



## Data Transfer between Two Pendrives without PC

Rupali C. Bachalkar, Shrutika D. Durge, Purvaja V. Pote, Sapana V. Ajmire, Prof. Reetesh V. Golhar  
Department of Electronics Engineering, RTMNU, Datta Meghe Institute of Engineering Technology & Research,  
Sawangi Meghe, Wardha, Maharashtra, India

**Abstract:** *Now day's portability is most important. So to achieve this, we are designing such a system which can carry anywhere. Using this system 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. Now a days to transfer a data between two pen drive we use PC or laptop, but it is not always possible to carry such a large device only for the data transfer. So to overcome this problem we design a system which is more compact. In our project we are transferring the data between two pen drives without using any computers or laptops. As shown in the block, whenever we insert two pen drive in to the USB port of Raspberry Pi, this can be done by giving the command to the processor. The processor indicates that the pen drive is inserted successfully till the user can not send any command to processor, the operation cannot start. After sending the particular command to processor, the processor will start fetching the data from source pen drive into buffer and the ARM processor wait for the signal from destination pen drive. When the processors get the signal from destination pen drive, so the data transfer operation begins. Only the ARM processor should get the external hard key input signal from the user. Once the user press the hard key the ARM processor get the information to transfer the data between two pen drive. While transferring the data the LED blinking rate will be increased when data transfer is completed then LED will stop blinking. This total operation is performed on the Raspberry Pi board by using a "Raspbian wheezy" operating system.*

**Keywords:** *ARM :-Advance RICS Machine, USB :- Universal Serial Bus, LED :-Light Emitting Diode, LCD :- Liquid Crystal Display*

### I. INTRODUCTION

Today the need for the portable devices is well known to us. We can easily find the USB and its applications everywhere around us. The applications of the USB are computer peripherals such as keyboard, pointing devices, digital cameras, printers, portable media players, disk drives and network adapter, both to communicate and to supply electric power. It has become common place on other devices, such as smart phones, PDAs and video game consoles. USB has effectively replaced a variety of earlier interfaces, such as serial and parallel ports, as well as separate power chargers for portable devices.

But the main disadvantage of USB devices is that it requires the use of PC for their operation. Carrying a PC 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 device into another. So a solution is provided by means of implementation of a small device that carries out the required task. The small footprint and ease of portability makes it a choice for the data transfer. This device will help the user to select a particular datafile from the mass storage device connected to one of the ports and transfers it to the other mass storage device using some controls like list, copy provided on the front panel.

### II. BLOCK DIAGRAM

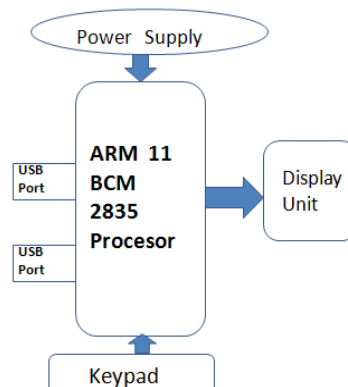


Fig 1. Block Diagram of Our Project

### III. EXPLANATION

The block diagram gives a general idea to design our project. The over view of our project for designing purpose is given in the block diagram. The important hardware part which are used in the system are ARM processor which is BCM 2835, Micro USB port gives 5v supply ,LCD display, keypad, and USB interface. The main part of our system is the ARM processor; this is the heart of our project which is used for interfacing. The ARM processor provides the required functionality as per our requirement, so it is main intention to choice the ARM processor. The 2.0 USB connector are connected to the ARM processor this two USB port are used to connect the other device such as keyboard, mouse, and external hubs. So as the Raspberry pi has the two USB port and it has the central host controller. The host controller manages attachment and removal of USB devices Manage data flow between host and devices Provide and manage power to attached devices Monitor activity on the bus.For communication it's most important to connect the two pen drives into the USB ports of controller. When it connect to the system first it does the job of initialization and then we provide the option such as copy, paste, cut etc. by using switches for dealing with the data. After completing this procedure the particular file name in the pen drive are shown in LCD display. To provide the option such as cut, copy, paste is provided with the help of switching kit in the replacement of the normal keyboard. When we pressed the specific key it is sensed by the ARM processor as an interrupt, it provides the required operation as per we send the information, then the output our data on LCD and then actual data transfer. For LCD initialization and to make LCD in the working condition, the graphical LCD drives are installed on the ARM processor. LCD 20\*4 display are connected which provide 20 characters per line by 4 lines. We have used the Linux OS for system which is more secured, open source and easily available for users. Linux kernel is ported in ARM's memory which manages device drivers and system libraries.

### IV. HARDWARE DESCRIPTION

#### 1. RASPBERRY PI

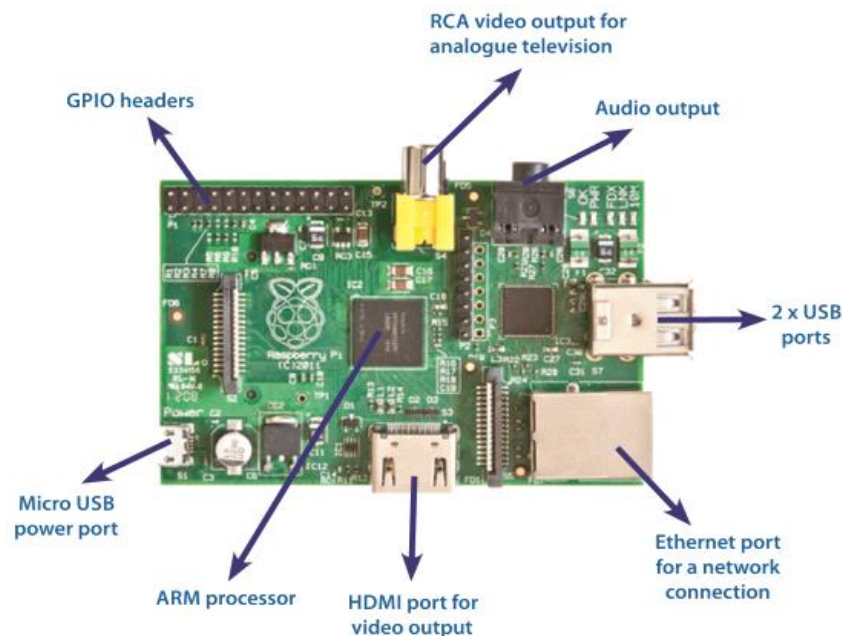


Fig 2. Raspberry Pi kit

The hardware description of the project is as follows:The Raspberry Pi is a small computer about the size of a credit card. It was developed in the UK by the Raspberry Pi Foundation with the hope of inspiring a generation of learners to be creative and to discover how computers are programmed and how they function.The device uses the ARM processor which does most of the hard work in order to run the Raspberry Pi.. The reason for this is that ARM processors are extremely efficient and fast when used in small devices. This makes the ARM processor the obvious choice for the Raspberry Pi.Even though the Raspberry Pi is a computer it does not have a hard drive like traditional computers, instead it relies on the SD card for the starting up and storing of information. For the Raspberry Pi the SD card does the same job as a hard drive does in a traditional computer. The SD card must contain the operating system, programs and the data needed to run the Raspberry Pi. The operating system tells the Raspberry Pi how to function, how to handle any input from the user and how to manage programs when they are running.

**Micro USB power port:-**The micro USB power port is used to power the Raspberry Pi device.

**HDMI Port:-**The HDMI output is used to plug into a modern television or monitor.

**Ethernet port:-**The Ethernet port is used to connect the raspberry Pi to the internet or a local network.

**USB port :-**USB 2.0 ports are used to plug in a keyboard, mouse, external hubs etc.

**Audio output:-**The audio output can be used to plug into an external amplifier or an audio docking station.

**GPIO headers:-**The GPIO headers are used to connect the Raspberry Pi to other hardware device. For example, they can be used to connect to LED'S, motors and other electronics components.

**RCA video output:**-The video output is used to connect to an older type television.

**ARM processor:**-The ARM processor can be thought of as the brains of the Raspberry pi.

## 2. USB

Ease of use was a major design goal for USB, and the result is an interface that's a pleasure to use for many reasons:

**One interface for many devices :-**USB is versatile enough for just about any standard PC peripheral function. Instead of having a different connector and cable type for each peripheral function, one interface serves many.

**Automatic configuration.**When a user connects a USB device to a PC, the operating system detects the device and loads the appropriate software driver. The first time the device connects, the operating system may prompt the user to insert a disc with driver software, but other than that, installation is automatic. Users don't need to reboot before using the device.

**Easy to connect.**A typical PC has multiple USB ports, and hubs make it easy to add ports without

**Wireless options.**USB originated as a wired interface, but technologies are now available for wireless communications with USB devices. It's opening up the PC.

## V. SOFTWARE DESCRIPTION

### 1. LINUX

Linux was originally developed as a free operating system for Intel x86-based personal computers. It has since been ported to more computer hardware platforms than any other operating system. It is a leading operating system on servers and other big iron systems such as mainframe computers and supercomputers. Linux supports a vast variety of hardware devices, probably more than any other OS. Linux supports a huge variety of applications and networking protocols. Linux is scalable, from small consumer-oriented devices to large, heavy-iron, carrier-class switches and routers. Linux can be deployed without the royalties required by traditional proprietary embedded operating systems. Linux has attracted a huge number of active developers, enabling rapid support of new hardware architectures, platforms, and devices.

### 2. PYTHON

Python is an interpreted, interactive, object-oriented programming language. Python is a widely-used high level programming language. Its elegant syntax allows you to clearly define application behavior using fewer lines of code than would be required in other languages like VB. It supports multiple programming paradigms including imperative, functional and object oriented styles, allowing a wide range of tasks to be performed.

#### Programme For Data Transfer Between Pendrives

```
import RPi.GPIO as GPIO
import time
import smtplib
import subprocess
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(4, GPIO.IN)
GPIO.setup(27, GPIO.IN)
GPIO.setup(17, GPIO.OUT)
# Define GPIO to LCD mapping
LCD_RS = 12
LCD_E = 16
LCD_D4 = 25
LCD_D5 = 24
LCD_D6 = 23
LCD_D7 = 21
# Define some device constants
LCD_WIDTH = 20 # Maximum characters per line
LCD_CHR = True
LCD_CMD = False
LCD_LINE_1 = 0x80 # LCD RAM address for the 1st line
LCD_LINE_2 = 0xC0 # LCD RAM address for the 2nd line
LCD_LINE_3 = 0x94 # LCD RAM address for the 3rd line
LCD_LINE_4 = 0xD4 # LCD RAM address for the 4th line
# Timing constants
E_PULSE = 0.00005
E_DELAY = 0.00005
def main():
    # Main program block
```

```
GPIO.setmode(GPIO.BCM) # Use BCM GPIO numbers
GPIO.setup(LCD_E, GPIO.OUT) # E
GPIO.setup(LCD_RS, GPIO.OUT) # RS
GPIO.setup(LCD_D4, GPIO.OUT) # DB4
GPIO.setup(LCD_D5, GPIO.OUT) # DB5
GPIO.setup(LCD_D6, GPIO.OUT) # DB6
GPIO.setup(LCD_D7, GPIO.OUT) # DB7
deflcd_init():
    # Initialise display
    lcd_byte(0x33,LCD_CMD)
    lcd_byte(0x32,LCD_CMD)
    lcd_byte(0x28,LCD_CMD)
    lcd_byte(0x0C,LCD_CMD)
    lcd_byte(0x06,LCD_CMD)
    lcd_byte(0x01,LCD_CMD)
deflcd_string(message):
    # Send string to display
    message = message.ljust(LCD_WIDTH, " ")
    for i in range(LCD_WIDTH):
        lcd_byte(ord(message[i]),LCD_CHR)
deflcd_byte(bits, mode):
    # Send byte to data pins
    # bits = data
    # mode = True for character
    #      False for command
    GPIO.output(LCD_RS, mode) # RS
    # High bits
    GPIO.output(LCD_D4, False)
    GPIO.output(LCD_D5, False)
    GPIO.output(LCD_D6, False)
    GPIO.output(LCD_D7, False)
    if bits&0x10==0x10:
        GPIO.output(LCD_D4, True)
    if bits&0x20==0x20:
        GPIO.output(LCD_D5, True)
    if bits&0x40==0x40:
        GPIO.output(LCD_D6, True)
    if bits&0x80==0x80:
        GPIO.output(LCD_D7, True)
    # Toggle 'Enable' pin
    time.sleep(E_DELAY)
    GPIO.output(LCD_E, True)
    time.sleep(E_PULSE)
    GPIO.output(LCD_E, False)
    time.sleep(E_DELAY)
    # Low bits
    GPIO.output(LCD_D4, False)
    GPIO.output(LCD_D5, False)
    GPIO.output(LCD_D6, False)
    GPIO.output(LCD_D7, False)
    if bits&0x01==0x01:
        GPIO.output(LCD_D4, True)
    if bits&0x02==0x02:
        GPIO.output(LCD_D5, True)
    if bits&0x04==0x04:
        GPIO.output(LCD_D6, True)
    if bits&0x08==0x08:
        GPIO.output(LCD_D7, True)
    # Toggle 'Enable' pin
    time.sleep(E_DELAY)
    GPIO.output(LCD_E, True)
    time.sleep(E_PULSE)
    GPIO.output(LCD_E, False)
```

```
time.sleep(E_DELAY)
if __name__ == '__main__':
    main()
    # Initialise display
    lcd_init()
    while True:
        ifGPIO.input(4)==False:
            # Send some test
            lcd_byte(LCD_LINE_1, LCD_CMD)
            lcd_string("Data Copping.....")
            time.sleep(1)
            GPIO.output(17,GPIO.HIGH)
            subprocess.call("mount /dev/sda1 /mnt/sda1",shell=True)
            time.sleep(3)
            subprocess.call("mount /dev/sdb1 /mnt/sdb1",shell=True)
            time.sleep(3)
            subprocess.call("cp -R /mnt/sdb1 /mnt/sda1",shell=True)
            time.sleep(3)
            lcd_byte(LCD_LINE_1, LCD_CMD)
            lcd_string("Congratulations ")
            lcd_byte(LCD_LINE_2, LCD_CMD)
            lcd_string("Copy is done... ")
            GPIO.output(17,GPIO.LOW)
            time.sleep(4)
            ifGPIO.input(27)==False:
                lcd_byte(LCD_LINE_1, LCD_CMD)
                lcd_string("Ejecting..... ")
                GPIO.output(17,GPIO.HIGH)
                subprocess.call("umount /dev/sda1",shell=True)
                time.sleep(1)
                subprocess.call("umount /dev/sdb1",shell=True)
                time.sleep(4)
                GPIO.output(17,GPIO.LOW)
            else:
                lcd_byte(LCD_LINE_1, LCD_CMD)
                lcd_string(" Wel-Come ")
                lcd_byte(LCD_LINE_2, LCD_CMD)
                lcd_string("For copy press 1")
                lcd_byte(LCD_LINE_3, LCD_CMD)
                lcd_string("For eject press 2")
                GPIO.output(17,GPIO.LOW)
```

## **VI. CONCLUSION**

The project undertaken satisfies the needs of the current generation that requires portable means of carrying data transfers. The important thing is data transfer is done without the involvement of PC. It also provides much security as Linux is a much secured Operating System. It has been developed by integrating features of all the hardware component & software used, using highly advanced raspberry pi board & with the help of growing technology the project has been successfully implemented. The advantage of this device is that it is battery operated, so there is no need of connecting power supply & data transfer can be take place at any time.

## **VII. FUTURE SCOPE**

The following are the further developments that can be done based on the project.

- USB to printer interface.
- Touchscreen based GUI.
- Wireless connectivity using Bluetooth.
- It can be embedded into various devices like TV, DVD player, etc.

## **REFERENCES**

- [1] [www.raspberrypi.org](http://www.raspberrypi.org)
- [2] [ieeexplore.ieee.org](http://ieeexplore.ieee.org).
- [3] [www.usb.org](http://www.usb.org)
- [4] [www.alldatasheets.com](http://www.alldatasheets.com)
- [5] [www.google.com](http://www.google.com)

- [6] “Pen Drive to Pen Drive and Mobile Data Transfer Using ARM” IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)
- [7] “USB to USB and Mobile Data Transfer Without Connecting to PC Using Arm Processor” International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 International Conference on Industrial Automation and Computing (ICIAC-12-13th April 2014)
- [8] P. Bapat, N. Lodh, R. Polas and S. Pulkurte, “USB TO USB Data Transfer Without Connecting To PC”, International Journal of Engineering Research & Technology (IJERT), 2(2), 2013.
- [9] IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 2, Issue 2, Apr-May, 2014 “Wired and Wireless Transmission of Data between Pen drives and Pen drives to Computer Using ARM”.