

## Pen Drive to Pen Drive and Mobile Data Transfer Using ARM

<sup>1</sup>Mr.V.S.Gawali ,<sup>2</sup> Mr.A.M.Agarkar  
<sup>1,2</sup> Shri Sant Gajanan Maharaj College Of Engg.Shegaon

**ABSTRACT:** Generally, we used to transfer data between two pen drives by using laptops or desktops. But it is not always possible to carry such a large size device to the particular location. So to overcome this problem, we are designing a hardware which is more compact to carry anywhere. With the help of this project we can not only transfer the data but also we can see the transfer of the particular file which we want to send by using LCD display. In our project we are transferring the data between two pen drives without using any computers or laptops. We have designed a project which is known as pen drive to pen drive data transfer, and pen drives to mobile in which we will be transferring the data between two pen drives and pen drives to mobile using ARM processor. In the block diagram we can observe, whenever we insert the pen drive into the USB port then a signal will be sent to the arm processor indicating that source pen drive is inserted so now arm processor will start fetching the data from the source pen drive into the buffer and arm processor waits for the signal from destination pen drive. When arm processor gets the signal from the destination pen drive now arm processor is ready to transfer the data between those. Only the arm processor should get the input from external hard key from the user, once the user press the hard key the arm processor gets the information to transfer the data between two pen drives. While transferring the data the led blinking rate will be increased when data transfer is completed then led will stop blinking.

### I. INTRODUCTION

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. For this user has to first find a computer then wait for it to boot up, then plug in his device, and then transfer the data. Different types of USB flash devices are used now-a-days. It is not necessary that all of these devices are supported by the computer and the operating system and their device drivers are available and installed.

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. More- over, transferring data via a computer involves a lot of power to be wasted, since the computer has to be entirely functional before it can transfer data. 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 flash device into another. Our project here can provide a valuable solution to all problems faced by person in above situations. Our aim is to build a small and handy device to transfer data from one USB Flash device to another.

The device will have following features:-

1. Small, lightweight and handy device
2. Powered by 9V battery
3. Support for all USB Flash devices formatted with FAT32 file System
4. Support for USB 2.0
5. Plug and Go function

#### 1.1 Universal Serial Bus (USB)

USB was initially designed to be an interface for communicating with many types of peripherals without the Limits and frustrations of older interfaces. Every recent PC includes USB ports that can connect to standard peripherals such as keyboards, mice, scanners, cameras, printers, and storage drives. It is a very useful protocol designed for a computer to communicate with almost any type of peripheral.

Some of its benefits for users include:

1. Single Interface for many devices.
2. Considerably high Data rate.
3. Automatic Configuration.
4. Easy connection.
5. Hot Pluggable.
6. No user Settings.
7. Frees Hardware Resources for other devices.

8. No extra Power Supply needed.
9. Low Cost and Low Power consumption.  
Benefits for Developer include.
  - Operating system support.
  - Peripheral support.
  - Open source support available online.
  - Versatility.

#### 1.1.1 Inside USB

As mentioned above, the connection comprises of four terminals.

1. Vcc
2. D-
3. D+
4. Ground

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 deferential 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.

Low Speed has data rate of 1.536Mbps/s, Full Speed has 12Mbps/s, High Speed has 480Mbps/s, and Super Speed has 5Gbits/s.

## II. LITERATURE REVIEW

1. IEEE paper published by Ducloux, J, Petrashin, P, in 2011 titled by Embedded USB dual role system for communication with mobile devices
  2. On the link EFY Times.com an innovation was published a title USB TO USB data transfer device by Mr.Shrnik Shikhare. Who have created a device which can transfer data from one USB to Another USB without using PC.
  3. IEEE paper published by Zhang Xiaoyan; Tie Yong in 2010 titled by Design and Realization of an embedded storage system based on LPC2387 microprocessor
  4. IEEE paper published by Ducloux, J, Petrashin, P, in 2012 titled by an Embedded USB dual role System Integrated for mobile devices
- 2.1Idea Innovation

The idea of this project was taken, looking at the problems faced in daily life of Flash Drive Users to transfer the data from one disk to another. 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 as small and handy as possible. Moreover, transferring data via a computer involves a lot of power to be wasted. There are numerous types of data transactions that are being carried out through these devices. However to operate these devices most of the times an operating system is required which calls for the hosts to be extremely complicated system hence accessing these devices requires complicated hardware, hence a controller that can handle the data transfer and initiates the USB transactions was searched.

## III.IMPLEMENTATION DETAILS

### 3.1 System block diagram

The study of the block diagram is very helpful to the designer.

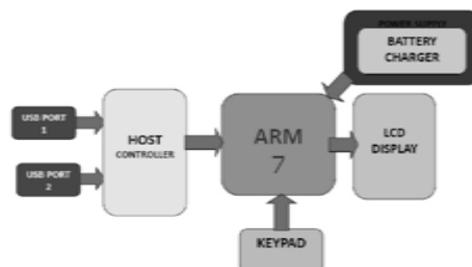


Fig.1 System Block Diagram

ARM (LPC 2138): 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 executes it and controls the Display as well as Vinculum VNC1L Embedded USB Host controller. When key is press it read the data from source pen drive and writes to destination pen drive.

LCD Display: Frequently, a C program must interact with the outside world using input and output devices that communicate directly with a human being. One of the most common devices attached to an ARM7 is an LCD display. Some of the most common LCDs connected to the ARM7 are 16x2 and 20x4 displays. This means 16 characters per line by 2 lines and 20 characters per line by 4 lines, respectively. It displays events taken into microcontroller step by step alphanumerically. So it is convenient us to know what is currently running in the system. It displays the start and finish of the data transfer.

Host Controller (VNC1L): It recognizes the input peripheral as host or slave according to firmware loaded in it. The sub module physically interfaces with the USB flash drives and is responsible for converting raw data and information to their proper NRZI encoding as specified by the USB technical specifications. Further-more, the sub module is capable of encoding or decoding the incoming NRZI data from the USB flash drives and forwards it to their respective sub modules for further processing. It handles all types of transfer between source destinations. We have USB Host controller which has two independent USB 2.0 low speed/high speed USB Host ports. Individual ports can be configured as host or slave.

USB Connectors: There are two types of USB connector are available in market, here we use a Type-A USB connector which is more popular use in almost all application. Another reason to use it is USB flash drive has Type-A USB male connector.

### 3.2 USB Host Controller IC - VNC1L

To begin with, selection of the USB host controller IC is to be done. Host controllers can be found in market, in two types viz. Dedicated IC for USB Host, Slave and device operations, and USB Host controllers included in microcontrollers. The second most important criteria to look for is the presence of two USB ports on the Host Controller to avoid the use of buffer and extra hardware. One more feature to look for was support for FAT-32 file system included hardwired on the Host controller to avoid complications in microcontroller code to decode it. A dedicated USB Host Controller from Vinculum was found- VNC1L. It has got following features over host controllers included on chip of General Purpose Microcontrollers:

- Two independent USB 2.0 Low speed / Full speed USB Host / Slave ports
- Entire USB protocol handled on the chip
- 8/32bit MCU core
- Twin DMA controllers for hardware acceleration
- 64kB embedded Flash ROM Prog. Memory, 4kB Data SRAM
- Interface to MCU / PLD / FPGA via UART/ FIFO/SPI interface
- Upto 28 GPIO interface pins for data I/O and command monitor interface
- 3.3V operation with 5V safe inputs

Selection criteria for VNC1L

- Twin Host Controllers inbuilt
- Availability of Twin DMA controllers
- Availability of ready VDAP
- NCU- 32bit Numeric Co-processor Unit for handling of FAT32 file system
- Handles entire USB protocol On Chip.

USB Flash Disk: These are source and target Devices. They will be connected to the USB Host Controller IC via the USB interface. Both the devices can be accessed simultaneously. The memory on them is in the form of Flash which is accessed by an on board hardware. These devices have also got protocol handling hardware included on device.

Before making the devices available for use, the Operating system formats the device with a file format. FAT-8, FAT-16, FAT-32, NTFS, are some file systems available in Microsoft Operating Systems. Majority of USB Flash devices are formatted in FAT-32 file system. There may sometimes be Partitions on the drive. Data encrypted with all above formats can be accessed with an efficient Firmware on the USB Host Controller.

## Pen Drive to Pen Drive and Mobile Data transfer using ARM

USB Host Controller IC: This is a microcontroller which has an interface to connect a USB device. It has a Central Processing Unit of its own. The requirement here is of 2 USB ports. It is the main device that handles the USB protocol and does the jobs of the USB host. It may contain 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. We can preferably look for capability like DMA so that CPU remains free while Data transfer

### 3.3 VNC1L Block Diagram

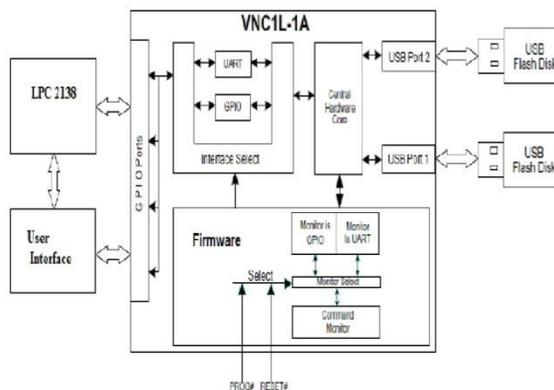


Fig.2 VNCIL block diagram

## IV. EXPERIMENTATION

### 4.1 system flowchart.

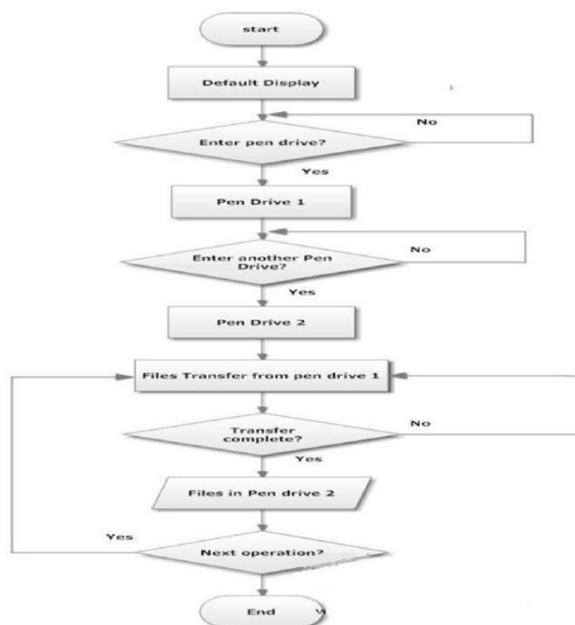


Fig.3 System flochart

### 4.2 Components Used

1. ARM processor
2. USB host controller IC- VNC1L
3. Power supply unit
4. Computer interface
5. Graphical LCD

## **V. RESULT ANALYSIS**

Problems are stepping stones towards success. Once we start working with the hardware we are bound to face some obstacles in our way to reach our goal. While working on the hardware of the project we initially had opted for an ARM board combined with the general purpose board. The idea was opted because we were new with the use of the VNC11 IC and so the method we opted was the try and error method. We opted the general purpose board so that we can easily demount the components if something goes wrong. Also it was used so that future modifications could be made easier if required or necessary. We initially mounted all the circuit components related to the ARM board and so far the work was going as we had planned. The problem arised as we mounted the VNCL1 IC, the board became bulky and it became difficult for us to mount all the components that were to be interfaced between ARM and VNCL1. Size of the board became the reason behind the bulkiness. We had to compromise with the spacing of the components and that made the work very untidy and difficult for us to distinguish between various components. So finally we decided to switch the boards. We then designed a board using ORCAD 9.0. This board was very specific with what we needed and was the correct decision which leads us to the perfect working of our project.

## **VI. CONCLUSION**

Transferring the data through USB in today's scenario is the most common task. But the problem is that for transferring the data to a personal computer or laptop is difficult if u don't have any of them. It is affordable to purchase a USB data drive than purchasing a laptop or PC. Therefore we came up with a handled battery operated affordable device which can transfer the data between two USB data drives without the help of PC or laptop.

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.

## **VII. 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.
4. Keypad and Graphical LCD could be replaced by touch screens which could make human work easier by drag and drop method.

## **REFERENCES**

- [1] [ieeexplore.ieee.org](http://ieeexplore.ieee.org).
- [2] Philips Semiconductors LPC2148 datasheet
- [3] FTDI Inc, [www.ftdichip.com/document](http://www.ftdichip.com/document) Reference No.: FT-000006v. Vinculum Firmware User Manual Version 2.05.
- [4] [www.keil.com](http://www.keil.com)
- [5] [www.sciencedirect.com](http://www.sciencedirect.com)
- [6] Future Technology Devices International Ltd. USB Data Packet Structure.
- [7] [www.usb.org](http://www.usb.org)
- [8] [www.alldatasheets.com](http://www.alldatasheets.com)
- [9] [EFY Times.com](http://EFYTimes.com)
- [10] [www.google.com](http://www.google.com)
- [11] ARM Assembly Language: Fundamentals and Techniques by William Hohl.
- [12] ARM system developers guide by Andrew N.Sloss.
- [13] Embedded systems by Raj kamal
- [14] ARM Architecture Reference Manual by David Seal.
- [15] ARM Assembly Language: Fundamentals and Techniques by William Hohl.
- [16] ARM Microcontroller Interfacing: Hardware and Software by Warwick A. Smith.
- [17] Microprocessors, pc hardware and interfacing by N.Mathivanan.