USB To USB Data Transfer Without PC

Mr. Tushar Sawant
M.E. Student (EXTC Dept.)
D. J. Sanghavi College of Engg., Mumbai
Email : tushar.sawant44@gmail.com

Mr. Sanjay Deshmukh
Asst. Professor (EXTC Dept.)
D. J. Sanghavi College of Engg. Mumbai
Email : vs_deshmukh@rediffmail.com

Abstract - Today, the need of life for many people in the world is importance of data in human life. It has become an inevitable part of day to day life, not only in technical field but also in non technical fields. The aim of the proposed system is to remove the every time need of the computer for the data transfer. The concept of the USB to USB will make it possible to carry out the mass data transfer anywhere, anytime. This user friendly device is actually recognized by its Connection type- the BUS; called the Universal Serial Bus. It comprises of just four connection wires between the Host (Computer) and the Device, governed by a set of rules called the Protocol. The portability and small footprint are the major advantages of the bridge.

Keywords - ARM7, FAT, Flash Disk, USB.

I. INTRODUCTION

The USB to USB Bridge is also based on the same concept-only thing is that it carries the data between the devices. Several data and application are developed daily which common computer user has to transfer from one USB Flash device into another, with the minimum wastage of time. Carrying a computer or a laptop just for the sake of data transfer is not affordable these days in the age when people want all devices to be handy. Moreover, transferring data via a computer involves a lot of power to be wasted. Also, the threat of viruses and malware has made the life of computer users more complicated. These viruses get activated as soon as the device is plugged into the system and get copied along with other data from one ash device into another. So a solution is provided by means of implementation of the bridge. The small footprint and ease of portability makes it a choice for the data transfer. This bridge will help the user to select a particular data file from the mass storage device connected to one of the ports and transfers it to the other mass storage device using some controls like explore, list, copy provided on the front panel. The Touch Screen displays the list of explored files from the mass storage device and does the selection of the operations to be performed. The linux environment provides the basis of all functions for the embedded design. The DMA is activated when the data transfer is initiated. Bidirectional data transfer is possible by means of half duplex communication.

II. REQUIREMENT OF SYSTEM

The motivation of the conceptual design of USB to USB Bridge was an answer to a question: what can be the solution to make the computer independent data transfer using USB? The only available answer resulted is that develop a system that will perform the same task as that of the computer. Making a system that will do the tasks of a computer – made us to think a little because the answer directly suggests implementing a system that will handle USB protocol along with the other processes.[1] The whole system has been built around these requirements.

Requirement 1: The first and very basic need was to have a look over the USB protocol. The protocol is easy to use but is somewhat difficult to implement in high and full speed modes.

Requirement 2: The data transfer speeds of the bridge should be commercially competitive.

Requirement 3: The USB to USB Bridge will be placed on a low end of user systems by commercial launch. The product needs to survive the market demands.

• Interface

Why USB when there are lot many interfaces available in the industry? The answer lies within the advantages that USB provides over the other interfaces and the number of USB devices in the digital world. USB is an easy to use interface. From the developer’s as well as users point of view, it is has the advantages like ease of use and ease of accessibility.[5]

• Processor

In the proposed system, data transfer is the main task. Thus, the system needs a processor that will handle the data transfer and related processes speedily. The processor should have a USB host with drivers available for it. The operating system used by the processor plays a major in the development of the system. The processors from ARM7 were standing above all considering the demands of the embedded product. The ARM7 processors are made for embedded design. The flexibility and choices in interfaces as well as robustness provided by the ARM7 is very high as compared to others.

III. IMPLEMENTATION

A. Hardware

The hardware selection and implementation is very important. The main hardware of the system is the processor. As the system will be used by different users, so along with the processor the other user interfaces will be needed. Here, in the system, Touch Screen is used to provide menu driven system that will put the options on screen.
USB 2.0

USB was initially designed to be an interface for communicating with many types of peripherals without the limits and frustrations of older interfaces. It is a very useful protocol designed for a computer to communicate with almost any type of peripheral. The VCC and Ground are the power supply signals to the device. D+ and D- are the data lines through which actual data transfer takes place. Both D+ and D- are bidirectional lines, both carrying data in a single direction at a time and data is modulated into differential voltage levels to be transferred over them. The directions of D+ and D- are reversed in TDM manner to transmit and receive data. USB is a master-slave bus with one master and multiple slaves. The master is called a host and the slaves are the peripherals. Only the host has the ability to initiate the data transfers; the slaves only respond to the host's instructions—they never initiate transfers. The communication made by USB is based on logical channels—known as "pipes". The connections established are 1 to 1 for the endpoints in pipes. A USB device can have 32 endpoints—two of which are reserved. So a total of 30 are present for normal use.[3]

The data transfer is having four types:
1. **Interrupt transfers**: for the devices needing quick but guaranteed response (e.g. pointing device)
2. **Isochronous transfers**: For some fixed data rate but data loss may take place (e.g. audio, video)
3. **Control transfers**: used for simple status check.
4. **Bulk transfers**: uses available bandwidth with no fixed data rate (e.g. file transfer).

Depending on the type of data transfer, there are two types of pipes: stream and message. The stream pipe is connected to a unidirectional endpoint for the interrupt, isochronous and bulk data transfer modes. The message pipe is connected to bidirectional endpoint for control data transfer.

**ARM 7**

Data transfer is the main task in the system. Thus, a processor that will handle the data transfer processes speedily is needed. The processor should have a USB host with USB device drivers installed in it for plumbing in a mass storage device which will be used to transfer data from one device to other.

The Raspberry Pi has a Broadcom BCM2835 system on a chip which includes an ARM7 700 MHz processor.[10]

**Development Board**

Host controller can be found in market in two types: Dedicated IC for USB Host, Slave and devices operations and USB Host controllers included in microcontrollers. The second most important criteria to look for this presence of two USB ports on the Host Controller to avoid the use of buffer and extra hardware. It contains an Operating System and also device drivers to interact with the device. It will be responsible for creation of endpoints, establishment of pipes, initiating transfers, enumeration etc. The Raspberry Pi has a dual port USB host which is very important for the system. The USB host Controller Interface provides full speed serial communication ports at a baud rate of 12 Mbits/sec.

**LCD**

To provide User interface, the bridge uses the LCD (Liquid Crystal Display). The contents of the mass storage device are displayed on the LCD. This helps the user to view and select the files or folders of interest from the USB device. Also the options like select, copy for data transfer are put on to the LCD.

**Software**

The OS is needed for the system as all the initializations, drivers, and data and flow control, error handling, resource sharing as well as multitasking is possible only with the help of a good operating system. A Linux system can actually be adapted to work with as little as 256 KB ROM and 512 KB RAM. So it's a lightweight operating system to bring to the embedded market. Drivers and other features can be either compiled in or added to the kernel at run-time as loadable modules.

### IV. ARCHITECTURE OF THE SYSTEM

As shown in figure 1, the architecture of the system mainly consists of a processor, a LCD and Touch Screen. It is the heart of the system. It is important unit of the system because it reads data which we programmed and stored into its ROM, then it execute it. The processors from ARM7 were standing above all considering the demands of the embedded product. The ARM7 processors are made for embedded design. The flexibility and choices in interfaces as well as robustness provided by the ARM7 is very high as compared to others. Data transfer is the main task in the system. Thus, a processor that will handle the data transfer processes speedily is needed. When we select the data with touch facility the data is read from source and write to the destination drive. The USB host allows the processor to gain the control of the USB devices. Various commands are used to access the information such as files, folders, total size and other system related information of the USB device. Once the USB mass storage device is connected, the contents of the USB device are displayed on the LCD.
• **Implementation Algorithm**
  - Select the suitable development board.
  - Port the OS and install the USB device driver.
  - Connect the USB device to check functionality of the USB device.
  - Interface the LCD as a User interfaces.
  - Check the communication between the USB device and the board.
  - Explore the device contents on LCD.
  - Select a particular file, and by using the option COPY, copy that file to destination device using touch screen.
  - The selected file is then copied into destination USB device that is connected in one of the two USB ports.
  - If another copy operation is to be performed, go to step v.
  - Terminate the process.

V. WORKING OF THE SYSTEM

a) **System Flow**

The system allows the user to select files or folders/directories for copying from a source USB drive to a user selectable directory in the destination USB drive. The USB drive has four pins. The VCC and Ground are the power supply signals to the device. D+ and D- are the data lines through which actual data transfer takes place. Both D+ and D- are bidirectional lines, both carrying data in a single direction at a time and data is modulated into differental voltage levels to be transferred over them. The directions of D+ and D- are reversed in TDM manner to transmit and receive data. It is connectivity with peer-to-peer communication.

![System flow diagram](image)

Figure shows the system setup while figure shows the general block diagram of the system. The system allows the user to select files or folders/directories for copying from a source flash drive to a user selectable directory in the destination flash drive. In addition, the system is able to check if there is sufficient memory space for the file/folder/directory to be copied onto the destination USB drive; if not, the system requests the user to delete some files or folder/directories to free some memory in the destination USB drive.

b) **Execution of the task**

First step show the initialization of the system. Next step defines the initialization of the USB peripheral and system components. The steps ahead define the exploring and selecting the operation. The second last step shows the operation completion whereas the last step shows the termination of the process. The flowchart of operation is as shown in the figure below.

![Flowchart of Task execution](image)

VI. TEST AND RESULTS

The 1st step is to boot process of the embedded linux. The next test performed was to check whether the hardware can recognize the USB device or not.

![USB mass storage device explored](image)

Going further, the proposed system was also tested to copy a file from one USB device connected at one end to a device at the other port using the linux commands like ls /media, cp.[6] The “minicom” facility in the Linux kernel acts in the same manner as the “HyperTerminal” in the Windows.[7] One important test was also carried out to check for the bidirectional data transfer in the half duplex mode. To do so, a single file from mass storage was copied to and fro using the minicom command.
VII. ADVANTAGES

• **Battery Operated**
  As the whole system operates on the 5 V supply and the core operates on 3.3V and also is very low power consuming so the system can be made to operate on the battery.

• **Portable**
  The proposed system can be made portable by means of making it a standalone platform. The processor along with the peripherals makes it to work independent of PC. One can carry out the data transfer anywhere, anytime.

• **Power Optimization**
  As both the processor and the USB2.0 specifications are designed to keep the lowest possible power consumption, the power optimization is done by using it only when it is necessary.

VIII. CONCLUSION

The USB to USB gives a concept to study the USB protocol as well as the working of the USB host along with the processor in group. A set of basic requirements were defined and used for design work of the USB concept based on practical tests and results. The ease of use and portability makes it a powerful but unique tool to do the data transfer. The advantage of this device is that it is battery operated so there is no need of power supply connection and data transfer can take place at any place. Currently we have design the system for transfer of data up to 2GB only, but it can be increased by proper selection of the bus. As a result, a more generalized but unique concept of USB to USB Bridge will be developed. In this respect, this study has given us a remarkable insight into the future of development of high speed USB 3.0 Bridge.

IX. FUTURE SCOPE

While working on the development of the system and exploring the peripherals that can be interfaced with the ARM 7 we found that with little modification in the project several new features could be added. Following are the things that can be done with few modifications.

1. Add USB host capability to embedded products.
2. Interface USB Flash drive to MCU/PLD/FPGA.
3. Using Bluetooth in our device, We can connect with any Bluetooth enable devices making the data transfer wireless.

REFERENCES