USB TO PARALLEL CONVERTER

Florin Grofu, *"Constantin Brancusi" University of Targu Jiu, Romania*
Constantin Cercel, *"Constantin Brancusi" University of Targu Jiu, Romania*

Abstract: The current age of technology is the result of numerous inventions and discoveries and responsible for this development is the need, then, our capability to transmit data and information in various ways. From reducing the cost of production of physical support through its miniaturization, manufacturers of the new computer systems eliminate traditional connectors for serial and parallel input / output ports. USB bus is present in all computers and tends to replace the standard ports. This paper presents the realization of a USB to Parallel converter for connection with microcontrollers using FTDI ICs and a test application for it operation.

Keywords: converter, system, RS232, parallel

INTRODUCTION

The main advantages of the USB port is that the devices can be connected and disconnected any time, and can be connected many devices in a tree structure. The USB bus is a fast serial interface, bidirectional, cheap and easy to use [1]. The advantages of this solution to the old RS232 serial interface are:

- transfer rate: up to 5Gbps (USB 3.0), compared with 115 Kbps for RS232;
- can connect up to 127 devices to the PC (it is a bus), compared with 2 devices connected through RS232;
- easy to use by end user add / remove devices in / from system is very easy;
- has a flexible protocol[1].

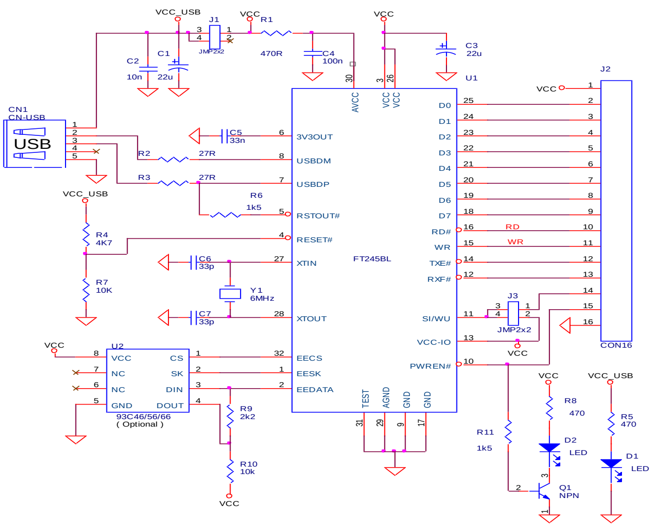

Fig.1. USB communication module

USB is a cable bus that allows data exchange between a host computer and a wide range of simultaneously accessible peripherals. The bus allows peripheral device to be attached, configured, used and disconnected while the host and other peripherals operate. USB was designed primarily for users who do not wish to enter into the hardware installation details, so complicated cabling system was replaced with a software control. All the problems supposed by interconnecting multiple devices with different performance and transfer rates are handled by software.

Physically, the bus consists of four wires, two are for power and two for data transfer, D + and D-. Up to 127 devices can be used simultaneously through these lines without having interrupt or DMA channels conflicts or overlaps of the input / output addresses. Supporting data transfer speeds of 12Mbps (USB 1.0), 480Mbps (USB 2.0.) and up to 5 Gbps (USB 3.0), bus can be used to transfer signals such as speech signals and images in real time. A major difference between USB devices and other data communications devices is the USB uses an isochronous communication protocol while serial devices using asynchronous and synchronous communication protocol. In isochronous communication, clock information (speed) derived from or included in the data stream and the delay is dependent on channel characteristics and can be determined. Using isochronous protocol, peripheral has guaranteed access to the bus data stream and can be maintained a constant data rate.

Considering the complexity of the USB interface, switching a LED, ordering a block of 8 relays, and reading of 8 switch sites or a combination of these, is very difficult. Now this is possible using a single chip without any help from a microcontroller. This may be due to developers from FTDI (Future Technology Devices International) who were able to release the new FT245BM USB FIFO and FT232 USB-UART devices [2].

The FT245BL provides an easy cost-effective method of transferring data to / from a peripheral and a host PC at up to 8 Million bits (1 Megabyte) per second. Its simple, FIFO-like design makes it easy to interface to any microcontroller or microprocessor via IO ports [3].

HARDWARE IMPLEMENTATION

Starting from the advantages of the FT245BL chip capable to realize all the required functions for an USB to Parallel converter, and due to it small dimensions, we minimize the electronic scheme and the effective construction using a small number of components.

To test the USB communication and the USB to Parallel converter features was designed an USB communication module based on 245BL FTDI chip which ensure USB communication and parallel data conversion. So, USB module has the diagram as presented in figure 1, being designed to select the power supply mode: from USB bus or from external source.

To particularize the USB communication module was used the possibility to connect a serial 93C46B type EEPROM memory with 64 words on 16 bits. The layout of PCB is presented in figure 2, with mention that all components are SMD type.

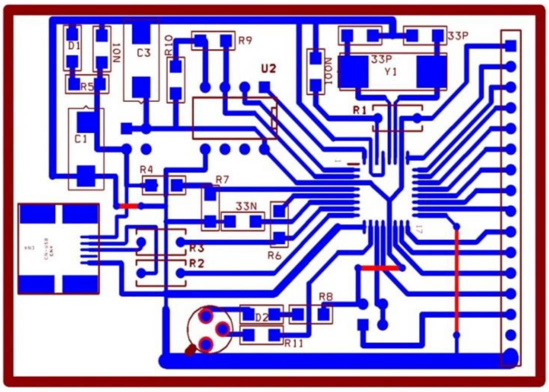

Fig.2. USB module PCB layout

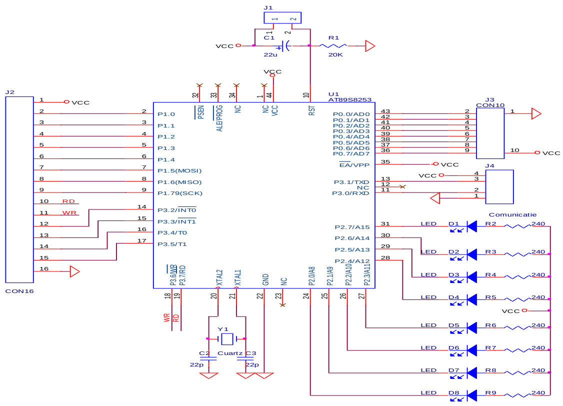

Fig.3. Microcontroller module

After practical implementation of the USB module was made customs settings. Without extern EEPROM memory or if it is not written, description parameters of USB module are the default parameters of 245BL circuit. These parameters are:

Vendor ID	0403
Product ID	6001
Manufacturer	FTDI
Description	USB <-> Serial Cable
Bus Powered	100mA
Serial Number	-

To modify and save these parameters in external EEPROM memory not need a special programmer. FTDI Company provides its users the Mprog application for configuring and saving data to EEPROM memory. To use this application as well as for connecting the USB communication module using LabWindows / CVI the D2XX FTDI driver installation is required.

Next, change the parameters as wanted but preliminary must study the documentation in order to not cause incompatibility between the software and the settings used. For example we can change the default settings VID = 0x0403 and PID = 0x6001, but if change these values have changed the driver INF and INI files must to be updated with the new VID and PID settings. For drivers to recognize the new device the settings need to be added into the files FTDIBUS.INF and FTDIPOT.INF [4].

You can also modify the maximum permissible current for internal protection and add a 8-digit serial number. After completing the configuration is saved to be programmed into EEPROM memory.

To test the speed of communication with an external device was designed and built an AT89S52 microcontroller module that plugs into the USB module. Microcontroller module diagram is shown in figure 3.

Can observe the proposed system is connected parallel to USB module, and has 8 LED’s to show different state in operating time. Have also a serial and a parallel output port to connect with other devices. Microcontroller module can be used also as an interface between USB module and external devices.

Microcontroller module work at 24MHz clock speed allowing to execute 2.000.000 instructions per second. It PCB is presented in figure 4 – a). AT89S52 microcontroller will be programmed so that it can transfer data packets with USB module through an application made in LabWindows/CVI.

The software application will sent to the microcontroller, via USB communication module, a data packet containing command parameters (the type of test to be made, the number of bytes to be transferred and the number of repetitions of the test), will then be transmitted and received data packets within size set by command the parameters. Transmission and receiving data package will be a certain number of times also determine by the command parameters. It can thus test duration of a test and by default the communication speed in different situations depending on the test performed.

Figure 4 – b) is shown the practical implementation of the two modules (USB and microcontroller).

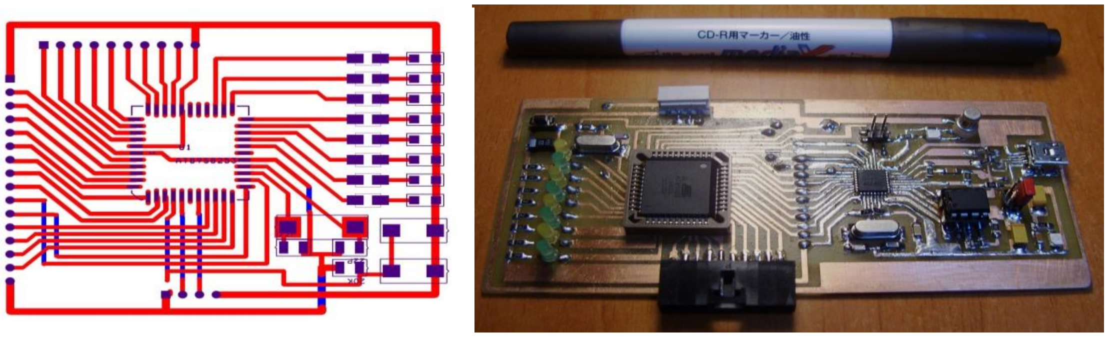

Fig.4. a) PCB of the microcontroller module b) Practical implementation

SOFTWARE APPLICATION

To test the speed and method of communication using implemented USB to Parallel converter, was performed a software application using development environment LabWindows/CVI.

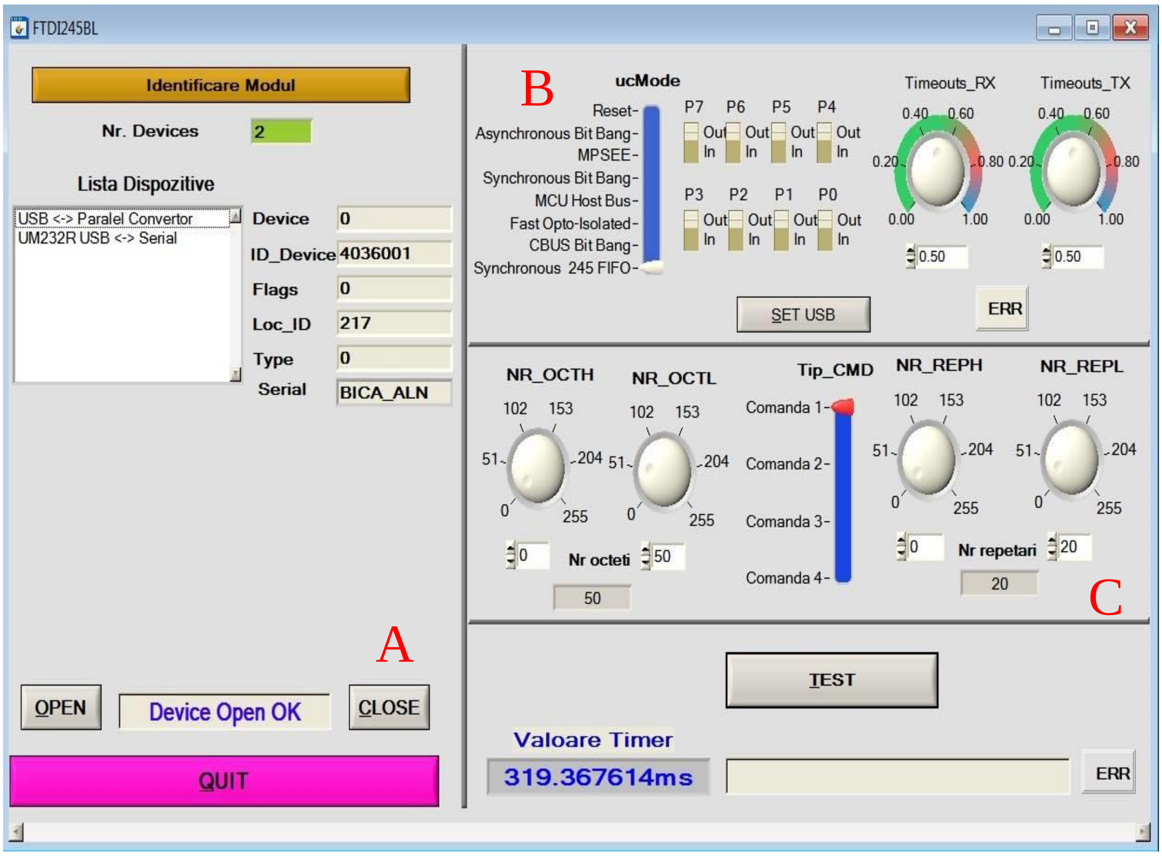

Fig.5. The interface of the application

One of the special characteristics of these devices is the "Bit-Bang" mode. This may change the 8 data lines of the FT245BL in input or output lines depending on need and the ability to send data or receive data without any external control. A first feature of this mode is interfacing a microcontroller / DSP / FPGA / etc. with a host computer. "Bit-Bang" is simply the latest extremely useful solution of this chip.

Data that is transferred from the FIFO to the output data lines is latched on the lines until a different byte is sent. No external buffer or latch is required to maintain the status of the data lines while in Bit-Bang mode [2].

To test the speed of communication, the software application implemented in LabWindows/CVI allows identifying FTDI modules connected to the PC, allows you to choose the desired device, it displays the descriptor parameters, set the mode and communication parameters, perform various tests communication and display times execution being a useful tool for testing USB communication possibilities.

Software implemented in order to test the possibilities of communication between a PC and a microcontroller using USB communication contains several functional blocks.

First operation area, presented in figure 5 – zone A, is dedicated to identify al FTDI devices connected to the PC.

To be able to work with FTDI devices such as the application developed in LabWindows/CVI, must load FTD2XX.lib library functions. Using functions from this library by pressing IDENTIFICARE MODUL are queried all connected FTDI devices and create a database with key descriptors of the identified devices. Also in NR.DEVICES box displays the number of identified FTDI devices and in the LISTA DISPOZITIVE will be passed these descriptions.

It is noted as the first device in the list is called USB <-> Parallel Convetor, name reallocated and saved in the EEPROM memory of the interface achieved. If is select one of the listed devices, all the other default parameters or renamed parameters (when using external 93C46B type EEPROM memory) will be displayed.

After selection of the desired devices from list this may be activated by pressing OPEN button. In the joined text box of this button will appear a message which indicate if the opening operation of the devices was successfully or not. If the devices was activated successfully (Device Open OK), can be made settings about communication mode using buttons from B area (figure 5 – zone B). Depending on the FTDI device linked, can be selected 8 working modes. Reset mode set the device on default communication parameters. For FTDI245BL module most useful communication modes are Asynchronous Bit-Bang and Synchronous FIFO.

In Asynchronous Bit-Bang mode, data lines are used directly without any control signals. Bit-Bang mode has the advantage that the each data pins can be configured independent as input or output pin. To make this setting easy was designed the selection buttons P0-P7 corresponding to data lines D0-D7.

Synchronous 245 FIFO mode is the operating mode that use control signals. So, using RXe and TXf lines communicate to the microcontroller connected to the USB interface if exist data into receive buffer to be read, or if can be written data into transmission buffer.

In the B area from software interface can be also set the waiting time (Timeaouts) for receiving and transmitting data.

SET USB button is to transfer the changes into USB to Parallel converter and into the application, and if there is an error the ERR indicator will be red.

To test the communication speed of the PC with realized microcontroller system was operate as follow.

From PC through the USB interface is transmitted a command to the microcontroller system, command set using C area from figure 5.

This command transmitted from the PC contain the type of command, number of test bytes settable between 1-65535 with the two buttons NR_OCTH and NR_OCTL, and how many times you can repeat the test value, from 1 to 65535, settable from the two buttons NR_REPH and NR_REPL. The command set as described above is sent to the microcontroller. Then, comes the effective test that consists of transmitting a data packet, with the size established, to the microcontroller. It checks the correctness of the data packet and then sends the same data package to be checked is the software from the PC. This test is repeated the number of times set by the command. Once the test is started, a software timer is started until the test is complete. The resulting value is displayed in the box VALOARE TIMER.

TESTS AND CONCLUSIONS

Test performed according to the settings from figure 5, namely the transmission and reception of a packet of 50 bytes and the test repeated 20 times, the time obtained was 319,367ms. The communication time for an octet obtained in this case is 319,367 / (50 bytes * (2 * 20 repeats)) = 160 µs which corresponds to a communication speed of 6250 bytes/second, which is 62500 bits/second while the direction of transmitting information is changed 20 times.

Following retesting for 50 bytes but with a number of 2000 repetitions the obtained time is 31.9 s, so equivalent transmissions speed of 62690 bits/second.

Following retesting for a total of 20000 bytes but with a total of 5 repetitions achieve a time of 5.185 s, so an equivalent transmissions speed of 385870 bits/second.

And testing data packet of 20000 bytes but one transmission, we get an equivalent transmission speed of 400000 bits/second.

In conclusion, after testing USB communication with a microcontroller system were obtained much faster communication speed than when using RS232 communication, speed which can greatly increase when transmitting large data packets. This is one of the main reasons why to the almost of automation devices communication "migrated" from the standard RS232 to USB communication.

REFERENCES

[1] OGRUTAN, Petre, GERIGAN, Carmen, BANCIU, Nicolae., "Memorii, interfete si periferice. Interfete specializate". [online]. Brasov: Editura Universitatii *Transilvania*, 2003, ISBN 973-635-118-1

[2] Don L. Powrie, "Bit-bang usb--perhaps the easiest usb interface yet", 2003

[3] *** http://www.ftdichip.com/Support/Documents/DataSheets.htm - FT245BL USB FIFO (USB - Parallel) I.C.

[4] *** http://www.ftdichip.com/Support/Documents/AppNotes.htm - Advanced Driver Options

[5] Grofu Florin, Cercel Constantin, "LPT to RS232 communication converter" – CISSE2009 – International Conference of Industrial Electronics, Technology & Automation, University of Bridgeport, USA, 4-12 Decembrie 2009, in press

[6] Microsoft Corp., "Extended Capabilities Port: Specifications", Revisions 1.06-1.14, 1992-1995