Dariusz KANIA
Wrocław University of Technology, Chair of Electronic and Photonic Metrology

An universal USB module for programming digital signal processors presented on the example of TI DSK6711 and TI DSK6713 modules

***Abstract**. This article presents a device that allows loading a program into a signal processor's memory and its execution without the need to use a computer and a specialized environment. The device allows to program a DSP with a file from a USB stick memory and exchange data from the DSP with the computer through the USB interface. The device can be applied in all industries where microprocessors are used and where the access to a computer or expensive programming environments is limited. The benefits of the device are as follows: faster action, no software license necessary, smaller size of the equipment (greater mobility), universal compatibility with different environments and interfaces, real-time processing, easy manufacturing and configuration.*

Streszczenie.** W artykule zaprezentowano urządzenie, które umożliwia załadowanie programu do pamięci procesora sygnałowego i jego wykonywanie bez potrzeby używania komputera i specjalistycznego środowiska. Urządzenie umożliwia programowanie procesora plikiem z pamięci USB oraz wymianę danych procesora z komputerem poprzez interfejs USB. Urządzenie może być wykorzystane w każdej gałęzi przemysłu, gdzie są wykorzystane mikroprocesory, a szczególnie w warunkach, gdzie nie zawsze możliwy jest dostęp do komputera czy drogich środowisk programistycznych. Korzyści z zastosowania urządzenia to przyspieszony czas działania, brak konieczności zakupu licencji na oprogramowanie, mniejsze gabaryty używanego sprzętu (większa mobilność), uniwersalność współpracy z różnymi środowiskami i interfejsami, praca w czasie rzeczywistym, łatwość wykonania i konfiguracji. **Uniwersalny moduł USB do programowania procesorów sygnałowych na przykładzie modułów TI DSK6711 i TI DSK6713.

Keywords: USB interface, DSP, data transmission, digital signal processing.
Słowa kluczowe: interfejs USB, procesor sygnałowy, transmisja danych, cyfrowe przetwarzanie sygnałów.

doi:10.12915/pe.2014.06.38

Introduction

Microprocessors are found in every area of life. Their popularity is due to the relatively low prices and very high capacities. Digital Signal Processors (DSP) are an interesting subcategory of microprocessors. Their architecture is optimized for signal processing [1]. They are often used in applications such as processing of video stream (for example face recognition, Haar's algorithm), audio (for example speech recognition) or medical signals (filtration and processing signals from sensors on the human body).

TI (Texas Instruments) is one of the market leaders in the production of digital signal processors. Its products are known throughout the world and are often used in academic institutions and in the industry. TMS320C671x is one of the most famous generations of DSPs. Processors of this family are available with a DSK (DSP Starter Kit) module. Launching applications is carried out using development tools and hardware. In order to program such processors, the computer should be connected to the module via a USB ('C6713) or LPT ('C6711) cable and CCS environment should be started.

TMS320C674x is a recent family of DSP made by TI. These DSPs are implemented, among others, in OMAP processors (together with an ARM core) and are often used in multimedia systems. Unfortunately, in this case appropriate software and a programmer is also needed to run an application.

Loading a program into the processor memory in both cases is done via a computer. This approach significantly limits the use of evaluation modules and digital signal processors. It is adequate only in laboratory conditions. Of course notebooks can be used, but in some cases carrying all the equipment around is inconvenient and often ineffective.

This article presents the design and structure of a device that was designed eliminate the inconveniences such as the need to use a computer together with an adequate development environment for loading a program into the processor's memory. The overall concept assumes that, after turning on the power, the processor's memory is automatically programmed and the program is executed. The files with the program should be located on a USB stick. In addition the device enables adequate communication between the signal processor and the computer via the USB interface in real time. In the absence of a connected computer, the data can be stored in USB memory and processed later. The device was tested on DSK (Development Starter Kit) modules produced by Texas Instruments, with processors: TMS320C6711 and TMS320C6713.

The article is based on a thesis [2]. Section 2 presents the concept of operation of the USB module and the characteristics of the used hardware. Section 3 is a technical description of the USB module. Section 4 includes information about the software of the device. Sections 5 and 6 present the data exchange capabilities using the USB module. Section 7 includes the conclusions and the summary of operation of the presented USB module.

The concept of operation of the USB module

The USB module was designed to allow signal processors run applications from a passive external storage, without using a development system. In addition, the module has to enable communication between the signal processor and a computer via a USB interface. The main problems of the design and structure of the device are as follows: proper matching of the USB module and the module with the signal processor (PCB shape, matching the data exchange interfaces), choosing the right processor on which the USB module is based, the software for both the DSP and the processor on the USB module. A DSP Starter Kit from Texas Instruments and Vinculum-II (VNC2) FTDI processor were used to implement the functionality of the module.

The DSK (DSP Starter Kit) module is a starter kit that allows to develop applications for Texas Instruments signal processors [3]-[8]. The DSK modules used in this research include a TMS320C6713 (DSK6713) and TMS320C6711 (DSK6711) processor. Both modules have the ability to choose the type of boot. The processors communicate with peripherals via the 32-bit External Memory Interface (EMIF), which allows to connect to the processor external devices, such as the data and program memory [9], [10].

The 'C67x processor family has large memory space. The internal RAM is located starting from the address 0x0 and can be used either as program or data memory. There are four distinct regions called CE0 - CE3 (chip enable) available via the EMIF. CE0 area is designed for SDRAM area, CE1 area is shared between the flash memory and CPLD. The flash memory is used as a boot option.

On the modules there are two connectors. On one of them (Memory I/F) there are data traces, address traces, control traces to connect the external memory to the processor via the EMIF. 3.3V and 5V power supply traces are also included.

The Vinculum-II processor is the second generation of USB Hosts by FTDI. The processor has the ability to handle mass memory with FAT file system, it has two USB 2.0 interfaces and an 8-bit FIFO (First In First Out) interface. The free integrated development environment VinIDE [11], [12] is also a key advantage.

The USB module uses CE2 area for external memory to communicate with the digital signal processor (via the EMIF). The VNC2 processor, on which the device is based, uses FIFO (First In First Out) interface in asynchronous mode to exchange data with the DSK module. In addition, the module has the ability to exchange data with USB flash memory via the USB 2.0 interface and with a computer via the virtual COM port.

PCB of the device

The following were used to connect the device with the DSK module: an integrated circuit with NOR gates, a circuit with NOT gates, and one system with three-state buffers (fig. 1). Two data traces, D8 and D9, enable software verification of the possibility of writing/reading the data into and from the memory at the given time (protection against forbidden operation). TXE# and RXF# traces are set into low state if the data is ready for reading or writing from the FIFO buffer. Both of these signals are buffered and their switching depends on the logical sum of CE2# and ARE# traces. CE2# signal is low during reading or writing to the memory of the signal processor, while ARE# signal is low during reading from the FIFO buffer. A4 trace allows to select whether the data should be read from the data traces D [9:0] or only from data traces D8 and D9. AWE# signal passes through two NOT gates. This is the easiest way to delay the signal by about 15ns. This was necessary for proper operation of the device on both DSK modules. The processor can work simultaneously in Host mode (receive data from the memory stick) or in Device mode (data exchange with a computer).

The USB module is designed to be able to work on both DSK modules (fig. 2-3). Most of the elements are located on the bottom layer (no conflict with extensions cards for the DSK). The USB B connector is used to connect a memory stick, and the mini-USB is used to connect a computer. The holes on the board provide stability. The dimensions of the board are 52.20 mm × 80.00 mm.

Software for the USB module

The procedure for starting the DSK modules with 'C6711/13 is as follows: 1kB of code (booting code) from the address 0x90000000 is copied into RAM address from 0x000 to 0x400. Then, the processor begins executing the code starting at the address 0x000. The purpose is to perform basic processor configuration and testing of processor peripherals. Then, the processor jumps to the address "_int_00" and waits to be programmed by the Code Composer Studio environment.

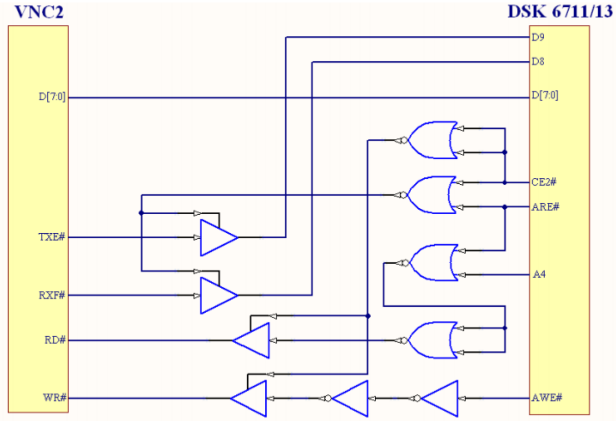

Fig. 1. The logic of the connection of the VNC2 to the DSK 6711/13 module

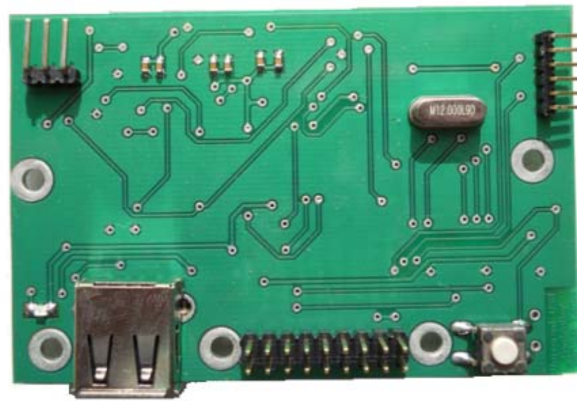

Fig. 2. The top layer of the designed USB module

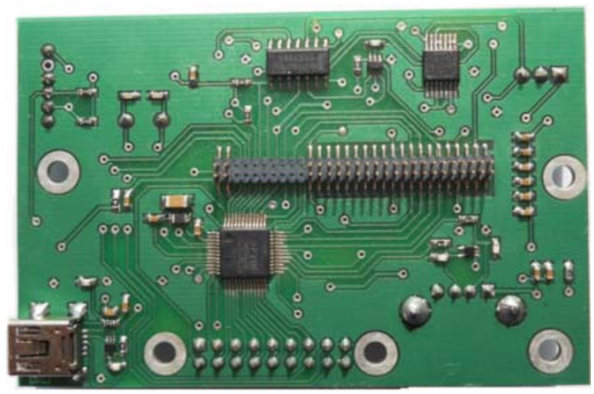

Fig. 3. The bottom layer of the designed USB module

This solution is inconvenient when the application is to be launched in less typical, non-laboratory locations. The universal USB module solves this problem. For proper operation of the device it is necessary to adequately program the VNC2 processor and the signal processor to enable proper communication.

A program written for the Vinculum-II processor works as follows: when a command to read a file from the external memory arrives, at first it is checked whether it relates to a file with the memory map of the signal processor or a file with a program. Then, the processor reads from the USB stick the specific file and begins sending it to the signal processor.

The USB stick hosts files from each project compiled in the CCS: the binary file with the program and the **.map* file containing the position of the sections in the signal processor memory. There are three switches on the DSK modules. One of them is used to select the boot options, the remaining two can be used to choose the right program number to be loaded to the digital signal processor. While the file with the memory map is (by default) automatically created by the CCS environment, the file with the program must be appropriately prepared. It cannot contain the so-called debug symbols generated during the compilation of the project. In this case they would be interpreted as program code. Thus, the file with the compilation outcome (*.out*) is converted to a binary file. The **.map* file contains the names of the sections, their source addresses, destination addresses, and the so-called "entry point" address. This address must be obtained in the main program (for the digital signal processor). For this purpose the initial bytes of this file are copied (the appropriate command is sent to the VNC2). The numbers of bytes that store the address vary depending on the program name. After obtaining the input address, the entire binary file with the program in the USB stick is copied to the program by the VNC2. The data is stored in the CE0 area of the external memory of the processor. With the file program in the memory and having the input address to the program, it is now sufficient to just make a jump to this address. After making the jump, the program from the mass memory is executed.

After programming modification and using of the USB module, the startup procedure of the DSK module is as follows:

1. Copying 1 kB of code from the 0x90000000 address to the RAM address from 0x000 to 0x400.
2. Start of the execution of the booting program – basic configuration of the processor.
3. Choosing whether the further procedure should be standard or carried out by using the USB module.
4. If the second option is chosen, the tests are carried out and the presence of a USB stick in the USB connector is checked. If not, the device waits for the stick to be connected.
5. On the basis of the switches settings on the DSK module two adequate project-specific files are copied (the binary file and the map file).
6. After copying the files to the memory, the processor jumps to the program's start address and starts executing it.

To perform the presented procedure, the Flash memory of the signal processor, which contains the booting code written in assembler and C should be adequately programmed. The code should be appropriately modified beforehand.

Testing the device

The aim of the study was to confirm the correct operation of the device or to find errors. The way the data is transferred from the USB module to the computer and the signal processor, and received via the USB module from the computer or the digital signal processor, was examined. A logical states analyzer was used to analyze the flow of data. The impact of the changes in the parameters of the control signals for the CE2 area (using DSP/BIOS) on the operation of the entire system (fig. 4-5) was tested. This way the minimum values of the parameters were obtained at which the system works correctly on both DSK modules. The circles with arrows on WR# and RD# traces represent the relation between data writing and readout on the edges of the signals. The circles on the CE2 signal represent the programming verification of the ability to read or write.

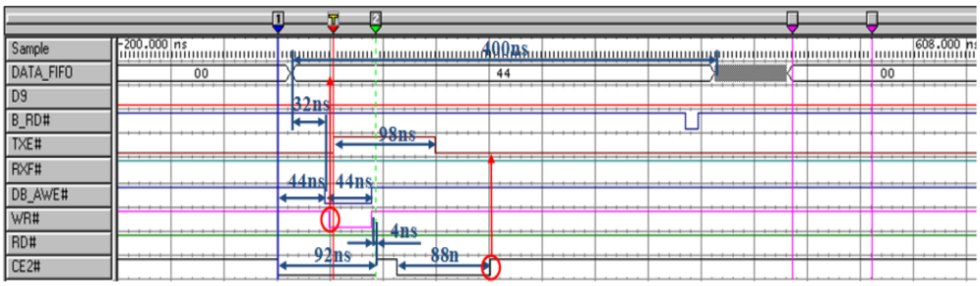

Fig. 4. An example of writing two numbers for DSK 6711

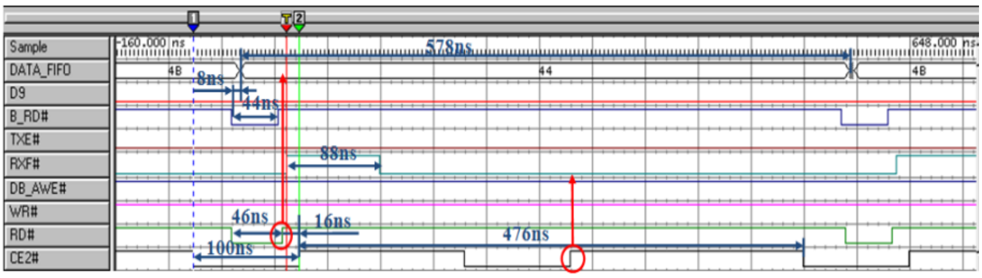

Fig. 5. An example of reading two numbers for DSK6713

Data exchange through the USB module

One of the possible applications of the device is real-time sending of the results of digital filtering from a signal processor to a computer (Matlab) using a USB cable. The data is received via a virtual COM port, the transmission speed can be selected in the port settings.

The main limitation of the speed of data transfer via the USB module are the minimum times of the VNC2 signals required for proper operation of the program. The minimum time between two readouts or writings, or between a readout and a writing is 100 ns. Our analysis, carried out using a state analyzer, shows that, for the DSK6711 module, the shortest time between two consecutive writings of a byte of data is about 1.5 μs, whereas between two successive readouts it is 1.75 μs. For the DSK6713 module those times are approximately 400 ns for writing and about 500 ns for reading. Shorter times are a result of the difference in the clocking frequency of the processor cores' signals. Another limitation is the low rate of data sending via the virtual COM port (in real terms about 2 Mb/s). The data sent from the processor to the computer also goes through the VNC2, which generates additional delay (the duration of one cycle is about 21 ns). The average writing speed for the DSK6711 is about 200 kB/s, and the readout speed is about 170 kB/s. For the DSK6713 module these times are approx. 218 kB/s for writing and approx. 213 kB/s for readout.

Conclusions

The Universal USB module presented in this article and based on the FTDI Vinculum-II processor allows to achieve some degree of independence from the Code Composer Studio environment or even from turning on the computer. It allows to program digital signal processors using files on a memory stick. In addition, the module enables the exchange of data between the DSP and the computer, eliminating the need to use other (often proprietary) interfaces. With the module it is possible, for example, to send data from the digital filters in real time.

The device is easy to manufacture. It does not require specialized technologies or procedures. It can be easily adapted to other modules and processors. It allows to send data in real time and to monitor it at the same time, without the need to stop the processor (as is the case for the CCS). It is not necessary to change the DSK hardware in order to connect the module. To transfer the data to a computer it suffices to properly configure the serial port. This enables compatibility with different environments and programs that support it. It is also possible to write the data onto a USB stick and then download it to a computer at home, after the experiments. The disadvantages of the device may include the necessity to build an electronic circuit or to purchase a USB cable. The transmission speed is limited mainly by the processing speed of the VNC2 processor. For DSK6711 average speed of data readout is 170 kB/s, 200 kB/s for writing. For DSK6713 these speeds are respectively 213 kB/s and 218 kB/s.

REFERENCES

[1] Smith S.W., *The Scientist and Engineer's Guide to Digital Signal Processing Getting Started with DSPs*, Second Edition, California Technical Publishing, San Diego 1999.
[2] D. Kania, The Algorithms of Digital Filtering for Signal Processors, master's thesis, Wrocław University of Technology, 2011.
[3] Spectrum Digital, DSP Starter Kit (DSK) for the TMS320C6713, 2005.
[4] Texas Instruments, TMS320C6713 DSK Technical Reference 506735-0001 Rev. B, November 2003.
[5] Texas Instruments, TMS320C6713 DSK, Document Number 506732, 2003.
[6] Texas Instruments, TMS320C6711 DSP Starter Kit, Document Number D600982.
[7] Texas Instruments, Code Composer Studio and the DSK6713, presentation in Power Point.
[8] Texas Instruments, Chapter 3 "DSK Hardware", presentation in Power Point.
[9] Texas Instruments, TMS320C6000 DSP External Memory Interface (EMIF) Reference Guide, Literature Number: SPRU266E, April 2008.
[10] Texas Instruments, TMS320C6000 EMIF to External Asynchronous SRAM Interface, 1999.
[11] FTDI, User Manual AN_151 Vinculum II User Guide, Version 1.2.2, 2010.
[12] FTDI, Vinculum Firmware Tools User Manual Version: 1.0, 2007.

***Autor**: mgr inż. Dariusz Kania, Politechnika Wrocławska, Katedra Metrologii Elektronicznej i Fotonicznej, ul. B. Prusa 53/55, 50-317 Wrocław, E-mail: Dariusz.Kania@pwr.wroc.pl.*