JIEE Časopis priemyselnej elektrotechniky
Journal of Industrial Electrical Engineering
ISSN 2454-0900
www.jiee.eu

The STM microcontrollers clock synchronization using by three Nucleo boards

[1]*Patrik JACKO*, [2]*Dobroslav KOVÁČ*

[1,2] Department of Theoretical and Industrial Electrical Engineering, FEI TU of Košice, Slovak Republic

[1] patrik.jacko.2@tuke.sk, [2] dobroslav.kovac@tuke.sk

***Abstract* — Many devices operate with two or more microprocessors or microcontrollers, where the each chip has a specific function. For example, one microcontrollers scans the input signal using by AD converters, the others microcontroller process and shows output data getting from another microchips. There must be insureed synchronization between microcontrollers for correctly communication and synchronization. This article describes clock synchronization of STM32F4 microcontrollers implemented on the Nucleo-64 development board, where one microcontroller operates as master and the other microcontrollers operate as slave.**

***Keywords* — clock synchronization, Nucleo64 board, STM32F446RE, STM32 microcontroller**

I. Introduction

The synchronization of two or more microprocessors included in the one device is the primary factor for the correctly device function. If there isn't synchronization, device can wrong work, or there can be a lot of errors. Each processor, microprocessor or microcontroller has to clocked by some device which generates clock signals. Usually it can be pulsed by a clock signal generators or oscillators. Of course, some microcontrollers includes internal oscillators too, therefore if there is used an internal oscillator to the processor clocking, it is necessary not connecting the external oscillator or pulse generator. In an some device can be "master-slave" structure, where master microcontroller clocks other slave devices. In this article we will be describing synchronization of three STM32F446RE microcontrollers of the Nucleo-64 development board.

II. The STM32F4 microcontroller and NUCLEO-64 development board

The Nucleo-64 board is a development tool for the educational programming with STM32 microcontrollers or developing prototypes and own projects. This board includes microcontrollers of the STMicroelectronics Company, ST-Link/V2-1 debugger, oscillators, user button and LED diodes.

Fig. 1 Nucleo-64 board

The Nucleo-64 board offers four high-speed clock possibilities:

- Clocking from ST-LINK debugger oscillator- In the Fig. 2 is a 8 MHz oscillator shown in the upper part of the board (num. 1). This oscillator clocks debugger processor and microcontroller STM32F4 too. In this case the frequency of the input clock signal cannot be changed, it is fixed at 8 MHz and connected to the OSC_IN (PH0 pin). [1]

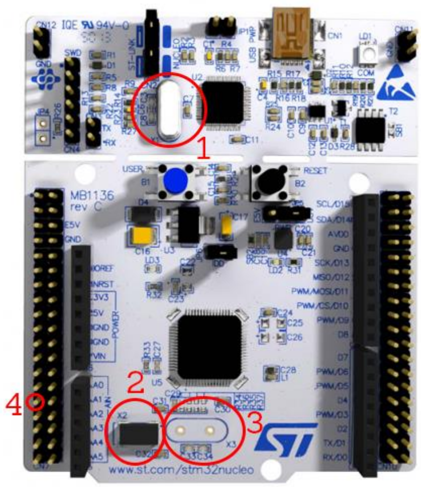

Fig. 2 Oscillators of Nucleo-64 board

- Microcontroller STM32F446RE can be clocked by HSE (High-speed External oscillator) that is marked by num.3. In this case the user can connect his own crystal. In the default state crystal is not provided. There is necessary abides minimum and maximum range of frequency: the minimal frequency is 4 MHz and maximal frequency is 26 MHz crystal. [1]

- It is possible clocks the microcontroller by an oscillator from an external device. In this case user can connect an external clock source to the pin PH0 (num. 4 in the Fig. 2).

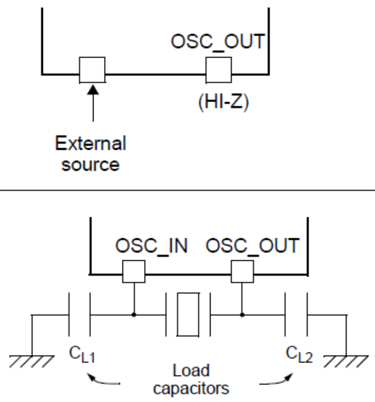

Fig. 3 Using the crystal oscillator (down) and external clock source (up) [3]

- Using by internal high-speed oscillator, that operates in 16MHz frequency. The microcontrollers includes this oscillator in the packet.

So, microcontroller STM32F446RE implemented on the Nucleo-64 board has these possibilities of the high-speed clocking system. The Fig. 3 shows the external clock source connecting, where is used only one pin (PH0 – num. 4 in the Fig. 2) and crystal oscillator connecting by using crystal and capacitors. The crystal is connected to the OSC_IN (PH0 pin) and OSC_OUT (PH1 pin). There is possible clocking by a low-speed oscillator (LSE). The Nucleo-64 board has two low-speed oscillators – Internal low-speed (LSI) with 32 kHz frequency implemented in the microcontroller and External low-speed oscillator (LSE) with 32,678 kHz on the Nucelo-64 board (in the Fig. 2 num.2). [3]

Thanks the PLL (Phase Lock Loop) programmer can clocks microcontrollers from 8 MHz up to 180 MHz. There are five PLL that divides or multiply the input clock signal. The PLL supports 0.95 MHz to 2.10MHz input clock frequency. In the Fig. 4 you can see dividing and multiplying of the input clock signal. Therefore PLL input signal is limited of 0.95 MHz to 2.10 MHz, the input signal is divided into 4 by PLLM. Then the incoming PLL clock signal is 2 MHz. After the multiplying by PLLN and dividing by PLLP, a microcontroller is clocked to 180 MHz. [2]

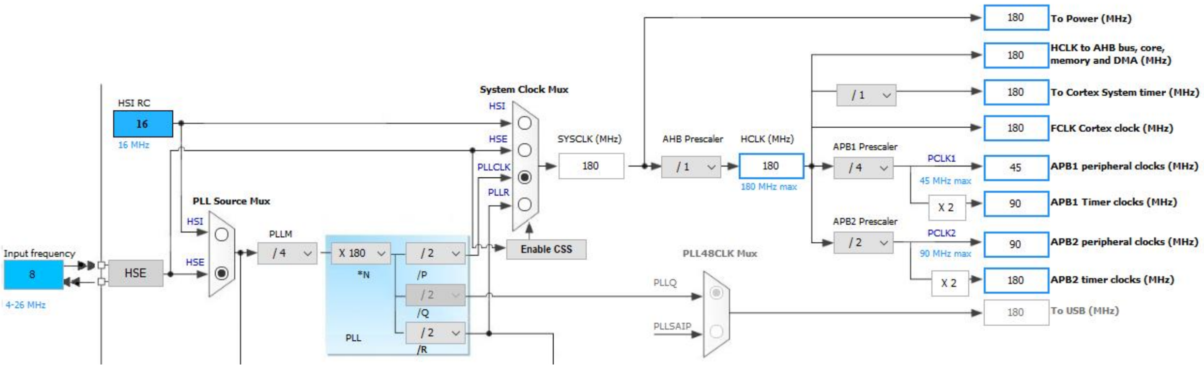

Fig. 4 Microcontroller clocking from external 8 MHz crystal

It is worth noting that microcontroller STM32F446RE operates with various frequencies for groups of peripherals. As you can see in the Fig. 4, the peripherals connected to the APB1 bus operates on the 45 MHz frequency, but its timers operates on the 90 MHz frequency. The peripherals connected to the APB2 operates on the 90 MHz frequency and its timers operates on the 180 MHz frequency.

III. NUCLEO-64 BOARDS SYNCHRONIZATION

The STM32F446RE microcontroller supports two clock-out capability named the MCO – microcontroller clock output:

- MCO1 – the programmer can output four different clock sources onto the MCO1 pin (PA8)
    - HIS clock
    - LSE clock
    - HSE clock
    - PLL clock

- MCO2 – the programmer can output four different clock sources onto the MCO1 pin (PC9)
    - HSE clock
    - PLL clock
    - System clock (SYSCLK)
    - PLLI2S clock

It is necessary to set the registers for clock output. We will be working with MCO1, where we will watch the HSE oscillator clock in the output. In the Fig. 5 is shown source code for MCO configuration applied to the PA8 pin.

```
void mco_config()
{
  RCC->AHB1ENR |= RCC_AHB1ENR_GPIOAEN;  //enable clock to the GPIOA
  GPIOA->MODER |= GPIO_MODER_MODER8_1;  //set the alternative function to the PA8

  RCC->CFGR |= RCC_CFGR_MCO1_1;   //HSE oscillator clock selected
  RCC->CFGR &= ~RCC_CFGR_MCO1PRE; //prescaler = 0;
}
```

Fig. 5 MCO1 configuration

Such as was mentioned in the introduction, we will be synchronizing three STM32F446RE microcontrollers implemented to the Nucleo-64 development board. In the first part we show three clock cycles of STM32F446RE microcontroller's where not synchronization is. The each

ISSN 2454-0900

microcontroller is clocked by own crystal. The Fig. 6 shows three 8 MHz signals of three microcontrollers.

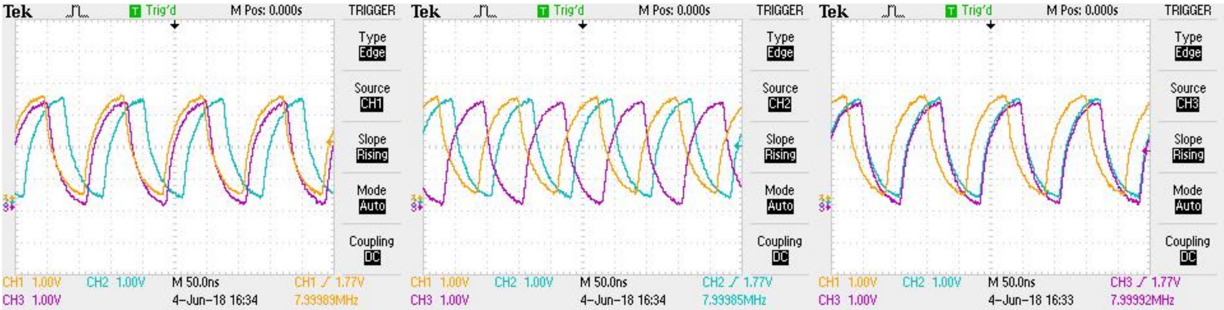

Fig. 6 Clock frequency of three Nucleo-64 board

In the Fig. 6 we can see oscilloscope measurement of three Nucleo-64 board's clock output. In this case, each Nucleo-64 board has own 8 MHz oscillator. The oscilloscope offers pressure measurement of the oscillator frequency. The first Nucleo-64 board was connected to the channel 1, where frequency is 7.99989 MHz. The second Nucleo-64 board was connected to the channel 2. In this case is frequency 7.99985 MHz and the last Nucleo-64 board connected to the channel 3 has frequency 7.99992 MHz. We can see also that output signals are not synchronized because Nucleo-64 board's frequency is different.

Since the Nucleo-64 board's frequency different, it is impossible works if the applications require exact synchronization of the microcontrollers.

Therefore, we use three Nucleo-64 board where one board will be present as a master device, which will be clocked source of the other two boards that will be present as slave devices.

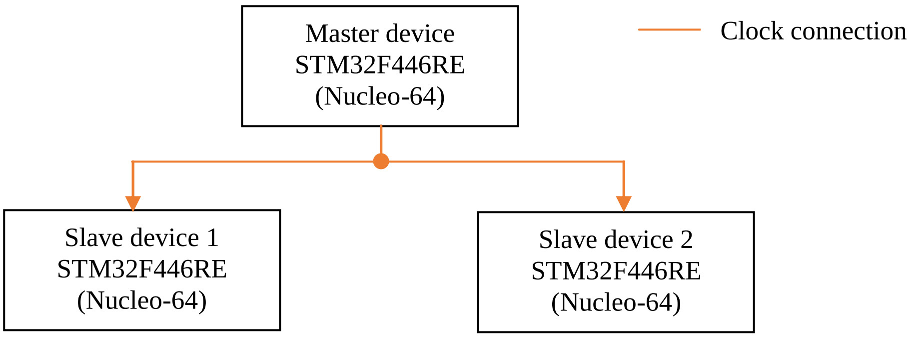

Fig. 7 Block scheme of Nucleo-64 boards

If we have Nucleo boards connected according Fig. 8, it is possible synchronize clocks of the Nucleo-64 boards, where Slave 1 and Slave 2 will be clock such as Master Nucleo-64. In the slave case, it is necessary enable bypass for external clock source. In the master case this step is not necessary, but there is requested the HSE enabling.

```
void clocks_config()
{
  RCC->CR |= ((uint32_t)RCC_CR_HSEON | RCC_CR_HSEBYP);  //HSE and BYPASS enable
  while((RCC->CR & RCC_CR_HSERDY)==0);                  //wait for HSE ready
  RCC->CFGR = RCC_CFGR_SW_HSE;                          //HSE is system clock
  while((RCC->CFGR & RCC_CFGR_SWS) != RCC_CFGR_SWS_HSE);//wait for system clock HSE
}
```

Fig. 8 Slave devices clock configuration

In the first command there is sets enabling HSE (high-speed external clock) and bypass for connecting external source. The condition *while* in the second line uses for waiting as far as will be HSE ready. In these lines we configure the CR register (clock-control register). The third line sets HSE as system clock by a switch and the last command waits until HSE will be set to the system clock. There are also a lot of configurations too, but the primary commands for clock source are shown in the figure.

If we use the Nucleo-64 board as a slave device, where the other device, or other source is the clock source for the Nucleo-64 board, it is necessary solders the resistor from SB50 to the SB55 on the bottom of board. This configuration is very important. [1]

After the configuration there is possible measurement output clock signal from the all Nucleo -64 boards.

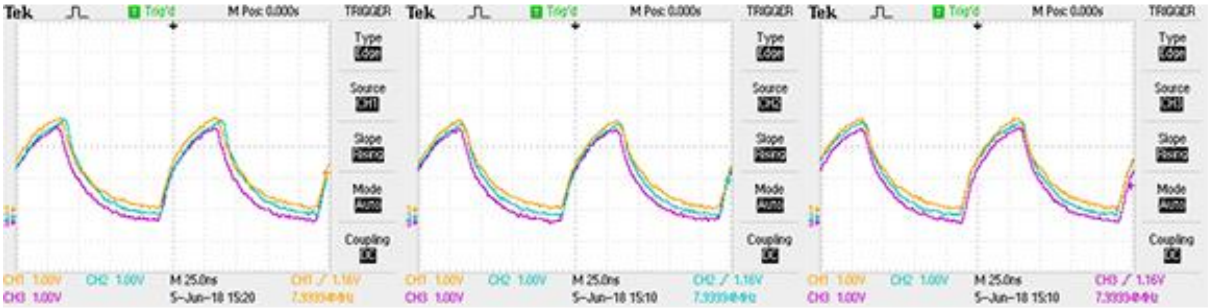

Fig. 9 Synchronized clock output from the three Nucleo-64 boars

In the Fig. 9 are shown output clock signals of three Nucleo-64 boards. You can see that each clock signals (oscilloscope) channel has 7.99994 MHz and all three signals are precise synchronized.

IV. CONCLUSION

The processor, microprocessor or microcontroller synchronization is a very important part of each device where two or more chips operate together. In this article we were described STM32F446RE synchronization. The results are shown in the figure where we can see that synchronization is correctly. If synchronization is set between microcontrollers, we can make a lot of applications with them, for example: fast synchronization input signal by AD converters of two or mode STM32 microcontrollers.

ACKNOWLEDGMENT

We support research activities in Slovakia / Project is co-financed from EU funds. This paper was developed within the Project "Centre of Excellence of the Integrated Research & Exploitation the Advanced Materials and Technologies in the Automotive Electronics ", ITMS 26220120055

REFERENCES

[1] STMicroelectronics, *User manual – STM32 Nucleo-64 boards*, [online], <http://www.st.com/en/microcontrollers/stm32f446re.html>
[2] STMicroelectronics, *STM32F446xC/E Datasheet*, [online], <http://www.st.com/en/microcontrollers/stm32f446re.html>
[3] STMicroelectronic, RM0090 Reference manual, [online], <http://www.st.com/content/ccc/resource/technical/document/reference_manual/4d/ed/bc/89/b5/70/40/dc/DM00135183.pdf/files/DM00135183.pdf/jcr:content/translations/en.DM00135183.pdf>
[4] Guzan M., Špaldonová D., Hodulíková A., Tomčíková I., Gladyr A.: *Boundary Surface and Load Plane of the Ternary Memory*, In: Electromechanical and energy saving systems. Vol. 15, no. 3 (2011), p. 163-167. - ISSN 2072–2052
[5] Dziak,J. : *Linear circuit simulation using MATLAB and modeling of nonlinear elements*, In: SCYR 2014 Proceeding from Comference: 20.5.2014: Herľany, S. 70 - 71, Košice : Technická univerzita v Košiciach, 2014 /978-80-553-1714-4/.
[6] Kováčová I.,: *EMC of power DC electrical drives* - 2005. In: Journal of Electrical Engineering. Vol. 5, no. 1 (2005), p. 61-66. - ISSN 1582-4594
[7] Vince T., Molnár J., Bučko R.: *Real-time regulation systems based on internet - optimization algorithm* - 2010. In: Transactions of KMOSU. Vol. 4, no. 3 (2010), p. 150-153. - ISSN 2072-8263
[8] Bereš M.: *Vstupné a výstupné charakteristiky znižovaco- zvyšovacích impulzových DC-DC meničov* - 2016. In: Electrical Engineering and Informatics 7 : proceedings of the Faculty of Electrical Engineering and Informatics of the Technical University of Košice. - Košice : FEI TU, 2016 S. 793-796. - ISBN 978-80-553-2599-6