

Appendix A:

```
import java.net.*;

import java.util.*;

import java.io.*;

public class UDPFileTx

{

    public static DatagramSocket ds;

    public static int buffer_size=100;

    public static int serverport=999;

    public static int clientport=666;

    public static byte buffer[]=new byte[buffer_size];

    public static void FileTx(String FileName) throws Exception{

        byte b[]=new byte[1500];

        InetAddress iaddress= InetAddress.getByName("131.200.2.10");

        int i=0;

        int FileNameLen = FileName.length();

        int Pos;

        int c;

        File f1= new File (FileName);

        Long file_Length = f1.length();
```

```
double pcd = (file_Length/1450.0);

double pccd=Math.ceil (pcd);

Double pcdd=new Double(pccd);

int PacketCount=pcdd.intValue(); //Packet Count

/* Filename Packet */

b[0]='F' ; //Type of Packet

b[1]=(byte)PacketCount; //Packet Count

b[2]=(byte)FileNameLen; //Length of the file name

for(i=0;i<FileNameLen;i++){

b[i+3]=(byte)FileName.charAt(i);

}

b[i+3]='\r';

b[i+4]='\n';

ds.send(new DatagramPacket(b,b.Length,iaddress,clientport)); //Filename
Packet

/* Data Packet */

FileInputStream fis = new FileInputStream(f1);

b[0]='D';

b[1]=(byte)(PacketCount);

pos=3;

while ((c=fis.read())!= -1){
```

```
b[poss++]= (byte)c;

}

fis.close();

b[2]=(byte)(poss-3);

b[pos++]='\r';

b[pos++]='\n';

ds.send(new DatagramPacket(b,b.Length,iaddress,clientport)); //Data
Packet sending

} //end of FileTx

public static void main(String args[]){

try{

    if(args.Length==1){

        System.out.println("Tx starts .....");

        ds=new DatagramSocket(serverport);

        FileTx(args[0]);

        System.out.println("Tx ends");

    }

    else

        System.out.println("Usage : java UDPFileTx test.txt\n");

    }catch(Exception e){ }

} // end of main
```

```
    } //end of class , UDPFileTx
```

Appendix B:

```
#include<stdio.h>
```

```
#include<string.h>
```

```
#include <fcntl.h>
```

```
#include<termios.h>
```

```
#include<sys/socket.h>
```

```
#include<net/ethernet.h>
```

```
#define MAXLINE 1024
```

```
char c[MAXLINE + 1],Data [MAXLINE*10],FileName[256];
```

```
int port_open(void)
```

```
{
```

```
int fd;
```

```
/* open com port 1 (COM1) */
```

```
fd=open("/dev/ttyso",O_RDWR | O_NOCTTY | O_NDELAY);
```

```
if (fd == -1)
```

```
{
```

```
perror("unable to open the port ./dev/ttyso");
```

```
}
```

```
else
```

```
{

fcntl(fd,F_SETFL,0);

}

return(fd);

} // end of port_open

void port_config(int fd)

{

struct termios settings ;

/* get the present portsettings */

tcgetattr(fd,&settings) ;

/* set the baud rates to 9600 */

cfsetispeed (&settings,B9600) ;

cfsetospeed (&settings,B9600) ;

/* set local mode and enable the receiver */

settings.c_cflag |= (CLOCAL | CREAD );

/* configure the port setting as No parity (8N1) */

settings.c_cflag &= ~PARENB ; // parity bit

settings.c_cflag &= ~CSTOPB ; // stop bit

settings.c_cflag &= ~CSIZE ; // Bit mask for data bits

settings.c_cflag |= CS8 ; // 8 data bits
```

```
/* Set the above values to the port */

tcsetattr (fd,TCSANOW,&settings);

} // end of port_config

void write_data(int fd,char *c,int Datalength)

{

int n,i ;

n= write(fd,c,Datalength);

if (n<0)

{

fputs ("write() of data failed! \n",stderr);

}

} // end of write_data

void port_close(int fd)

{

close(fd);

} // end of port_close

void read_ethernet_data()

{

int sockfd,n,i,j,k ;

int PacketCount;
```

```

int FileNameLength ;

int PacketNumber ;

int Lod1,LengthofData;

char recvline [MAXLINE +1];

if ((sockfd =
socket(AF_INET,SOCK_PACKET,htons(ETH_P_ALL)))<0)

printf("Socket error\n");

while ((n= read (sockfd , recvline , MAXLINE )) >0)

{

j=0 ;

for (i=0;i<n;i++) {

if(i>41) // user data of the frame starts at 42 byte

{

c[j] = recvline[i];

j++;

}

}

if (c[0] == 'F') //Filename packet

{

PacketCount= c[1];

FileNameLength = c[2];

```

```
for (i=0;i<FileNameLength;i++){  
  
    FileName[i] = c[i+3];  
  
}  
  
FileName[i]='\0';  
  
k=0;  
  
} // end of Filename packet  
  
if(c[0] == 'D' ) //Data packet  
  
{  
  
    PacketNumber = c[1];  
  
    Lod1 = c[2];  
  
    LengthofData = Lod1;  
  
    for (i=0; i<LengthofData ;i++) {  
  
        Data[k] = c[i+3];  
  
        k++;  
  
    }  
  
} // end of data packet  
  
if (PacketCount == PacketNumber)  
  
    break ;  
  
}  
  
if (n<0)
```

```
printf ("Read error \n");

printf ("FileName : %s \n",FileName);

printf ("Data : ");

for (i=0;i<k;i++)

printf("%c" ,Data[i]);

printf("\n");

} // end of read_ethernet_data


void convert_data_send (char *EthernetData ,char packet ,int fd)

{

int i;

char SerialComData[256];

int EthernetDataLength, Length;

int PacketCount , PacketNumber;

EthernetDataLength = strlen (EthernetData);

PacketCount = 1;

if (packet == 'F') //filename packet

{

SerialComData[0] =0x02;

SerialComData[1] = 'F';
```

```
SerialComData[2] = PacketCount;

SerialComData[3] = EthernetDataLength;

for (i=0;i<EthernetDataLength;i++)

SerialComData [i+4] = EthernetData[i];

SerialComData [i+4] = 0x10;

SerialComData [i+5] = 0x0d;

SerialComData [i+6] = '\0';

PacketNumber = 0;

write_data (fd, SerialComData, EthernetDataLength+6);

} //end bof filename packet

if (packet == 'D' ) //data packet

{

while(EthernetDataLength >0)

{

SerialComData[0] = 0x02;

SerialComData[1] = 'D';

SerialComData[2] = PacketNumber+1;

SerialComData[3] = EthernetDataLength/256;

SerialComData[4] = EthernetDataLength%256;

if (EthernetDataLength >1024)
```

```
Length = 1024 ;

else

Length = EthernetDataLength;

for (i=0;i<Length; i++){

SerialComData[i+5] = EthernetData[i];

}

SerialComData[i+5] = 0x10;

SerialComData[i+6] = 0x0d;

SerialComData[i+7] = '\0';

write_data (fd, SerialComData, Length+7);

EthernetDataLength = Length;

PacketNumber++;

} // end of while

} // end of data packet

} // end of convert_data _send

int main (void)

{

int fd;

read_ethernet_data();

fd = port_open();
```

```
port_config(fd);

convert_data_send(FileName,'f',fd);

convert_data_send(Data,'D',fd);

port_close(fd);

return 0;

} //endof main
```

Appendix C:

```
#include <stdio.h>

#include <string.h>

#include <unistd.h>

#include <fcntl.h>

#include <errno.h>

#include <termios.h>

int port_open(void)

{

int fd;

/*open com port 1(com1) */

fd=open("/dev/ttyS0",O_RDWR | O_NOCTTY | O_NDELAY);

if (fd == -1)

{perror("unable to open the port : /dev/ttyS0");
```

```
}

else

{

fcnt1(fd,F_SETFL,0);

}

return (fd);

} //end of port_open


void port_config(int fd)

{

struct termios settings ;

/* get the present port settings */

tcgetattr(fd,& settings);


/*set the baud rates to 9600 */

cfsetispeed(&settings,B9600);

cfsetospeed(&settings,B9600);


/*set local mode and enable the receiver */

settings.c_cflag |= (CLOCAL | CREAD );


/*configure the port as no parity (8n1) */
```

```
settings.c_cflag &=~PARENB ; //parity bit

settings.c_cflag &=~CSTOPB ; //stop bit

settings.c_cflag &=~CSIZE ; // bit mask

settings.c_cflag  |= CS8 ; //8 data bits

/*set the above values to the port */

tcsetattr (fd,TCSANOW, &settings );

} // end of port_config


char* read_data( int fd)

{

char array [256];

char *ptr;

int nbytes,i;

ptr =array;

while (nbytes = read (fd ,ptr ,array + sizeof(array) - ptr -1))>0)

{

ptr += nbytes;

if ((ptr[-1] == '\n') || (ptr[-2] == 0x10))

break;

}
```

```
*ptr = '\0';

return (&array[0]);

} //end of read_data

void receive_packets (int fd)

{

char *str;

int i,j,k;

char FileName[256];

char Data [10240];

int PacketCount =1,PacketNumber ,LengthOfData;

int FileNameLength;

FILE *fp;

char ch ;

str = read_data(fd);

if (str[1] == 'F' ) // filename packet

{

PacketCount = str [2] - 48; // packet count

FileNameLength = str[3] - 48; // length of the file name

for (i=0;i< FileNameLength;i++)

FileName [i] =str [i+4];
```

```
FileName[i] = '\0';

K = 0;

printf ("File Name: %s\n", FileName);

}

for (i=0;i<PacketCount;i++)

{

str = read_data(fd);

if(str[1] == 'D') // data packet

{

PacketNumber = str[2] - 48;

LengthOfData = str[3] - 48;

for(j=0;j < LengthOfData;j++)

{

Data[k] = str[j+4];

K++;

}

} // end of data packet

}

Data[k] = '\0';

printf("Data : %s",Data);
```

```
fp=fopen(FileName,"w");

if(fp==NULL)

printf("unable to open file \n");

else

{

for(j=0;j<k;j++)

{

ch=Data[j];

fputc(ch,fp);

}

fclose(fp);

} // end of receive_packets

void port_close(int fd)

{

close (fd);

} // end of port_close

int main (void)

{

int fd;
```

```
char *a;  
  
fd = port_open ();  
  
port_config(fd);  
  
receive_packets(fd);  
  
port_close(fd);  
  
return 0;  
  
} // end of main
```