

## MIDI CONTROLLED AUDIO-DSP SYSTEM

Piotr BOBIŃSKI, Bartosz BIELAWSKI

Warsaw University of Technology  
Institute of Radioelectronics  
Nowowiejska 15/19, 00-665 Warszawa, Poland  
e-mail: pbo@ire.pw.edu.pl

*(received June 15, 2007; accepted November 30, 2007)*

The paper presents a MIDI controlled hardware station for digital audio signal processing. The system consists of an evaluation board ADSP-21364 EZ-KIT with the SHARC-21364 processor, and a MIDI to the SPI converter designed for the communication with a DSP processor. Information about hardware and protocols is included. The design of the converter device and the implemented software modules that enable to control the applications running on the DSP processor are described. Finally, two applications of the developed systems are introduced: synthesis of music with steering from a MIDI keyboard, and realization of recording with a control from a MIDI console.

**Keywords:** MIDI, SPI, audio DSP, SHARC processor.

### 1. Introduction

Digital processing of audio signals is one of the fastest developing DSP domains. The need to achieve the best quality and the real time processing of audio signals requires specialized hardware to be used in audio DSP systems. The Analog Devices' SHARC processors dominate the floating-point Digital Signal Processing market, delivering exceptional core and memory performance complemented by outstanding I/O throughput. SHARC family makes floating-point processing economical for applications where dynamic range is a key performance such as home and car audio, medical, industrial and instrumentation products. The ADSP-21364 offers the highest speed performance – 300 MHz/1800 MFLOPs – within the third generation SHARC Processor family [2]. This level of performance makes the ADSP-21364 particularly well suited to address the increasing requirements of the professional audio market segment. High core performance combined with audio centric peripherals including a high-precision, 8-channel asynchronous sample rate converter makes the ADSP-21364 a very good choice for the applications requiring industry-leading equalization, reverberation, and other audio effects processing.

The ADSP-21364 EZ-KIT Lite provides developers with a cost-effective method for initial evaluation of the ADSP-21364 SHARC Processor architecture via a USB-based, PC-hosted tool set [1]. With this EZ-KIT Lite users can learn more about Analog Devices' ADSP-21364 hardware and software developments and quickly prototyping platform for a wide range of applications. The main disadvantage of using such an evaluation board in both educational and professional applications is its limited capability of controlling developed algorithms in real time. Four on-board push buttons are in most cases not enough for flexible control of applications' parameters during runtime. Therefore the idea of developing a custom control interface for the evaluation board has appeared. To make this interface more flexible from the user's point of view, the MIDI (Musical Instrument Digital Interface) standard was chosen as a well known, industry-standard that enables electronic musical instruments, computers and other equipment to communicate, control and synchronize with each other in real time [3, 4]. On the other side (the side of a controlled device) the SPI (Serial Peripheral Interface) has been used, that allows controlling not only this particular board, but also the variety of devices with such an interface [7].

The main aim is to provide easy to setup and use interface that separates the developer from thinking of unnecessary low-level details. It is also required, that the system works in real time with smallest possible delays, allowing uninterrupted focus on subject of study. Therefore this system is perfect for development stage of audio-DSP systems, when there's no user interface ready.

The paper is organized as follows. The hardware specification for both the evaluation board and the developed interface converter is presented in section 2. Details of the developed converter device are given in section 3. Examples of applications of the whole system are shown in section 4. Conclusions are listed in the last section of this paper.

## 2. Hardware specification

The audio-DSP system consists of three main parts: the ADSP-21364 EZ-KIT Lite evaluation board, the MIDI to SPI converter, and the steering device, which could be any kind of the MIDI device, depending on application's requirements.

The EZ-KIT Lite package includes the ADSP-21364 SHARC Processor desktop evaluation board along with an evaluation suite of the Visual DSP++ development and debugging environment including the C/C++ compiler, assembler, and linker. Except for the SHARC processor, the board includes 1 M × 8-bit FLASH memory, 512 K × 8-bit SRAM, 512 Kbit SPI FLASH memory, AD1835 stereo, 96 kHz, 24-bit codec, the USB-based debugger interface [1]. The main features of the ADSP-21364 SHARC processor are listed below [2]:

- 300 MHz /1.8 GFLOPs SIMD SHARC core supporting IEEE 32-bit floating-point, 40-bit floating-point and 32-bit fixed-point data types,
- 3 Mbits SRAM,
- 23 zero-overhead DMA channels,
- High-precision 8-channel asynchronous sample rate converter,

- Digital Audio Interface (DAI) enabling user-definable access to peripherals including an input data port (IDP) and general purpose I/O,
- 6 serial ports supporting I2S, left-justified sample pair, and TDM modes,
- 2 SPI-compatible ports supporting master and slave modes,
- 16 Pulse Width Modulation (PWM) channels.

The MIDI to SPI converter is built on a single board and contains several blocks where the most important is the Atmel's AT91SAM7 microcontroller that uses the ARM7 core [5]. It is a fast and cheap  $\mu$ C with integrated RAM and ROM, and rich peripherals capable of controlling the whole system in real time. Main features of this  $\mu$ C are:

- 32 bit RISC core capable of running at up to 60 MHz,
- 64 kB-integrated flash memory in-system programmable,
- 16 kB-integrated RAM,
- 3 UARTs, SPI, USB, Timers/Counters,
- 1.8 V integrated regulator for core.

More details about the MIDI to SPI converter are given in the next section, where both the hardware and software are described.

### 3. MIDI-SPI converter

#### 3.1. Hardware description

The Main double-sided board of the device contains all blocks and peripherals required for the system to work. This includes the power supply block, the MIDI, the Extended SPI and RS232C interfaces, device status LED indicators, and the extension slot which can be used for optional display, keyboard and MCU – the Atmel's AT91SAM7. The complete converter device is presented in Fig. 1.

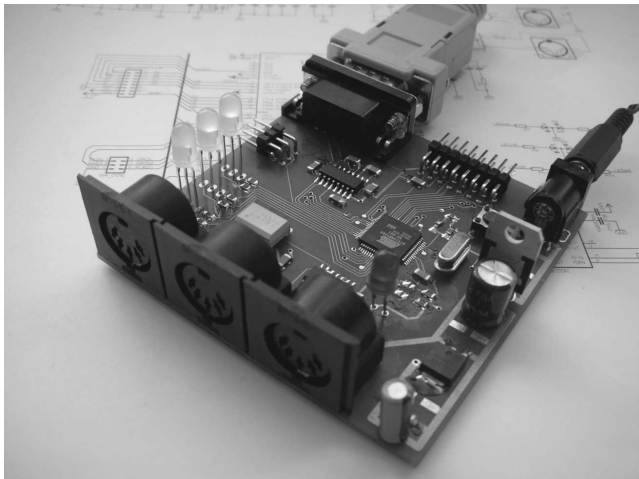


Fig. 1. MIDI to SPI converter.

The power supply block consists of two linear regulators that supply voltages of +5 V and +3.3 V, protection diodes against wrong polarity and filtering capacitors. The voltage of +1.8 V required by the MCU core is provided by a regulator built into the  $\mu$ C chip. According to standard design rules, 100 nF filtering capacitors are used all over the PCB to reduce noise generated by all digital ICs.

The MIDI interface includes three DIN-5 sockets:

- MIDI IN for incoming data,
- MIDI THRU to enable daisy-chaining of MIDI devices,
- MIDI OUT for sent data.

The signal from the MIDI IN socket is fed directly through a series resistor to the 6N138 optocoupler translating current signals to voltage signals, which are understandable by MCU. The input socket has no common pins with the device's ground to eliminate noise spreading across the MIDI. MIDI OUT and MIDI THRU are driven by the 7400 quad NAND gate working as an amplifier. The difference is that MIDI THRU uses signals from MIDI IN while MIDI OUT uses signals directly from MCU.

The second main interface in the MIDI to SPI converter is the Extended SPI. The pin layout is exactly the same, as on the ADSP-21364 DSP board. While a standard SPI consists of 4 signal lines (MISO, MOSI, MCLK, SS) and one ground, this one has an additional line which may be used to trigger interrupts in a master device and thus force it to query the slave. The connector is standard  $2 \times 3$  pinheads with a 2.54 mm raster.

The RS232C interface built into the converter performs two functions. The first is system programming. This feature enables the  $\mu$ C flash memory to be written without the help of any external programmers except for the standard RS232C connector in PC. Even if no COM is available, it is still possible to use the USB to RS232 converter. The second function of the RS232C interface is to debug and control the device. The firmware enables users to watch data incoming from both the MIDI and the SPI device using a standard serial terminal. By sending data from the PC to the converter, one can change data flow, main device parameters, and send data directly to the MIDI or SPI devices. This feature is described more precisely in the next subsection.

The hardware of the interface consists of a DB9 female connector mounted on the board, and a MAX3232 device which handles voltage level translation ( $\pm 12$  to +3.3 V) and is directly connected to the MCU's UART.

The device status is reflected by four LEDs. One of the LEDs is located in the power block to indicate power supply. The remaining three are duo-LEDs showing the status of six internal MCU buffers. During startup or configuration changes, the indicators show if the operation was successful.

The expansion slot is a standard  $2 \times 8$  2.54 raster pin header and allows an optional LCD driver (e.g. HD44780) and an analog keyboard to be connected. This feature creates a possibility to change some settings manually. All lines, except for the power lines, are driven by the firmware, so it is possible to use them for other purposes.

### 3.2. Software description

The software for MCU is written in plain C and compiled with gcc (GNU C Compiler) under Linux OS. Flash and RAM as well as the core and peripheral SFRs (Special Function Registers) are located in a flat 32 bit address space, which allows to write consistent, PC-like code.

The main tasks of the program are gathering, processing and routing data to proper interfaces. To allow smooth and undisturbed data flow, six big cyclic buffers are declared, – two for each of the SPI, MIDI and UART interfaces. The firmware is half-interrupt driven. This allows the code to process and send data asynchronously with data reception. Data incoming from the UART, MIDI and SPI interfaces causes the program to stop any current operation and forces it to execute the ISR (Interrupt Service Routine). As soon as data is stored in a proper buffer, the suspended operation is resumed. Data processing and non-blocking sending is done in the main program loop. Parsing functions empty input buffers, process data and send it to correct output buffers. Figures 2 and 3 show the diagrams for the interrupt service subroutines and for the main program accordingly.

The most important function of the converter is to receive data from the MIDI IN, process it and send through SPI. The basic processing in this case would be parsing data and then sending it to a buffer for every event on each channel. This solution would require a lot of memory and DSP would query the converter for data all the time. It is clear that coefficients should be stored in the memory of the signal processor.

A better solution in this case was to invent a custom protocol which would simplify the work of DSP. The idea is to bind every event on each channel to a number from the range of 0 to 255. These values are stored in a translation table. Parsed datum selects

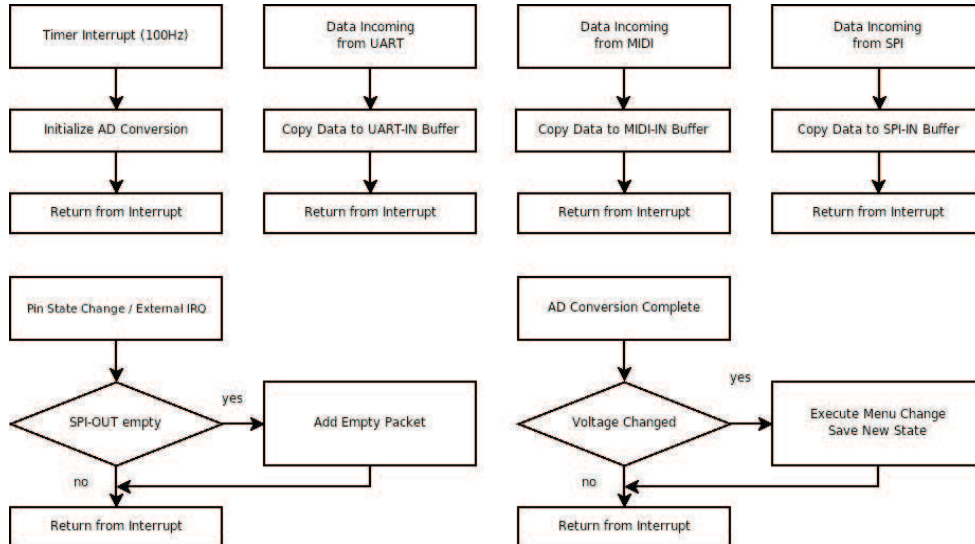


Fig. 2. The interrupt service routines (ISR).

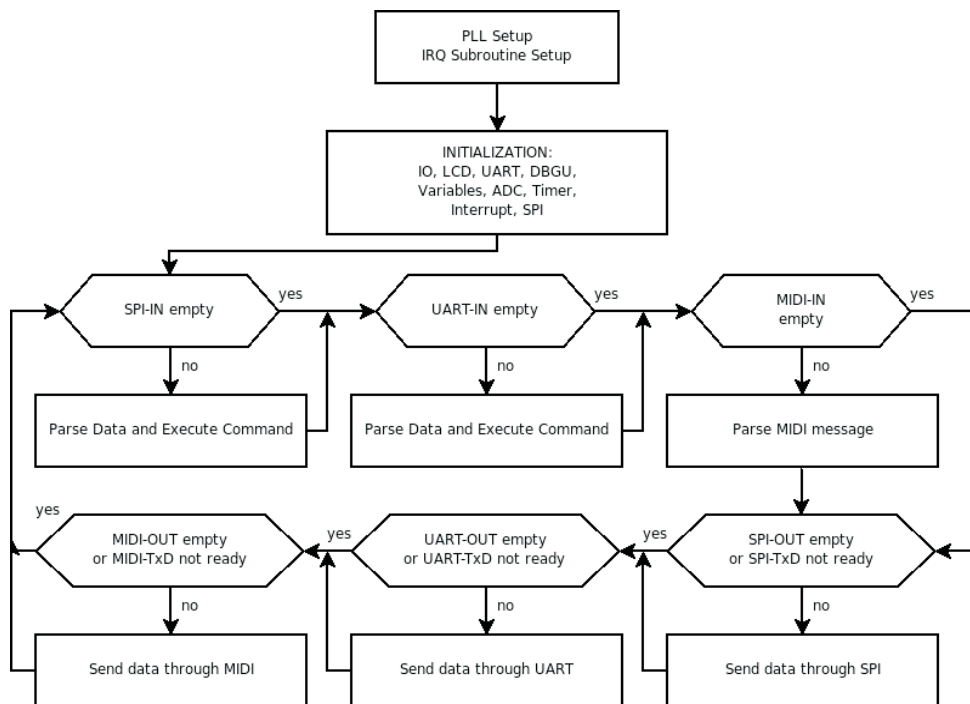


Fig. 3. The main program diagram.

an appropriate value from the table and copies it to a packet header. The second byte is filled up with a payload from a MIDI message. All messages in this protocol are 2 bytes long and may be sent and received as 16-bit words. DSP has to be configured as a SPI slave device. After the reception of data, all it has to do is to put the second byte into a proper place. The recommended option is to use the first byte as an index to a 256-bytes table. This approach enables the DSP user to use up to 254 coefficients and greatly simplifies the way the data are managed because it is possible to get information directly from the input buffer. There are two special values for the first byte which are reserved and not recognized as an address. The first is  $0 \times 00$ , which means that the event should not be forwarded to the DSP board and the second is  $0 \times FF$ , which is designed to be used as an empty message when data needs to be read but there is nothing to be sent.

#### 4. Applications

The most basic use for the MIDI to SPI converter is tuning digital filters in real time. During tests, the device was steered by the Behringer BCF2000 MIDI console [6], which enables the control of up to 16 parameters at once. Further faders and rotary encoders are available after the preset/bank change. This device supports feedback so changing its settings from the DSP board or PC is also possible.

Fast response time, obtained with fast  $\mu\text{C}$  and high SPI throughout (much higher than MIDI's 31.25 kB/s), almost instantly provides DSP with data coming from controller. The main limit in perceiving sound change is not the technology, but human's hearing. According to [4] signal delays up to even 15 ms are inaudible.

The converter may also be used as an interface for creating custom music synthesis applications based on DSP. The synthesized sound parameters can be controlled using standard MIDI keyboards (both with static or dynamic keys) enabling to set such parameters as, for example, key pressing velocity. If a keyboard is equipped with additional manipulators, such as knobs or faders, they can also be used to control the algorithmic parameters. The emulation of MIDI consoles and keyboards on a PC via an UART is also supported by the firmware. This feature extends the possibility to create even more complex systems with the use of the most common programming tools.

Currently MIDI to SPI converter is being used in Electroacoustics Division of Institute of Radioelectronics as a part of developed audio mastering system. Switching from inconvenient four-button keyboard and 8-led display took less than one day. In future it is planned to use MIDI to SPI converter during various students' activities. Mentioned before hardware MIDI controllers will be replaced by software consoles to reduce the cost of system while retaining it's capabilities. Screenshot of such program is shown in Fig. 4.

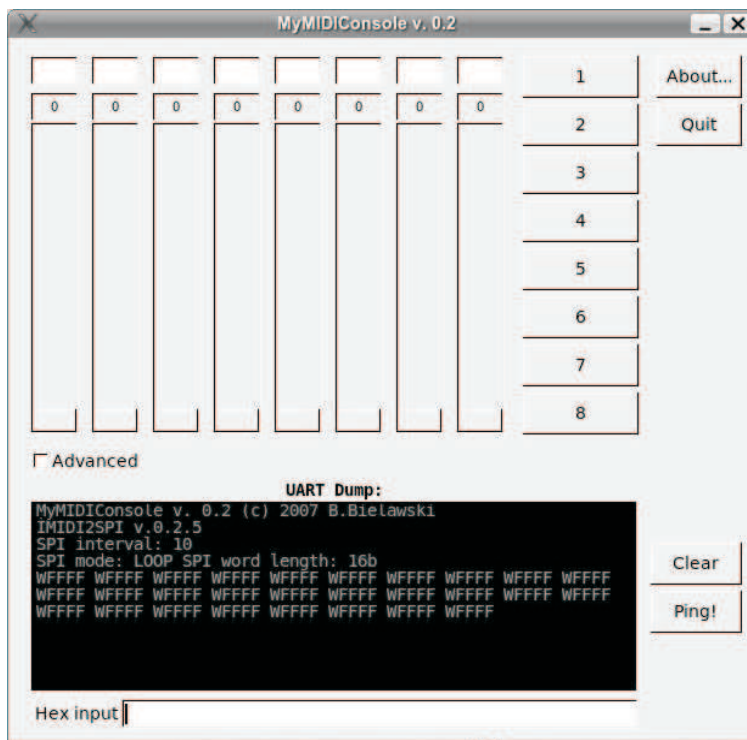


Fig. 4. Screenshot of virtual MIDI console.

## 5. Conclusions

Due to the lack of similar equipment available on the market, it was necessary to create such interface. The developed MIDI to SPI converter allows to control any device with the SPI interface from any kind of a MIDI device. In this paper, for instance, the digital audio signal processing system is presented. The system consists of an evaluation board ADSP-21364 EZ-KIT with the SHARC-21364 processor which can be controlled from the BFC2000 MIDI console through the described converter. The use of universal interfaces, such as MIDI and SPI makes the designed system very flexible and useful for both educational and professional applications.

## References

- [1] ADSP-21364 EZ-KIT Lite Evaluation System Manual, Analog Devices, Inc., 2005.
- [2] ADSP-21364 SHARC Processor, Technical Data, Analog Devices, Inc., 2005.
- [3] Complete MIDI 1.0 Detailed Specification v. 96.1 (2nd edition), MIDI Manufacturers Association (MMA), 2001.
- [4] CZYŻEWSKI A., *Dźwięk cyfrowy*, EXIT, Warszawa 1998.
- [5] <http://www.atmel.com/products/AT91/> – AT91SAM 32-bit Microcontrollers.
- [6] <http://www.behringer.com/BCF2000/> – B-CONTROL FADER BCF2000.
- [7] SPI Block Guide V03.06, Freescale Semiconductor, Inc., 2003.