Engineering Science and Technology, an International Journal 19 (2016) 964–969

HOSTED BY

Contents lists available at ScienceDirect

Engineering Science and Technology, an International Journal

journal homepage: http://www.elsevier.com/locate/jestch

Full Length Article

Microcontroller USB interfacing with MATLAB GUI for low cost medical ultrasound scanners

Jean Rossario Raj [a,*], S.M.K. Rahman [a,b], Sneh Anand [a,b]

[a] Center for Bio-Medical Engineering, Indian Institute of Technology, New Delhi, India
[b] Biomedical Engineering Unit, All India Institute of Medical Sciences, New Delhi, India

ARTICLE INFO

Article history:
Received 13 December 2015
Received in revised form
1 January 2016
Accepted 12 January 2016
Available online 1 February 2016

Keywords:
MATLAB
MEX program
Microcontroller
Ultrasound scanner
USB
USBXpress

ABSTRACT

This paper presents an 8051 microcontroller-based control of ultrasound scanner prototype hardware from a host laptop MATLAB GUI. The hardware control of many instruments is carried out by microcontrollers. These microcontrollers are in turn controlled from a GUI residing in a computing machine that is connected over the USB interface. Conventionally such GUIs are developed using 'C' language or its variants. But MATLAB GUI is a better tool, when such GUI programs need to do huge image/video processing. However interfacing MATLAB with the microcontroller is a challenging task. Here, MATLAB interfacing through an intermediate MEX 'C' language program is presented. This paper outlines the MEX programming methods for achieving the smooth interfacing of microcontrollers with MATLAB GUI.

1. Introduction

Microcontroller-based systems offer major advancement as an internal and external control. Microcontrollers can control majority of the internal devices in a typical circuit board. Moreover majority of the chips also have built-in interfaces that can be controlled by the microcontroller. They have USB interface through which it is interfaced with external devices such as a computer or a server [1]. A low-cost USB interface for operant research using Arduino and Visual Basic is presented in [2]. Many data acquisition systems use microcontroller and USB interface [3]. The microcontroller programming is done in traditional 'C' language. Microprocessors and microcontrollers provide the path for the integration of hardware and software [4]. Microcontrollers have far-reaching applications in the field of instrumentation [5,6].

The application in the host computer or the server is generally written in languages like 'C' or its variants like C++, Visual C++, Java, etc. [7]. The microcontroller manufacturers provide interface programs in the form of windows dynamically linked libraries (DLL), which can be compiled along with the applications for interfacing with the microcontroller [8,9].

For applications involving signal processing, image processing or video processing, MATLAB is a more suitable programming language [10,11]. The quality metrics and visual perception in ultrasound imaging under different imaging conditions using MATLAB test bench were studied in [12]. In MATLAB, there are many built-in functions and routines that enable faster rollout of the application [13]. MATLAB has a very large database of built-in algorithms for mathematical modeling, image processing, simulations and computer vision applications [14,15]. MATLAB has the ability to read in a wide variety of both common and domain-specific image formats.

However the major challenge is in interfacing the MATLAB application with the microcontroller to get the best of both worlds. Such requirements involve live reading of images, videos, and direct processing and display [16]. One application where we came across the challenge is while developing a prototype for a medical ultrasound scanning machine [17,18]. The application was made in MATLAB to take advantage of the rich image and video processing capabilities [10,19]. The hardware has USB interfacing to the microcontroller for configuration of the various parameters of the internal chips and for performing different operations [20–22]. The microcontroller interfaces with different internal chips, such as ADC, transmit side FPGA (TxFPGA), receive side FPGA (RxFPGA), etc. [23,24]. The block schematic of microcontroller interfaces is shown in Fig. 1. The ADC has custom serial bus programming whereas FPGAs used Serial Peripheral Interface (SPI) programming. The interface programs toward these devices were also developed for the microcontroller as well as the FPGAs.

Neither MATLAB nor the microcontroller manufacturers provide the interface logic for interworking. Hence interworking between

* Corresponding author. Tel.: +91 9811084119; fax: +91 1126862037.
E-mail address: bmz108120@cbme.iitd.ac.in (J.R. Raj).
Peer review under responsibility of Karabuk University.

http://dx.doi.org/10.1016/j.jestch.2016.01.008

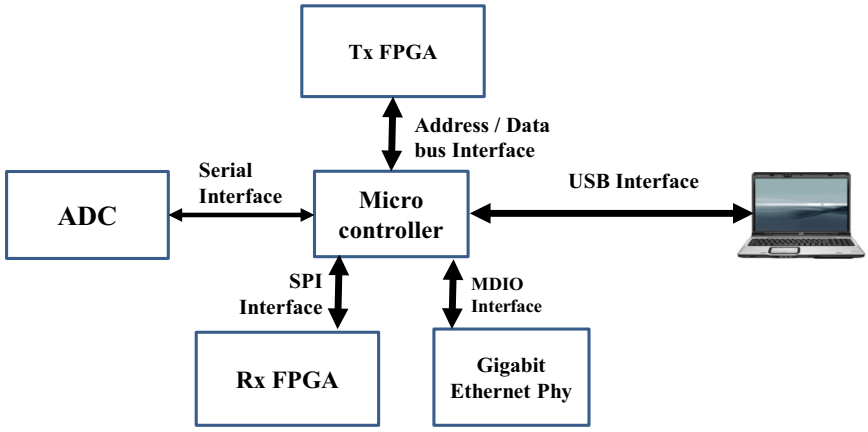

Fig. 1. Block schematic of the experimental setup.

the MATLAB GUI and the microcontroller is a challenging task. The novel method used for interfacing using a C language interface function called MEX program is described in this paper.

2. Materials and methods

2.1. Interfacing architecture

The logical block schematic of the MATLAB GUI and microcontroller interfacing with USB is given in Fig. 2, which indicates the microcontroller side as well as the MATLAB GUI side logic used for the study. A Silicon Laboratories C8051F340 microcontroller is used. It operates at a maximum speed of 48 MHz with 4 k of on-board RAM (XRAM) and 64 k flash memory. The microcontroller has integrated USB receiver and USB controller. It is 48 IO pins configured as 5 IO buses of 8 IOs each. The microprocessor in the ultrasound scanner prototype interfaces with a bank of ADCs and FPGAs. The microprocessor firmware is written in C language. One IO Bus is used as the programming header for programming the microcontroller. The firmware is transferred to the microcontroller using the programmer hardware by M/s Silicon Labs. The development required the USBXpress API provided by M/s Silicon Labs for including in the microcontroller firmware, firmware development, USBXpress driver installation in the host computer, installation of C++ compiler (Visual C++ used), development of the MEX file, development of the MATLAB GUI, and compilation and linking of the MEX file from MATLAB.

2.2. USBXpress API

The Silicon Laboratories USBXpress provides the host and device software solution for interfacing communication bridges to the USB. A high-level Application Program Interface (API) for both the host software and device firmware is used to provide USB connectivity. The USBXpress includes windows device drivers, device driver installer, host interface function library (host API) provided in the form of a DLL, and device firmware interface function library. Some of the important functions performed by the host API are given in Table 1. While performing each function, the function returns the status of the operation like SI_SUCCESS or SI_DEVICE_NOT_FOUND or SI_INVALID_PARAMETER, along with any return values to take suitable corrective steps in read or write logics. The API is used in an interrupt driven mode. The c8051340.h and USB_API.h files supplied as part of the application are added to the project in addition to the "main" firmware file. While building the target, the USBXpress library file, USBX_F320_1.LIB is linked as an external object.

2.3. Microcontroller firmware

In the developed prototype, the microcontroller configures the different chips like ADC, FPGA, etc., through the control from the MATLAB GUI. Hence the microcontroller firmware implements the USB interface logic as well as the interface logic for the other devices. The header and the source files developed for the interfacing with the ADC, TxFPGA, RxFPGA and PHY device using the four IO Buses are included in the firmware. Even though the ADC side does not require any separate software logic, the FPGA side requires the hardware logic written in languages such as Verilog or VHDL. Similar interface logic is written in the FPGA side also for the read and write operations using the SPI interface. The SPI interface has read, write, chip enable, Master in Slave Out (MISO) and Master Out Slave In (MOSI) pins for the interconnection. The USB_API.h includes the definitions needed by common code to control the state of the USB peripheral, but this doesn't need to know about the specific implementation. The USB_API.h for the USB interface functions and other files like C8051F340_def.h, compiler_defs.h, stddef.h, stdio.h etc. are included in the compilation. The flowchart of the firmware is given in Fig. 3.

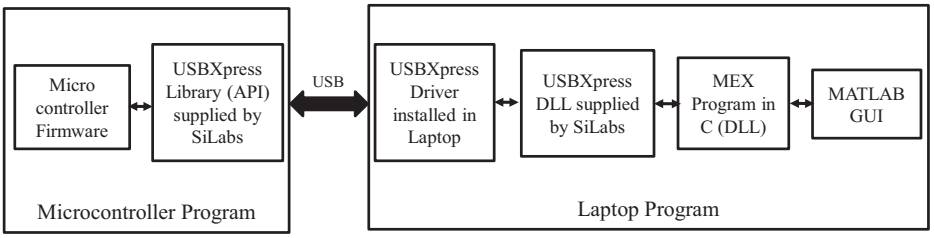

Fig. 2. Microcontroller MATLAB interface.