frekvenciou 90 MHz napočítal 6410 hodinových impulzov. Tieto impulzy predstavujú dĺžku trvania prenosu dát (200 bitových slov) z zariadenia master do zariadenia slave. K parametrom časovača je vhodné poznamenať, že počítal s nulovou predeličkou. To znamená že výsledný čas prenosu dát určíme podľa vzorca (1).

$$\text{čas} = \frac{\text{hodnota\_CNT}}{\frac{\text{frekvencia\_časovača}}{\text{hodnota\_predeličky} + 1}} = \frac{6410}{\frac{90000000}{0 + 1}} = 71,2 \mu\text{s} \quad (1)$$

Výsledný čas prenosu dát je 71,22  $\mu\text{s}$ . Uvažujme nad teoretickým časom prenosu 200 16-bitových slov. Ak vieme, že odoslanie jedného bitového slova pri frekvencii 45 MHz trvá 355,5  $\mu\text{s}$  (pozri Obr. 5), vypočítame, že použitím radiča DMA výsledný čas prenosu 200 16-bitových slov je 71,1  $\mu\text{s}$  čo sa s našim vypočítaným časom zhoduje na desatinu mikrosekundy. Rozdiel ktorý vznikol medzi teoretickým a reálnym časom je spôsobený zaokrúhľovaním spúšťaním prerušenia radiča DMA po odoslaní všetkých dát.



Obr. 5 Logická analýza prenášania dát rozhraním SPI

Ak výsledný čas prepočítame podľa vzorca (2), dostaneme výslednú frekvenciu ktorou boli jednotlivé bity odosielané (DFF predstavuje počet bitov jedného bitového slova).

$$f = \frac{1}{t_{\text{prenosu}}} \times DFF \times N = \frac{1}{71,2 \times 10^{-6}} \times 16 \times 200 = 44,93 \text{ MHz} \quad (2)$$

## V. ZÁVER

Ako môžeme vidieť z nameraných a teoretických časov, rýchlosť odosielania dát sériovou linkou SPI s použitím radiča DMA je rapídne vyššia ako prenos rovnakého množstva dát bez použitia DMA. V predchádzajúcom článku sme uviedli, že prenos 200 16-bitových slov trval 182,32  $\mu\text{s}$ , čo je až 2,5x dlhší čas ako prenos s použitím DMA. Z toho usudzujeme, že v prípade potreby prenosu veľkých objemov dát je použitie radiča DMA v sériovej komunikácii najlepšia a najrýchlejšia voľba.

## POĎAKOVANIE

Príspevok bol pripravený na podporu slovenského grantového projektu FEI-2018-50.

## LITERATÚRA

- [1] STMicroelectronic, "STM32F446xC/E" (PDF), 2015
- [2] STMicroelectronic, "RM0090 Reference manual" (PDF), 2016
- [3] Jacko, P.: Logická analýza prenosu dát sériovou linkou SPI a aplikácia DMA, In: JIEE Časopis priemyselnej elektrotechniky. Roč.2, č.4 (2018), s. 33-38, ISSN: 2454-0900
- [4] The DMA controller description and configuration of the STM32 microcontrollers / Patrik Jacko, Dobroslav Kováč - 2018. In: Časopis priemyselnej elektrotechniky. Roč. 2, č. 1 (2018), s. 83-88. - ISSN 2454-0900