Design and Implementation of a Frequency-Tunable Ultrasonic Signal Receiving System (Postprint)

Authors: Li Hailong, Zhai Guangjie, Jiang Yuanda

Date: 2017-03-10T00:00:00+00:00

Abstract

To achieve high-speed acquisition of ultrasonic signals at different frequencies, a design scheme for an ultrasonic signal reception and acquisition system with tunable receiving frequency based on programmable filters, USB2.0, and C# is proposed, and the system' s hardware and software design is completed. The hardware portion is primarily used for digital quantization of analog signals and their upload to the host computer, while the software portion utilizes C language and C# programming to implement normal data acquisition, upload, and storage. Practical applications demonstrate that the system features simple operation, accurate measurement, and flexible usage, fulfilling the design requirements.

Full Text

Design and Implementation of an Adjustable-Frequency Ultrasonic Signal Receiving System

LI Hai-long[1],[2], ZHAI Guang-jie[1], JIANG Yuan-da[1] [1] National Space Science Center, Chinese Academy of Sciences, Beijing 100190, China
[2] University of Chinese Academy of Sciences, Beijing 100049, China

Abstract: To achieve high-speed acquisition of ultrasonic signals across different frequencies, this paper presents a design scheme for an ultrasonic signal receiving and acquisition system with adjustable receiving frequency, based on a programmable filter, USB 2.0, and C#. The complete hardware and software system was designed and implemented. The hardware portion digitizes analog signals and uploads them to the host computer, while the software components, programmed in C and C#, enable normal data acquisition, transmission, and storage. Practical applications demonstrate that the system features simple operation, accurate testing, and flexible usage, meeting all design requirements.

Keywords: Ultrasonic receiving; Programmable filter; C#; USB 2.0

Ultrasonic waves, defined as sound waves with frequencies above 20 kHz, exhibit excellent directionality, strong penetration capability, and characteristic behaviors such as reflection and refraction. Due to these properties, ultrasonic waves of different frequencies find extensive applications across diverse fields, making them widely used in production, scientific research, and daily life [1]. However, during signal reception, the acoustic signal captured by the ultrasonic transducer and converted to an electrical signal is extremely weak, typically only reaching the millivolt level, making it highly susceptible to interference from other signals [2]. Therefore, ensuring a high signal-to-noise ratio is critical for accurate signal reception. Since amplification circuits amplify both useful signals and noise, ultrasonic receiving circuits must incorporate appropriate filtering functionality. Most conventional ultrasonic receiving circuits employ fixed filter types and filtering methods, which limits their flexibility and versatility. The programmable filtering component in this design effectively addresses these limitations, enabling flexible filter configuration and broad applicability.

After amplification, filtering, and analog-to-digital conversion, the acquired data must be transmitted to the host computer via an appropriate communication interface. Serial communication is widely adopted due to its minimal wiring requirements and low cost, particularly for long-distance transmission where it avoids inconsistencies in multi-line characteristics. Common serial communication standards include RS232, RS422, RS485, and USB. USB (Universal Serial Bus) is an external bus protocol standard, and data acquisition cards based on USB interfaces offer numerous advantages including hot-plugging, high transmission rates, strong reliability, excellent universality, easy expandability, and high cost-effectiveness. Consequently, this system employs a USB 2.0 interface for data upload [3]. The connection between the host computer and USB device, as well as functions such as data transmission and storage, are implemented through host software. C# is an object-oriented high-level programming language released by Microsoft for the .NET Framework, integrating extensive practical libraries that greatly simplify program development and have gained widespread application in data acquisition fields. Therefore, the host computer software for this system was developed using C#.

This paper focuses on the design of an adjustable-frequency ultrasonic signal receiving circuit for frequencies below 150 kHz and a data acquisition system based on USB 2.0 and C#.

1 System Overall Architecture

The system comprises both hardware circuits and software design components. The overall system architecture is illustrated in [Figure 1: see original paper]. The hardware circuits include an in-phase amplification circuit, programmable filter circuit, ADC circuit, FPGA circuit, microcontroller main controller, and

USB 2.0 controller. The USB 2.0 controller chip utilizes Cypress' s EZ-USB FX2, the FPGA chip employs Altera' s Cyclone series EP1C6Q240C8, and the ADC device adopts AD' s ADS822E. The software design primarily encompasses host computer program design, USB 2.0 firmware programming, microcontroller main controller programming, FIFO programming, and programmable filter programming.

In the system architecture, the amplification circuit boosts the weak signal received by the ultrasonic probe. The programmable filter circuit implements different filters for various frequencies through software configuration. The ADC circuit digitizes the amplified and filtered analog signal. The FIFO buffers the digitized ADC data to prevent data loss due to mismatched sampling and USB interface transmission rates. The microcontroller main controller manages the FIFO, ADC, and USB operations to ensure timely and accurate data upload from the lower computer to the host. The USB 2.0 controller facilitates communication between the lower computer system and the host computer.

2.1 In-phase Amplification Circuit

The amplification section of the ultrasonic receiving circuit is implemented using two cascaded inverting amplifier circuits to achieve in-phase amplification. Since the received signal amplitude is at the millivolt level, substantial amplification is required to prevent the signal from being masked by noise. Therefore, the operational amplifier must have a high gain-bandwidth product to satisfy both frequency and amplification requirements [4]. This design employs the AD8051 operational amplifier with a gain-bandwidth product of 110 MHz, which meets the practical requirements. To flexibly accommodate real-world needs, the circuit adopts a two-stage amplification scheme with adjustable gain. The amplification range is 47 to 940.

In the first-stage amplification circuit, shown in [Figure 2: see original paper], resistor R1 is 10 kΩ and resistor R2 is 470 kΩ. In the second-stage amplification circuit, resistor R4 is 10 kΩ and the adjustable resistor R5 ranges from 10 kΩ to 200 kΩ. The first-stage inverting amplification gain is calculated as A_f1 = -R2/R1 = -470kΩ/10kΩ = -47. The second-stage inverting amplification gain is A_f2 = -R5/R4, which ranges from -10kΩ/10kΩ to -200kΩ/10kΩ, yielding -1 to -20. Therefore, the overall amplification circuit gain is A_f = A_f1 × A_f2, providing a range of 47 to 940.

2.2 Programmable Filtering

Conventional filter circuits, whether active or passive, require tedious debugging, and each circuit can only implement a single filter type for fixed-frequency signals, making it impossible to achieve multi-mode filtering. The MAX262 dual second-order universal switched-capacitor active filter from Maxim Integrated uses a microcontroller to precisely control its filter functions. Without requiring external components, it can construct various filter types including

band-pass, low-pass, high-pass, band-stop, and all-pass filters, all configurable through programming. The center frequency is adjustable from 1 Hz to 140 kHz [5], satisfying the requirements for common ultrasonic frequencies. Therefore, this design employs the MAX262 chip to implement different filters, enabling the system to perform multiple filtering approaches for various signals through configurable filter types, thereby meeting diverse ultrasonic signal data acquisition requirements. Maxim provides the MAX260V1 design software for the MAX262 series filters, which significantly simplifies the design process.

2.3 ADC Sampling Circuit Design

A critical performance metric for analog-to-digital conversion is data conversion accuracy. High sampling frequency and high-resolution ADCs enable more accurate data conversion. The ADS822E from Analog Devices is a 10-bit sampling ADC with a maximum conversion rate of 40 MSPS, meeting the signal frequency requirements of this system. The chip integrates all necessary functions for analog-to-digital conversion, including track-and-hold (T/H) and reference voltage sources. The ADS822E requires a 5 V power supply and supports both internal and external reference voltage modes. Its digital outputs are compatible with TTL and CMOS levels, and data output is controlled through its output enable pin OE [6]. This system configures the input analog voltage range as ±1 V.

The PLL within the FPGA provides the required clock signals for the system to ensure clock stability, including the sampling clock for ADS822E, the clock for the filter circuit, and the FIFO clock. After analog signals are sampled and converted to digital signals by ADS822E, they are sent to the FIFO within the FPGA for buffering to prevent data loss caused by mismatched sampling and USB interface transmission rates. The FIFO depth in this design is 1024, and since the ADC outputs 10-bit binary data, the data bus width is configured as 16 bits [7].

2.4 USB Interface Circuit Design

The EZ-USB FX2 offers powerful interface design modes including General I/O mode, Slave FIFO mode, and GPIF mode. Both Slave FIFO mode and GPIF mode enable high-speed data transmission. This system employs Slave FIFO mode, which allows communication between USB data and external logic without intervention from the on-chip CPU, thereby improving transmission efficiency.

In Slave FIFO mode, the interface circuit between EZ-USB FX2 and the external logic controller is shown in [Figure 3: see original paper]. The IFCLK is the interface clock, generated at 40 MHz by the FPGA through PLL division. FLAGA through FLAGD are FIFO flag pins indicating internal FIFO empty/full status (active low). In synchronous mode, SLOE enables the output of data bus FD (active low by default); data can only be read from the FIFO

pointer when SLOE is active. SLRD and SLWR serve as synchronous read and write signals for the FIFO (active low by default); data can only be written to or read from the FIFO on the rising edge of IFCLK when SLRD or SLWR is active. FD[15:0] is the 16-bit bidirectional data bus responsible for data transmission between the lower and upper computers. FIFOADR[1:0] selects the internal FIFO, where "00" selects endpoint EP2 and "10" selects endpoint EP6 [8].

3 System Software Design

The system software design comprises five main components: microcontroller main controller program, programmable filter control program, USB program, FPGA FIFO program, and C# application program, as illustrated in [Figure 4: see original paper].

3.1 USB Program Design

USB firmware program development includes several key steps. First, USB device descriptors are defined. The EZ-USB FX2 is then configured to operate in Slave FIFO mode, with the FIFO working clock IFCLK generated by the FPGA' s PLL. Endpoint EP2 is configured as an OUT endpoint for receiving commands from the host computer, while EP6 is configured as an IN endpoint for uploading acquired data from the lower computer. The endpoint data bus width is set to 16 bits with a data buffer size of 2048 bytes [9][10]. After this configuration, the USB interface can receive data from external logic; when external logic writes data to USB, the USB chip automatically uploads the data without CPU intervention, resulting in high transmission rates.

The USB device driver is critical for USB peripheral development, serving as the bridge between the host application and USB firmware. Its primary function is to enable the host operating system to recognize the USB device and facilitate communication between the host and the USB peripheral [11]. This system' s driver utilizes CYUSB.inf and CYUSB.sys from the Cypress Suite USB development kit [12].

3.2 Microcontroller Main Controller Program Design

The microcontroller serves as the system' s main controller, generating control signals for the EZ-USB FX2, FIFO, programmable filter, and ADC. It monitors the empty/full flags of the EZ-USB FX2's internal FIFO to generate appropriate read/write signals. By detecting the empty/full status of the FPGA' s FIFO, the microcontroller controls the ADC' s output enable pin OE to enable data acquisition and transmission.

3.3 C# Application Program Design

The host computer software primarily handles USB device connection, data acquisition start/stop control, data saving, and USB chip FIFO reset functions.

The host software was developed using C# in the Visual Studio 2010 .NET environment. The .NET framework integrates numerous practical class libraries; this system primarily utilizes functions from CyUSB.dll and the Thread class for USB communication and multi-threaded programming [13].

The host program workflow is shown in [Figure 5: see original paper]. When the user clicks the "Connect USB Device" button, the program attempts connection. Upon successful connection, it displays "Connection Successful" along with the device's VID and PID numbers. If connection fails, it displays "Connection Failed." After successful connection, clicking "Start Acquisition" sends a command to the lower computer to begin operation, causing acquired data to be uploaded to the host and simultaneously saved to a specified directory. When acquisition completes, clicking "End Acquisition" displays "Acquisition Completed," terminates the process, and resets the USB chip's FIFO [14][15].

4 System Testing

Following the completion of hardware circuit and software program development, joint hardware-software testing was conducted to verify system performance. The test procedure involved: first configuring the EZ-USB FX2 driver, then using Cypress USB Console software to download the firmware program. As shown in [Figure 6: see original paper], the host computer detected the USB device with VID 0x04B4 and PID 0x1004, confirming successful USB device connection. After proper connection between the USB device and host PC, data acquisition could commence.

Clicking the "Start Acquisition" button caused the lower computer's acquired data to be saved to a file in the specified directory. The collected data was represented as hexadecimal values ranging from 0000 to 00FF. Testing was performed by inputting sinusoidal signals of 40 kHz and 100 kHz at the analog input terminal. Verification confirmed that the actual saved data matched theoretical values exactly, demonstrating the reliability of the system's data transmission.

5 Conclusion

Both the software and hardware of this system have been debugged and successfully applied in actual projects, operating reliably. The system's signal amplification and programmable filtering capabilities enable accurate acquisition of weak ultrasonic signals across different frequencies. Leveraging Cypress's USB development kit significantly reduces the development difficulty of USB 2.0 interfaces, while the extensive class libraries integrated in the Visual Studio 2010 .NET environment facilitate rapid development of the host computer program, shortening the development cycle. The system features low hardware cost, flexible architecture, easy expandability, and strong practical value for promotion.

References

[1] YANG Zhao-fei. Improvement of receiving circuit for digital ultrasonic system[J]. Shanxi Electronic Technology, 2013(2):3-5.
[2] SUN Ling-yi, GAO Qin-he, CAI Wei, et al. Design of ultrasonic transmitting and receiving circuit driven by low-voltage power supply[J]. Instrument Technique and Sensor, 2010(10):77-79.
[3] HU Xiao, ZHANG Qi, ZHANG Lian-chao. Characteristics and application of USB2.0 controller CY7C68013[J]. Microcontrollers & Embedded Systems, 2002(10):48-50.
[4] SHEN Li-jun. Research on design and key technologies of ultrasonic phased array transmitting and receiving circuits[D]. Nanjing University of Aeronautics and Astronautics, 2014.
[5] JIANG Yu, CHEN Xun, YANG Xue, et al. Implementation of programmable filter based on MAX262[J]. Ordnance Industry Automation, 2001, 20(2):36-39.
[6] QI Hong-tao, SU Tao. Design of high-speed AD sampling based on FPGA[J]. Aero Weaponry, 2010(1):35-39.
[7] LEI Hai-wei, LIU Jun. Design and implementation of soft FIFO in FPGA[J]. Microcomputer Information, 2008, 24(2).
[8] QIAO Li-yan, WU Jiang-wei, XU Hong-wei. Design of USB2.0 interface for data acquisition card[J]. Electronic Measurement Technology, 2010, 33(1):57-60.
[9] JIANG Jin-tao, YANG Ming. Performance characteristics and data transmission implementation of USB2.0 controller EZ-USB-FX2[J]. Computer Engineering and Applications, 2005, 41(11):94-96.
[10] GOU Xin-yun, ZHANG Yu, JI Zhong-mei. Firmware development for USB2.0 interface chip CY7C68013[J]. Microcomputer Information, 2005, 21(4):182-183.
[11] GAO Bei. Design of data transmission system based on USB2.0[D]. Xidian University, 2011.
[12] WANG Bing-bing. Research on data transmission system based on EZ-USB FX2 and FPGA[D]. Xidian University, 2010.
[13] Karli Watson, Marco Bellinaso. C# Programming with the Public Beta[M]. Translated by KANG Bo. Tsinghua University Press, 2002.
[14] HE Wei, CHEN Yong-qiang. Research and case analysis of file processing in C#[J]. Computer Knowledge and Technology: Academic Exchange, 2009, 5(21):6046-6047.
[15] ZHENG Jie. USB 3.0 Programming Bible[M]. Publishing House of Electronics Industry, 2013.

First Author Biography: LI Hai-long (1992-), male, from Jining City, Shandong Province, Master' s degree, graduate student, research interests: computer technology.

Note: Figure translations are in progress. See original paper for figures.