```
import java.net.*;
import java.util.Scanner;
import java.util.concurrent.*;
import java.io.*;
import javax.sound.sampled.*;
import java.util.ArrayList;
import java.awt.Desktop;
public class SocketProgg {
            static int MAX_SAMPLES_AUDIOBUFFEROUT=127872*2*2;
            static byte[] rxbuffer = new byte[2048];
            static byte[] audioBufferOut = new byte[MAX_SAMPLES_AUDIOBUFFEROUT];
            final static long NANOSEC_PER_SEC = 1000*1000*1000;
            public static void main(String[] args) throws IOException, LineUnavailableException {

               long startTime=System.nanoTime();
                        //Initialize DatagramSockets and clientPacket

                        SocketProgg sc = new SocketProgg();
                        DatagramSocket s = new DatagramSocket();  //socket for the server
                        byte[] hostIP = { (byte)155,(byte)207,(byte)18,(byte)208 };
                        int clientPort = Integer.parseInt(args[0]);
                        int serverPort = Integer.parseInt(args[1]);
                        DatagramSocket r = new DatagramSocket(clientPort); //socket for the client
                        r.setSoTimeout(8000);
                        DatagramPacket q=sc.clientPacket(r);

                        //EchoPackets

                        String packetInfo = args[2];
                        String packetInfo1 ="E0000";
                        boolean WithoutDelay=false; //flag if code is delay or without
                        DatagramPacket p=sc.ServerPacket(s,packetInfo,hostIP,serverPort); //initialize packet
with EchoPacketInfo
               sc.EchoPackets(s,r,p,q,WithoutDelay);
                        p=sc.ServerPacket(s, packetInfo1, hostIP, serverPort);  //initialize packet without
Delay
                        WithoutDelay=true;
                        sc.EchoPackets(s,r,p,q,WithoutDelay);

                        //Images
                        int NumberofPhotosInSuccession; //NumberOfPhotosWeWantInSuccess
                        boolean flagOfCam; //if true then we have FIX cam and if false we have PLZ cam
                        Scanner in = new Scanner(System.in);

                        //Image OF FIX Cam
                        String ImageInfoFix =args[3]+"UDP=128FLOW=ONCAM=FIX";
                        System.out.print("Give the number Of photos you want in succession :");
                        NumberofPhotosInSuccession = in.nextInt();
                        flagOfCam=true;
                        p=sc.ServerPacket(s,ImageInfoFix,hostIP,serverPort); //initialize packet with
ImageInfo
                        DatagramPacket n=sc.ServerPacket(s,"Next",hostIP,serverPort);
                        sc.Image(s, r, p, q,NumberofPhotosInSuccession,flagOfCam,n);

                        //Image Of PTZ Cam
                        String ImageInfoPLZ =args[3]+"FLOW=ONCAM=PTZ";
                        System.out.print("Give the number Of photos you want in succession :");
                        NumberofPhotosInSuccession = in.nextInt();
                        flagOfCam=false;
                        p=sc.ServerPacket(s,ImageInfoPLZ,hostIP,serverPort); //initialize packet with
ImageInfo
            sc.Image(s, r, p, q,NumberofPhotosInSuccession,flagOfCam,n);
                        in.close();

                                    //EchoPacketsWithTemp
                         String temp="T00";
                         packetInfo=args[2];
                        packetInfo= packetInfo+temp;
                        p=sc.ServerPacket(s,packetInfo,hostIP,serverPort); //initialize packet with
EchoPacketInfo
               sc.EchoPacketsTemp(s,r,p,q);
```

```
			//AudioClipRecievesOfRepertorio
			String AudioInfoRep= args[4]+"F200";
		p=sc.ServerPacket(s,AudioInfoRep,hostIP,serverPort); //initialize packet with 
AudioRepertorioInfo
		String NumberOfPacketsRep=AudioInfoRep.substring(6);  //numberOfPackets to be sent 
from the host server
		int NumberOfPacketsReps=Integer.parseInt(NumberOfPacketsRep);
		System.out.println(NumberOfPacketsReps);
		boolean flagOfRepertorio=true; //flagFor Repertorio Or gennhtria
		sc.AudioClipOnlyRecieverPacks(s,r,p,q,NumberOfPacketsReps,flagOfRepertorio);

			//AudioClipRecievesOfGennhtria
			String AudioInfoGen= args[4]+"T200";
		p=sc.ServerPacket(s,AudioInfoGen,hostIP,serverPort); //initialize packet with 
AudioGennhtriasInfo
		String NumberOfPacketsGen=AudioInfoGen.substring(6);  //numberOfPackets to be sent 
from the host server
		int NumberOfPacketsGens=Integer.parseInt(NumberOfPacketsGen);
		System.out.println(NumberOfPacketsGens);
		sc.AudioClipOnlyRecieverPacks(s,r,p,q,NumberOfPacketsGens,!flagOfRepertorio);



			//AudioClip DPCM


			String AudioInfoDPCM = args[4]+"F300";
		p=sc.ServerPacket(s,AudioInfoDPCM,hostIP,serverPort); //initialize packet with 
AudioDPCMInfo
		String NumberOfPacketsDPCM=AudioInfoDPCM.substring(6);  //numberOfPackets to be sent 
from the host server DPCM
		int NumberOfPacketsAudio=Integer.parseInt(NumberOfPacketsDPCM);
		System.out.println(NumberOfPacketsAudio);
		boolean t1 = true; //to flag DPCM(true) Or AQPCM(false)
		sc.AudioClip(s,r,p,q,NumberOfPacketsAudio,t1,true);

		//AudioClip AQDPCM 1st Time


		String AudioInfoAQDPCM = args[4]+"AQF300";
		String NumberOfPacketsAQ=AudioInfoAQDPCM.substring(8);  //numberOfPackets to be sent 
from the host server AQDPCM
		int NumberOfPacketsAudioAQ=Integer.parseInt(NumberOfPacketsAQ);
		System.out.println(NumberOfPacketsAQ);
		p=sc.ServerPacket(s,AudioInfoAQDPCM,hostIP,serverPort); //initialize packet with 
AudioAQDPCMInfo
		t1=false;
		boolean flagOf2ndTimeAQDPCMSent=false; //If it is the first time or the second
		sc.AudioClip(s,r,p,q,NumberOfPacketsAudioAQ,t1,flagOf2ndTimeAQDPCMSent);

		//AudioClip AQDPCM 2nd Time


		AudioInfoAQDPCM = args[4]+"AQF300";
		NumberOfPacketsAQ=AudioInfoAQDPCM.substring(8);  //numberOfPackets to be sent from 
the host server AQDPCM
		NumberOfPacketsAudioAQ=Integer.parseInt(NumberOfPacketsAQ);
		System.out.println(NumberOfPacketsAQ);
		p=sc.ServerPacket(s,AudioInfoAQDPCM,hostIP,serverPort); //initialize packet with 
AudioAQDPCMInfo
		t1=false;
		flagOf2ndTimeAQDPCMSent = true; //If it is the first time or the second
		sc.AudioClip(s,r,p,q,NumberOfPacketsAudioAQ,t1,flagOf2ndTimeAQDPCMSent);



		//IthakiCopter
		serverPort=38048;
		clientPort=48038;
		Socket st = new Socket (InetAddress.getByAddress(hostIP),serverPort); //Initialize TCP 
server-socket
		//IthakiCopter 1st Sending
		r = new DatagramSocket(clientPort); //only for UDP reciever
		q=sc.clientPacket(r); //only for UDP reciever
```

```
                String InfoToBeSent="AUTO FLIGHTLEVEL=200 LMOTOR=200 RMOTOR=200 PILOT \r\n";
//Message to be sent
                boolean TimeSent=false; //To know if it is the 1st time or the 2nd
                sc.IthakiCopter(st,r,q,InfoToBeSent,TimeSent);
                st.close();
                //IthakiCopter 2nd Sending
                st = new Socket (InetAddress.getByAddress(hostIP),serverPort);
                InfoToBeSent="AUTO FLIGHTLEVEL=400 LMOTOR=200 RMOTOR=200 PILOT \r\n";
//Message to be sent
                TimeSent=true; //To know if it is the 1st time or the 2nd
                sc.IthakiCopter(st,r,q,InfoToBeSent,TimeSent);
                st.close();
                //OBD-II Vehicle
                serverPort=29078;
                st=new Socket (InetAddress.getByAddress(hostIP),serverPort);
                sc.OBDII(st);
               // System.out.println("FINISH");
               // System.out.println("FINISH");
                s.close();
                         r.close();
                         st.close();
                         long endTime=System.nanoTime();
                         long duration=(endTime-startTime);
                         System.out.println("Duration in nanosecs:"+duration);
                         System.out.println("Duration in secs:"+duration/NANOSEC_PER_SEC);
                         }
//Initialize Server Packet UDP
public DatagramPacket ServerPacket (DatagramSocket s,String code,byte[] hostIP,int serverPort)
throws IOException {
             byte[] txbuffer = code.getBytes();
             InetAddress hostAddress = InetAddress.getByAddress(hostIP);
             DatagramPacket p = new DatagramPacket(txbuffer,txbuffer.length, hostAddress,serverPort);
             return p;



                                                                          }
//Initialize client Packet UDP
public DatagramPacket clientPacket (DatagramSocket r) {
             DatagramPacket q = new DatagramPacket(rxbuffer,rxbuffer.length);
             return q;


                            }
//EchoPackets
public  void EchoPackets(DatagramSocket s,DatagramSocket r,DatagramPacket p,DatagramPacket
q,boolean WithoutDelay) throws IOException {
            FileWriter writer1 = null;
            FileWriter writer2 =null;



            long startTime=System.nanoTime();
            long t;
            int dif;
            int SecondsForR=8;
            ArrayList<Integer> rarray = new ArrayList<Integer>();
            try {
                        if(!WithoutDelay) {
                         writer1= new FileWriter("C:\\Users\\Μάριος\\Desktop\\7ο Εξαμηνο\\Δικτυα
Υπολογιστων ΙΙ\\G1.txt");
                         writer2=new FileWriter("C:\\Users\\Μάριος\\Desktop\\7ο Εξαμηνο\\Δικτυα
Υπολογιστων ΙΙ\\G2.txt");
                                                                                    }
                        else if(WithoutDelay) {
                                    writer1= new FileWriter("C:\\Users\\Μάριος\\Desktop\\7ο Εξαμηνο\\Δικτυα
Υπολογιστων ΙΙ\\G3.txt");
                                    writer2=new FileWriter("C:\\Users\\Μάριος\\Desktop\\7ο Εξαμηνο\\Δικτυα
Υπολογιστων ΙΙ\\G4.txt");
                                                                                    }

                        while ((System.nanoTime()-startTime)<4*60*NANOSEC_PER_SEC) {
                                    s.send(p);
                                    t=System.currentTimeMillis();
                                    try {
```

```
                        r.receive(q);
                    String message = new String(rxbuffer,0,q.getLength());
                        System.out.println(message);
                        dif=(int)(System.currentTimeMillis()-t);
                        rarray.add(dif);
                        writer1.write(String.valueOf(dif)+"  ");
                            }
                        catch(IOException e){
                                    System.out.println(e);
                                                                    }


                                            }
                for(int i=0;i<rarray.size();i++) {
                            int NumberOfPackets=0;
                            int CountOfSecs=rarray.get(i);
                            for(int j=i;j<rarray.size();j++) {
                                    CountOfSecs+=rarray.get(j);
                                    NumberOfPackets++;
                                    double l=((double)CountOfSecs)/1000.0;
                                    //System.out.println(l);
                                            if(l>SecondsForR) {
                                            NumberOfPackets--;
                                            int R=NumberOfPackets*32*8/SecondsForR;
                                            writer2.write(String.valueOf(R)+"  ");
                                            break;

                                                                            }

            }
                                                                            }


                writer1.close();
                writer2.close();
                }
            catch(IOException e){
                System.out.println(e);
                                            }
        }
        //Image
        public  void Image(DatagramSocket s,DatagramSocket r,DatagramPacket p,DatagramPacket q,int
        NumberofPhotosInSuccession,boolean flag,DatagramPacket n) throws IOException {
            FileOutputStream out=null;
            boolean Byteflag = false ;
            Integer value;
            ArrayList<Integer> Bytes=new ArrayList<Integer>();

            int LastByte2=0;
            int LastByte1=0;
            int LocalCounterForPacket=0;
            Desktop dt = Desktop.getDesktop();
                for(int j=0;j<NumberofPhotosInSuccession;j++) {

                                    File file = null;
                                    if(flag==true) {
                                    file= new File("C:\\Users\\Μάριος\\Desktop\\7ο
        Εξαμηνο\\Δικτυα Υπολογιστων ΙΙ\\ImageFIX"+(j+1)+".jpeg");


                                    }
                                    else if(flag==false) {
                                            file= new File("C:\\Users\\Μάριος\\Desktop\\7ο
        Εξαμηνο\\Δικτυα Υπολογιστων ΙΙ\\ImagePTZ"+(j+1)+".jpeg");
                                                                        }
                                    out= new FileOutputStream(file);
                                    s.send(p);
                                    LastByte1=5;
                                    LastByte2=5;
                                    while(LastByte1 != 0xFF && LastByte2 != 0xD9) {

                                            try {
                                                    LocalCounterForPacket++;
                                                    r.receive(q);
                                                    LocalCounterForPacket=0;
```

```
                                                                        s.send(n);
                                                                        rxbuffer=q.getData();
                                                                        for(int i=0;i<q.getLength();i++) {

value=Byte.toUnsignedInt(rxbuffer[i]);
                                                                                      Bytes.add(value);

        if(Bytes.size()>=2) {

        if(Bytes.get(0)==0xFF && Bytes.get(1)==0xD8 && Byteflag==false) {

                System.out.println("Good Start");

                out.write(Bytes.get(0).byteValue());

                out.write(Bytes.get(1).byteValue());

                Byteflag = true;



                                                        }


        else if(Byteflag) {

                        out.write(rxbuffer[i]);

                        if(Bytes.get(Bytes.size()-2)==0xFF && Bytes.get(Bytes.size()-1)==0xD9) {

                                  System.out.println("Good Breaking");

                                        LastByte1=0xFF;

                                        LastByte2=0xD9;

                                  dt.open(file);

                                        Byteflag=false;

                                        Bytes.clear();




                                                                                      }

                      }

                                        }


                                                        }







                                                                                              }

                                                        catch(SocketTimeoutException e){
                                                                if(LocalCounterForPacket>=3) {
```

```
                                                                                j=j-1;
                                                                                break;


                                        }
                                                                        else continue;


                                        }
                                                                }
                                }
                out.close();
}

 //EchoPackesWithTemp
public  void EchoPacketsTemp(DatagramSocket s,DatagramSocket r,DatagramPacket
p,DatagramPacket q) throws IOException {
                FileWriter writer;
                File file = new File("C:\\Users\\Μάριος\\Desktop\\7o Εξαμηνο\\Δικτυα Υπολογιστων
II\\Temp.txt");
                writer = new FileWriter(file);
                for(int i=0;i<8;i++) {
                                s.send(p);
                                r.receive(q);
                                String message = new String(rxbuffer,0,q.getLength());
                                System.out.println(message);
                                String [] str=message.split(" ");
                                System.out.println(str[6]);
                                writer.write(str[6]+" ");



                                                                }
                 writer.close();
}


//AudioClipForOnlyRecievePackets
public void AudioClipOnlyRecieverPacks(DatagramSocket s,DatagramSocket r,DatagramPacket
p,DatagramPacket q,int NumberOfPacketsAudio,boolean flagOfRepertorio) {
                FileWriter writer1;
                File file=null;
                int NumberOfPacketsRecieved=0;
                try {
                                s.send(p);
                                if(flagOfRepertorio) {
                                file = new File("C:\\Users\\Μάριος\\Desktop\\7o Εξαμηνο\\Δικτυα Υπολογιστων
II\\G10.txt");
                                                                        }
                                if(!flagOfRepertorio) {
                                file = new File("C:\\Users\\Μάριος\\Desktop\\7o Εξαμηνο\\Δικτυα Υπολογιστων
II\\G9.txt");
                                                                        }
                                writer1= new FileWriter(file);
                                for(;;) {
                                        try {
                                                r.receive(q);
                                          // System.out.println("Good Start");
                                          //System.out.println(q.getLength());
                                                NumberOfPacketsRecieved++;
                                          rxbuffer=q.getData();
                                                for(int i=0;i<q.getLength();i++)
                                                {
                                                        //System.out.println(rxbuffer[i]);


                writer1.write(String.valueOf(Byte.toUnsignedInt(rxbuffer[i]))+"  ");
                                                }
                                                if(NumberOfPacketsRecieved==NumberOfPacketsAudio) {
                                                        //System.out.println("Good Break");
                                                        break;

                                                         }
                                }
                                catch(SocketTimeoutException e){
                                                        break;
                                                                    }
                        }
                }
                catch (IOException e) {
                                e.printStackTrace();
                        }
                }
//AudiosClip
public void AudioClip(DatagramSocket s,DatagramSocket r,DatagramPacket p,DatagramPacket q,int
NumberOfPacketsAudio,boolean t1,boolean flagOf2ndTimeAQDPCMSent) throws
LineUnavailableException, IOException  {
                final int BytesOfPacketDPCM = 128;
                final int BytesOfPacketAQDPCM = 132;
                int NumberOfSumBytes=0;
                int InfoOfSamplesInAByte=0;
                ArrayList<Byte> SumBytes= new ArrayList<Byte>();
                int NumberOfPacketsReceived=0;
                int Q = 0;
                int DPCM=0;
                int AQDPCM=1;
                final int SamplesCodedInByte=2;  //samples that coded in a byte
                FileOutputStream out = null;
                FileWriter writer1;
                FileWriter writer2;
                FileWriter writer3=null;
                FileWriter writer4=null;
                File file1=null;
                File file2=null;
                File file3=null;
                File file4=null;
                if(t1) {
                        file1= new File("C:\\Users\\Μάριος\\Desktop\\7ο Εξαμηνο\\Δικτυα Υπολογιστων
II\\G11.txt");
                        file2=new File("C:\\Users\\Μάριος\\Desktop\\7ο Εξαμηνο\\Δικτυα Υπολογιστων
II\\G12.txt");
                          }
                else if(!t1) {
                        file1=new File("C:\\Users\\Μάριος\\Desktop\\7ο Εξαμηνο\\Δικτυα Υπολογιστων
II\\G13.txt");
                        file2=new File("C:\\Users\\Μάριος\\Desktop\\7ο Εξαμηνο\\Δικτυα Υπολογιστων
II\\G14.txt");

                        if(!flagOf2ndTimeAQDPCMSent) {
                        file3=new File("C:\\Users\\Μάριος\\Desktop\\7ο Εξαμηνο\\Δικτυα Υπολογιστων
II\\G15.txt");
                        file4=new File("C:\\Users\\Μάριος\\Desktop\\7ο Εξαμηνο\\Δικτυα Υπολογιστων
II\\G16.txt");
                                                                                }
                        else if(flagOf2ndTimeAQDPCMSent) {
                                file3=new File("C:\\Users\\Μάριος\\Desktop\\7ο Εξαμηνο\\Δικτυα
Υπολογιστων II\\G17.txt");
                                file4=new File("C:\\Users\\Μάριος\\Desktop\\7ο Εξαμηνο\\Δικτυα
Υπολογιστων II\\G18.txt");
                                                                                  }
                        writer3=new FileWriter(file3);
                        writer4=new FileWriter(file4);

                                }
                writer1=new FileWriter(file1);
                writer2=new FileWriter(file2);

                try {
                         s.send(p);

                        for(;;) {
                                try {
                                    r.receive(q);
```

```
                                                    NumberOfPacketsReceived++;
                                              //System.out.println("GoodStart");
                                             // System.out.println(NumberOfPacketsReceived);
                                              rxbuffer=q.getData();
                                                    for(int i=0;i<q.getLength();i++)
                                                    {
                                                            //System.out.println(rxbuffer[i]);
                                                            SumBytes.add(rxbuffer[i]);


                                                    }
                                                    if(NumberOfPacketsReceived==NumberOfPacketsAudio) {
                                                            //System.out.println("Good Break");
                                                            break;
                                                    }




                                                                                            }
                                    catch(SocketTimeoutException e){
                                                            break;
                                                                                            }
                    }
            }
            catch (IOException e) {
                                    e.printStackTrace();
                    }
            if(t1) {
                                    //System.out.println("Good Intro In DPCM");
                              out = new FileOutputStream ("C:\\Users\\Μάριος\\Desktop\\7ο Εξαμηνο\\Δικτυα
Υπολογιστων ΙΙ\\audio"+DPCM+".WAV");
                              InfoOfSamplesInAByte=2;
                              NumberOfSumBytes=NumberOfPacketsReceived*BytesOfPacketDPCM;
                                    Q=8;
                                    int count=0;
                                    //demulation of DPCM
                                    Integer Sample1;
                                    Integer Sample2=0;
                                    for(int i=0;i<NumberOfSumBytes;i++) {
                                            int a = SumBytes.get(i);
             int Nibble1 = ((0xF0 & a)>>4);//The first nibble
             int Nibble2 = (0xF & a);//The second nibble
             int beta = 3;
             int difference1 = (Nibble1-8)*beta;
             int difference2 = (Nibble2-8)*beta;
             writer1.write(String.valueOf(difference1)+"  "+String.valueOf(String.valueOf(difference2))+"
");
             Sample1 = Sample2+difference1;
             Sample2 = Sample1 + difference2;
             writer2.write(String.valueOf(Sample1)+"  "+String.valueOf(String.valueOf(Sample2))+" ");
             audioBufferOut[count]=Sample1.byteValue();
             count++;
             audioBufferOut[count]=Sample2.byteValue();
             count++;


            }
                                    }
            else if(!t1) {
                                    //System.out.println("Good Intro In AQDPCM");
                                    if(!flagOf2ndTimeAQDPCMSent) {
                                             out = new FileOutputStream ("C:\\Users\\Μάριος\\Desktop\\7ο
Εξαμηνο\\Δικτυα Υπολογιστων ΙΙ\\audio"+AQDPCM+".WAV");
                                    }
                                    else if(flagOf2ndTimeAQDPCMSent) {
                                            out = new FileOutputStream ("C:\\Users\\Μάριος\\Desktop\\7ο
Εξαμηνο\\Δικτυα Υπολογιστων ΙΙ\\audio"+AQDPCM+1+".WAV");
                                    }

                              InfoOfSamplesInAByte=4;
                                    Q=16;
                                    int count=0;
                                    NumberOfSumBytes=NumberOfPacketsReceived*BytesOfPacketAQDPCM;
                                    //demulation of AQDPCM
```

```
                                        Integer Sample1;
                                        Integer Sample2=0;
                                        Integer mean;
                                        Integer step;
                                        int counterByteHeader=0;
                                        int counterBytePacket=0;
                                        int meanMSB;
                                        int meanLSB;
                                        int stepLSB;
                                        int stepMSB;

                                        for(int i=0;i<NumberOfPacketsReceived;i++) {
                                                counterByteHeader=BytesOfPacketAQDPCM*i;
                                                counterBytePacket=4+BytesOfPacketAQDPCM*i;
                                                int a=(int)SumBytes.get(counterByteHeader);
                                          meanLSB=a;
                                                a=SumBytes.get(counterByteHeader+1);
                                                meanMSB =a*256;
                                                mean= (meanMSB + meanLSB);
                                                writer3.write(String.valueOf(mean)+" ");
                                                a=(int)SumBytes.get(counterByteHeader+2);
                                                stepLSB=a;
                                                a=(int)SumBytes.get(counterByteHeader+3);
                                                stepMSB =a*256;
                                                step= (stepMSB + stepLSB);
                                                writer4.write(String.valueOf(step)+" ");
                                                for(int
j=counterBytePacket;j<(BytesOfPacketDPCM+counterBytePacket);j++) {
                                                        a=SumBytes.get(j);
                                                        int Nibble1 = ((0xF0 & a)>>4);//The first nibble
                                 int Nibble2 = (0xF & a);//The second nibble
                            int Difference1 = Nibble1-8;
                            int Difference2 = Nibble2-8;
                            writer1.write(String.valueOf(Difference1)+" "+String.valueOf(Difference2)+" ");
                            //Creation of samples
                            Sample1 = (Difference1*step+mean); //First demodulated sample (16 bits)
                            Sample2 = (Difference2*step+mean); //Second demodulated sample (16 bits)
                            writer2.write(String.valueOf(Sample1 & 0xFF)+" "+String.valueOf(Sample1 >>8)+"
"+String.valueOf(Sample2 & 0xFF)+" "+String.valueOf(Sample2 >>8)+" ");
                            audioBufferOut[count]=(byte)(Sample1 & 0xFF);
                            count++;
                            audioBufferOut[count]=(byte)(Sample1 >>8);
                            count++;
                            audioBufferOut[count]=(byte)(Sample2 & 0xFF);
                            count++;
                            audioBufferOut[count]=(byte)(Sample2 >> 8);
                            count++;
                                                }

                                        }



                                        writer3.close();
                                        writer4.close();



                                        }
                SumBytes.clear();
                writer1.close();
                writer2.close();

                /* for(int i=0;i<(NumberOfPacketsAudio*BytesOfPacketDPCM);i++) {
                        System.out.println(audioBufferOut[i]);
                } */
                                        //Play Of the audio
                                        AudioFormat linearPCM = new AudioFormat(8000,Q,1,true,false);
                                        SourceDataLine lineOut = AudioSystem.getSourceDataLine(linearPCM);



lineOut.open(linearPCM,NumberOfPacketsReceived*BytesOfPacketDPCM*InfoOfSamplesInAByte);
                                        lineOut.start();
lineOut.write(audioBufferOut,0,NumberOfPacketsReceived*BytesOfPacketDPCM*InfoOfSamplesInAByte
);
                                        lineOut.stop();
                                        lineOut.close();

                              //Save of the audio

                                        ByteArrayInputStream bais = new ByteArrayInputStream(audioBufferOut);
                                 AudioInputStream audioInputStream;
                                        audioInputStream = new
AudioInputStream(bais,linearPCM,NumberOfPacketsReceived*BytesOfPacketDPCM*SamplesCodedInByt
e);
                                        AudioSystem.write(audioInputStream, AudioFileFormat.Type.WAVE,out);
                                        audioInputStream.close();
                                        bais.close();
                                        out.close();

                                        //System.out.println("GOOD ENDING JOB DONE");
                              }
//IthakiCopter
public void IthakiCopter(Socket st,DatagramSocket r,DatagramPacket q,String Info,boolean
TimeSent) throws IOException   {
              final long NANOSEC_PER_SEC = 1000*1000*1000;
              long startTime=System.nanoTime();
              boolean FirstTCPflag= true;
              FileWriter writer;
              File file=null;
    InputStream in = st.getInputStream();
    OutputStream out = st.getOutputStream();
                              if(!TimeSent) {
                                        file= new File("C:\\Users\\Μάριος\\Desktop\\7ο Εξαμηνο\\Δικτυα
Υπολογιστων ΙΙ\\G19.txt");


                              }
                              else if(TimeSent) {
                                        file= new File("C:\\Users\\Μάριος\\Desktop\\7ο Εξαμηνο\\Δικτυα
Υπολογιστων ΙΙ\\G20.txt");
                              }
                              writer= new FileWriter(file);
              while ((System.nanoTime()-startTime)<2*60*NANOSEC_PER_SEC) {

              try {
                                        //System.out.println("START");

                                        out.write(Info.getBytes());
                                        String message="";


                                        TimeUnit.SECONDS.sleep(1);


                                        //TCP recieve
                                             in.read(rxbuffer);
                                                            for(int i=0;i<rxbuffer.length;i++) {
                                                                                message+=(char)rxbuffer[i];
                                                                                if(rxbuffer[i]==0) {
                                                                                          break;
                                                                                }
                                                                                if(message.contains("ITHAKICOPTER
LMOTOR=LLL RMOTOR=RRR ALTITUDE=AAA TEMPERATURE=TT.TT PRESSURE=PPPP.PP TELEMETRY
<CR><LF><br>\r\n"+"<br>"+"\r\n") && FirstTCPflag ) {
                                                                                message="";
                                                                                FirstTCPflag=false;


                                                                                }

                                                                                if(message.endsWith("\r\n") && !
FirstTCPflag) {
                                                                                          break;
```

```
                                                                }


                                        }





                                /*
                                        //UDP recieve
                                         r.receive(q);
                                         rxbuffer=q.getData();
                                                for(int i=0;i<q.getLength();i++) {
                                                        message+=(char)rxbuffer[i];


                                                }
                                */
                                                        System.out.println(message);
                                                        String [] str=message.split(" ");
                                                        String [] realParts=str[3].split("=");
                                                        writer.write(realParts[1]+" ");


                                 // System.out.println("END");


                        } catch (IOException | InterruptedException e) {


                                e.printStackTrace();
                        }

                        }
    out.close();
    in.close();
                        writer.close();
}
//Vehicle OBD-II
public void OBDII(Socket st) throws IOException {
                InputStream in=st.getInputStream();
                OutputStream out=st.getOutputStream();
                int XX;
                int YY;
                String XXHex="";
                String YYHex="";
                final long NANOSEC_PER_SEC = 1000*1000*1000;


                FileWriter Fout1 = new FileWriter("C:\\Users\\Μάριος\\Desktop\\7ο Εξαμηνο\\Δικτυα
Υπολογιστων II\\OBD-"+1+" parameter.txt");
                FileWriter Fout2 = new FileWriter("C:\\Users\\Μάριος\\Desktop\\7ο Εξαμηνο\\Δικτυα
Υπολογιστων II\\OBD-"+2+" parameter.txt");
                FileWriter Fout3 = new FileWriter("C:\\Users\\Μάριος\\Desktop\\7ο Εξαμηνο\\Δικτυα
Υπολογιστων II\\OBD-"+3+" parameter.txt");
                FileWriter Fout4 = new FileWriter("C:\\Users\\Μάριος\\Desktop\\7ο Εξαμηνο\\Δικτυα
Υπολογιστων II\\OBD-"+4+" parameter.txt");
                FileWriter Fout5 = new FileWriter("C:\\Users\\Μάριος\\Desktop\\7ο Εξαμηνο\\Δικτυα
Υπολογιστων II\\OBD-"+5+" parameter.txt");
                FileWriter Fout6 = new FileWriter("C:\\Users\\Μάριος\\Desktop\\7ο Εξαμηνο\\Δικτυα
Υπολογιστων II\\OBD-"+6+" parameter.txt");
                long startTime=System.nanoTime();
                while ((System.nanoTime()-startTime)<4*60*NANOSEC_PER_SEC) {
                                        //Engine run time
                        out.write("01 1F\r".getBytes());
                        in.read(rxbuffer);
                        XXHex=(char)rxbuffer[6]+""+(char)rxbuffer[7]; //XX in HEX
                        YYHex=(char)rxbuffer[9]+""+(char)rxbuffer[10];  //YY in HEX
                        XX=Integer.parseInt(XXHex,16);  //XX in Decimal
                        YY=Integer.parseInt(YYHex,16);       //YY in Decimal
                        int EngineRunTime =256*XX+YY;
                        //System.out.println(EngineRunTime);
                        Fout1.write(String.valueOf(EngineRunTime)+"  ");
                        for(int i=0;i<rxbuffer.length;i++) {
                                        if(rxbuffer[i]==0) break;
```

```
                    //System.out.print(" "+rxbuffer[i]);
                    rxbuffer[i]=0;
                                                                                    }


          //System.out.println("\r");

               //Intake air temperature


          out.write("01 0F\r".getBytes());
          in.read(rxbuffer);
          XXHex=(char)rxbuffer[6]+""+(char)rxbuffer[7]; //XX in HEX
          XX=Integer.parseInt(XXHex,16);  //XX in Decimal
          int IntakeAirT=XX-40;
          //System.out.println(IntakeAirT);
          Fout2.write(String.valueOf(IntakeAirT)+"  ");
          for(int i=0;i<rxbuffer.length;i++) {
               if(rxbuffer[i]==0) break;
               //System.out.print(" "+rxbuffer[i]);
               rxbuffer[i]=0;          //clear rxbuffer
                                                                                    }
          //System.out.println("\r");


               //Throttle position
          out.write("01 11\r".getBytes());
          in.read(rxbuffer);
          XXHex=(char)rxbuffer[6]+""+(char)rxbuffer[7]; //XX in HEX
          XX=Integer.parseInt(XXHex,16);  //XX in Decimal
          int ThrottlePos = (XX*100)/255;
          //System.out.println(ThrottlePos);
          Fout3.write(String.valueOf(ThrottlePos)+"  ");
          for(int i=0;i<rxbuffer.length;i++) {
               if(rxbuffer[i]==0) break;
               //System.out.print(" "+rxbuffer[i]);
               rxbuffer[i]=0;  //clear rxbuffer
                                                                                    }
          //System.out.println("\r");



                    //Engine RPM
          out.write("01 0C\r".getBytes());
          in.read(rxbuffer);
          //System.out.println((char)rxbuffer[0]+""+(char)rxbuffer[1]+" "+
(char)rxbuffer[3]+""+(char)rxbuffer[4]);
          XXHex=(char)rxbuffer[6]+""+(char)rxbuffer[7]; //XX in HEX
          YYHex=(char)rxbuffer[9]+""+(char)rxbuffer[10];  //YY in HEX
          XX=Integer.parseInt(XXHex,16);  //XX in Decimal
          YY=Integer.parseInt(YYHex,16);       //YY in Decimal
          int EngineRPM = ((XX*256)+YY)/4;
          //System.out.println(EngineRPM);
          Fout4.write(String.valueOf(EngineRPM)+"  ");
          for(int i=0;i<rxbuffer.length;i++) {
               if(rxbuffer[i]==0) break;
            //System.out.print(" "+rxbuffer[i]);
               rxbuffer[i]=0;  //clear rxbuffer
                                                                                    }
          //System.out.println("\r");


               //Vehicle speed
          out.write("01 0D\r".getBytes());
          in.read(rxbuffer);
          //System.out.println((char)rxbuffer[0]+""+(char)rxbuffer[1]+" "+
(char)rxbuffer[3]+""+(char)rxbuffer[4]);
          XXHex=(char)rxbuffer[6]+""+(char)rxbuffer[7]; //XX in HEX
          XX=Integer.parseInt(XXHex,16);  //XX in Decimal
          int VehicleSpeed  = XX;                  // XX in Decimal
          //System.out.println(VehicleSpeed);
          Fout5.write(String.valueOf(VehicleSpeed)+"  ");
               for(int i=0;i<rxbuffer.length;i++) {
                    if(rxbuffer[i]==0) break;
                    //System.out.print(" "+rxbuffer[i]);
                    rxbuffer[i]=0;  //clear rxbuffer
```

```
        }
                        //System.out.println("\r");

                        //Coolant temperature
                out.write("01 05\r".getBytes());
                in.read(rxbuffer);
                XXHex=(char)rxbuffer[6]+""+(char)rxbuffer[7]; //XX in HEX
                XX=Integer.parseInt(XXHex,16);  //XX in Decimal
                int  CoolantT  = XX-40;
                //System.out.println(CoolantT);
                Fout6.write(String.valueOf(CoolantT)+"  ");

                        for(int i=0;i<rxbuffer.length;i++) {
                                if(rxbuffer[i]==0) break;
                                //System.out.print(" "+rxbuffer[i]);
                                rxbuffer[i]=0;  //clear rxbuffer

        }
                //System.out.println("\r");
                }
        out.close();
        in.close();
        Fout1.close();
        Fout2.close();
        Fout3.close();
        Fout4.close();
        Fout5.close();
        Fout6.close();
}
}
```