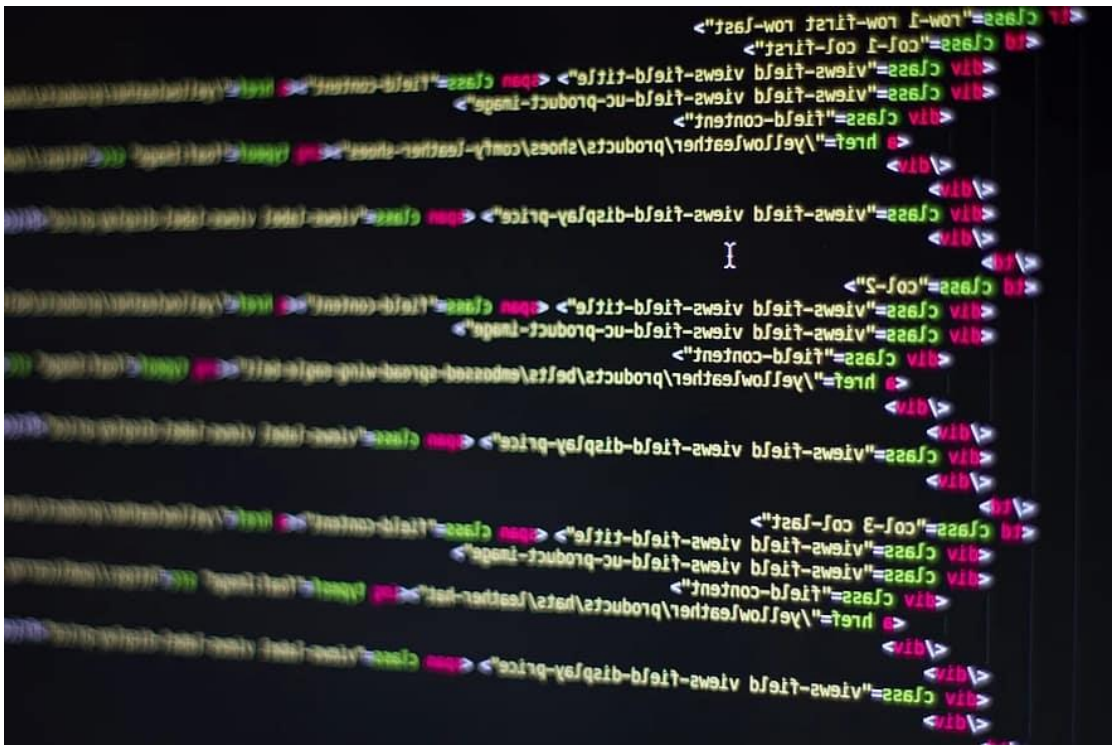
Δίκτυα Υπολογιστών ΙΙ

Ονοματεπώνυμο : Κελέση Ελπίδα

ΑΕΜ : 09410

Email : elpidakelesi@ece.auth.gr



Καθηγητής : Δημήτριος Μητράκος

Ακαδημαϊκό έτος : 2020-2021

Εξάμηνο : Χειμερινό

```
//vivliothikes diaxeirisis diadiktiakwn porwn
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.net.Socket;

//vivliothikes diaxeirisis ipologistikwn porwn
import java.io.*;

//vivliothiki gia eggrafi kai anaparagwgi ixou
import javax.sound.sampled.*;

/*vivliothikes oi opoies dn anasteloun tin ektelesi tis efarmogis oso perimenoun paketo apo tin
methodo receive() */
import java.nio.ByteBuffer;
import java.nio.ByteOrder;
//vivliothikes gia to xrono
import java.time.*;
import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;

import java.util.ArrayList;
import java.util.Iterator;
import java.util.Scanner;
import static java.lang.System.exit;


public class Net2 {


        //metavlites gia apothikeusi arxeiwn kai emfanisi imerominias
        private static LocalDate date;
   private static FileOutputStream image;
```

```
private static FileOutputStream echo;
private static FileOutputStream time;

        //ports apo ithaki
private static int server_port = 38007;
private static int client_port = 48007;

//monopati gia apothikeusi twn apotelesmatwn
private  static String path="C:\\Users\\User\\Documents\\Eclipse\\diktia1\\src\\diktia1\\Session";

private static final String[] vehicleoption = new String[] { "01 1F", "01 0F",
                                                "01 11", "01 0C",
                                                 "01 0D", "01 05" };

public Net2() {
}

public static void main(String[] args) throws IOException, LineUnavailableException {

   //kwdikoi apo ithaki
   String echo = "E7440";
   String image = "M6638";
   String audio = "A0972";
   String copter = "Q0700";
   String vehicle = "V1531";
   int mode=0;
   int mode2=0;

                System.out.println(" ");
                System.out.println("The menu for this application is : " );
                System.out.println(" ");

   Scanner input = new Scanner( System.in );
```

```
//------------------------------The Menu of the Application-------------------------------------------

 //dialegoume poio paketo theloume na paroume apo to menu
 System.out.println("Choose mode :");
 System.out.println("Press 1 in order to take the Echo packages");
 System.out.println("Press 2 in order to take the Image packages");
 System.out.println("Press 3 in order to take the Audio packages");
 System.out.println("Press 4 in order to take the Copter packages");
 System.out.println("Press 5 in order to take the Vehicle packages");
 System.out.println("Press 6 in order to exit the program.");

 // diavazei ton arithmo pou epileksame apo to menu
 int selection1 = Integer.parseInt(input.nextLine());



 //analoga me to selection energopoieitai i katallili sinartisi

 //Echo packages
    if(selection1 == 1) {
       System.out.println( "Press 1 in order to take the echo packages without Delay and with
Temperature." );
       System.out.println("Press 2 in order to take the echo packages with Delay.");
       mode = Integer.parseInt(input.nextLine());
       if (mode == 1){
          Echo(1,"E0000");
       }
       else if (mode == 2) {
          Echo(2,echo);
       }
       else{
          System.err.println("You didn't press the right button !");
          System.out.println("The program has stopped.");
```

```
        exit(-1);
    }
}
//Image packages
else if(selection1==2) {
    System.out.println("Press 1 in order to take mesurements from FIX camera.");
    System.out.println("Press 2 in order to take mesurements from PTZ camera.");
    mode = Integer.parseInt(input.nextLine());
    if (mode == 1){
        ImageRequest(1,image);
    }else if (mode == 2) {
        ImageRequest(2,image);
    }else{
                System.err.println("You didn't press the right button !");
        System.out.println("The program has stopped.");
        exit(-1);
    }
}
//Audio packages
else if(selection1== 3) {
    System.out.println("Press 1 for non adaptive coding (DPCM).");
    System.out.println("Press 2 for adaptive coding (AQ-DCPM)");
    mode = Integer.parseInt(input.nextLine());
    if(mode ==1){
                System.out.println("Press 1 for music.");
        System.out.println("Press 2 for frequency.");
        mode2 = Integer.parseInt(input.nextLine());
        if (mode2 == 1)
            Dpcm(8, "F", null,audio);
        else if (mode2 == 2)
            Dpcm(8, "T", null,audio);
        else {
                System.err.println("You didn't press the right button !");
```

```
            System.out.println("The program has stopped.");
            exit(-1);
        }
    }else if (mode == 2) {
        Aq(16, "F", "AQ",audio);
    }else{
                System.err.println("You didn't press the right button !");
        System.out.println("The program has stopped.");
        exit(-1);
    }
}
//Helicopter packages
else if(selection1== 4) {
    System.out.println( "Press the flightlevel you want." );
    mode = Integer.parseInt(input.nextLine());
    IthakiCopter(mode);
}
//Vehicle packages
else if(selection1 ==5) {
    System.out.println( "Press 0 for Engine run time." );
    System.out.println( "Press 1 for Intake air temperature." );
    System.out.println( "Press 2 for Throttle Position." );
    System.out.println( "Press 3 for Engine rpm." );
    System.out.println( "Press 4 for Vehicle speed." );
    System.out.println( "Press 5 for Coolant temperature." );
    System.out.println( " " );
    mode = Integer.parseInt(input.nextLine());
    if (mode == 0){
        Vehicle(mode,vehicle);
    }else if (mode == 1) {
        Vehicle(mode,vehicle);
    }else if (mode ==2){
        Vehicle(mode,vehicle);
```

```
            }else if (mode == 3) {
               Vehicle(mode,vehicle);
            }else if (mode==4){
               Vehicle(mode,vehicle);
            }else if (mode==5) {
               Vehicle(mode,vehicle);
            }else{
               System.err.println("Please, select a valid vehicle Operation.");
               exit(-1);
            }
         }

         else if (selection1==6) {
            System.out.println("The program has stopped.");
            exit(1);
         }
         else {
            System.err.println("You press the wrong button.");
            exit(-1);
         }
   }

//-------------------------------------------------Echo Packages-----------------------------------------------------

   public static void Echo(int selection,String code) throws IOException {

      //dimiourgia kai lipsi socket
      DatagramSocket socket = new DatagramSocket();
      DatagramSocket receive = new DatagramSocket(client_port);
      date = LocalDate.now();
      byte[] txbuffer;
      byte[]hostIP = { (byte)155,(byte)207,(byte)18,(byte)208};
      InetAddress hostAddress = InetAddress.getByAddress(hostIP);
```

```
//paketa echo xoris to delay
if (selection == 1) {
      code=code+"T00\r";
      File Echo = new File(path + "/Echo.txt");
      File Echo_Time = new File(path + "/Echo_Time.csv");
   echo = new FileOutputStream(Echo);
   time = new FileOutputStream(Echo_Time);
   System.out.print(" The echo packages with code " );
   System.out.println(code );
   System.out.println( " started at " + date );
   System.out.println(" ");
}
//paketa echo me to delay
else if (selection == 2) {
   File Echo_Delay = new File(path + "/Echo_Delay.txt");
   File Echo_Time_Delay = new File(path + "/Echo_Time_Delay.csv");
   echo = new FileOutputStream(Echo_Delay);
   time = new FileOutputStream(Echo_Time_Delay);
   System.out.print("The echo packages with code " );
   System.out.println(code);
   System.out.println(" started at " + date );
   System.out.println(" ");
}
txbuffer = code.getBytes();
receive.setSoTimeout(8000);
byte[] rxbuffer = new byte[2048];
DatagramPacket p = new DatagramPacket(txbuffer, txbuffer.length, hostAddress, server_port);
int packages = 0;
long echo_time = 300000; // 5 lepta
long start = System.currentTimeMillis();
long time_that_arrives, time_that_is_received;
String packets = "" ,time_per_packet = "";
```

```
    while (System.currentTimeMillis() - start < echo_time) {
        String message;
        time_that_arrives = System.currentTimeMillis();
        socket.send(p);
        DatagramPacket q = new DatagramPacket(rxbuffer, rxbuffer.length);
        try {
            receive.receive(q);
            message = new String(rxbuffer, 0, q.getLength());
            System.out.println(message);
            packages += 1;
            time_that_is_received = System.currentTimeMillis();
            time_per_packet += time_that_is_received - time_that_arrives + ",";
            packets += message + "\r\n";
        } catch (Exception x) {
            System.err.println("Ops something went wrong, package didn't arrive.");
        }
    }
    // grafei ta string tou echo kai tou xronou ( anamesa se kathe paketo) sta arxeia
    try {
            System.out.println("The total packages are :"+ packages);
        echo.write(packets.getBytes());
        time.write(time_per_packet.getBytes());
        echo.close();
        time.close();
        System.out.println("The echo packages are written in the files.");
    } catch (IOException x) {
    }
    receive.close();
}

//--------------------------------Image Package-----------------------------------------------------------------------

public static void ImageRequest(int selection, String code) throws IOException {
```

```
        DatagramSocket socket = new DatagramSocket();
        DatagramSocket receive = new DatagramSocket(client_port);
        date = LocalDate.now();
byte[] txbuffer;
byte[]hostIP = { (byte)155,(byte)207,(byte)18,(byte)208};
InetAddress hostAddress = InetAddress.getByAddress(hostIP);

if (selection == 1) {
   File image_FIX = new File(path + "/FIX_image.jpeg");
   image = new FileOutputStream(image_FIX);
   code = code + "CAM=FIX" ;
   System.out.println("The image with code " + code + " started at " + date );
   System.out.println(" ");
}
else if (selection == 2) {
   File image_PTZ = new File(path + "/PTZ_image.jpeg");
   image = new FileOutputStream(image_PTZ);
   code = code + "CAM=PTZ";
   System.out.println("The image with code " + code + " started at " + date );
   System.out.println(" ");
}

int length1;
int length2 = 0;
txbuffer = code.getBytes();
DatagramPacket p = new DatagramPacket(txbuffer, txbuffer.length, hostAddress, server_port);
receive.setSoTimeout(8000);
byte[] rxbuffer = new byte[2048];
ArrayList<Byte> message = new ArrayList<>();
socket.send(p);
do {
   length1 = length2;
```

```
        DatagramPacket q = new DatagramPacket(rxbuffer, rxbuffer.length);
        try {
           receive.receive(q);
           length2 = q.getLength();
           byte[] packet = q.getData();
           for (int i = 0; i < length2; i++) {
              message.add(packet[i]);
           }
        } catch (Exception x) {
           System.err.println("Ops something went wrong, image didn't arrive.");
           exit(-1);
        }
     } while (length2 >= length1);

     try {
        image.write(convertBytes(message));
        image.close();
        System.out.println("The image is written in the file.");
     } catch (IOException x) {
     }
     receive.close();
  }

//---------------------------------------Audio Package-------------------------------------------------------------

   public static void Dpcm(int bits, String option, String encode, String audio) throws IOException,
LineUnavailableException {

           DatagramSocket socket = new DatagramSocket();
           DatagramSocket receive = new DatagramSocket(client_port);
           date = LocalDate.now();
           byte[] txbuffer;
           byte[]hostIP = { (byte)155,(byte)207,(byte)18,(byte)208};
     InetAddress hostAddress = InetAddress.getByAddress(hostIP);
```

```
int number_of_packets =999;
String check;
check = option ;
String code = audio + option  + number_of_packets ;
File gen;
FileOutputStream samples_stream;
FileOutputStream differences_stream;

if (check == "T") {
   System.out.println("The frequency Dcpm starts arriving at " + date);
   gen = new File(path + "/dcpmt.wav");
   File deigmata = new File(path + "/dcpmt_samples.csv");
   File differences1 = new File(path + "/dcpmt_differences.csv");
   samples_stream = new FileOutputStream(deigmata);
   differences_stream = new FileOutputStream(differences1);
   }
else {
  System.out.println("The audio Dcpm starts arriving at " + date);
  gen = new File(path + "/dcpmf.wav");
  File deigmata = new File(path + "/dcpmf_samples.csv");
  File differences1 = new File(path + "/dcpmf_differences.csv");
  samples_stream = new FileOutputStream(deigmata);
  differences_stream = new FileOutputStream(differences1);
   }

   txbuffer = code.getBytes();
   DatagramPacket p = new DatagramPacket(txbuffer, txbuffer.length, hostAddress, server_port);
   receive.setSoTimeout(8000);
   byte[] rxbuffer = new byte[2048];
   socket.send(p);

   //metavlites
```

```
        byte[] dedomena;
        String samples = "", differences = "";
        int packages = 0;
        int length, niddle, niddle1, Sample1, Sample2, sum;
        double average;
        ArrayList<Integer> Samples1 = new ArrayList<>();
        ArrayList<Byte> Samples = new ArrayList<>();

        do {
           DatagramPacket q = new DatagramPacket(rxbuffer, rxbuffer.length);
           try {
              sum = 0;
              Sample2 = 0;
              Samples1.add(Sample2);
              receive.receive(q);
              length = q.getLength();
              dedomena = q.getData();
              // Get the niddles from the byte //
              for (int j = 0; j < length; j++) {
                 niddle = ((0b11110000 & dedomena[j]) >> 4) - 8;
                 niddle1 = (0b00001111 & dedomena[j]) - 8;
                 differences += niddle + ",";
                 differences += niddle1 + ",";
                 Sample1 = niddle + Sample2;
                 Sample2 = niddle1 + Sample1;
                 Samples1.add(Sample1);
                 Samples1.add(Sample2);
                 sum = sum + Sample1 + Sample2;
              }
              average = sum / ( 2 * length);
              for (int i = 0; i < Samples1.size(); i++){
                 Samples.add((byte) (Samples1.get(i) - average));  // Mean Value must be 0, for each
package
                 samples += Samples.get(i) + ",";
```

```
                }
                Samples1.clear(); // For each new loop, PreSamples are empty
                packages++;
            } catch (Exception x) {
                System.err.println("(Audio) Package didn't arrive. ");
                System.out.println(x);
                exit(-1);
            }
        } while (packages < number_of_packets);
        receive.close();
        AudioFormat NON_adaptive = new AudioFormat(8000, bits, 1, true, false);
        SourceDataLine finalconsole = AudioSystem.getSourceDataLine(NON_adaptive);
        //energopoieitai i exodos, to antikeimeno morfopoiisis tou ixou = NON_adaptive
        // kai to megethos tis eswterikis mnimis = 3200 (audiobuffer)
        finalconsole.open(NON_adaptive,32000);
        finalconsole.start();
        //methodos write() -> gia anaparagwgi ixou, prwto orisma -> ta deigmata ixou pros
anaparagwgi,
        //deutero orisma arxi kai trito mikos -> tou dianismatos pros anaparagwgi
        finalconsole.write(convertBytes(Samples),0,256*packages);
        //apenergopoieitai prosorina to antikeimeno tou ixou
        finalconsole.stop();
        //apenergopoieitai plirws to antikeimeno tou ixou
        finalconsole.close();

        try {
            samples_stream.write(samples.getBytes());
            samples_stream.close();
            differences_stream.write(differences.getBytes());
            differences_stream.close();
            ByteArrayInputStream dedomena1 = new ByteArrayInputStream(convertBytes(Samples));
            AudioInputStream clip = new AudioInputStream(dedomena1, NON_adaptive,
Samples.size());
            AudioSystem.write(clip, AudioFileFormat.Type.WAVE, gen);
```

```
        } catch (IOException ioe) {
            throw new IllegalArgumentException(ioe);
        }
    System.out.println("The audio ended.");
  }



  public static void Aq(int bits, String option, String encode,String audio) throws IOException,
LineUnavailableException {

                    DatagramSocket socket = new DatagramSocket();
                    DatagramSocket receive = new DatagramSocket(client_port);
                    date = LocalDate.now();
                    byte[] txbuffer ;
                    byte[]hostIP = { (byte)155,(byte)207,(byte)18,(byte)208};
                    InetAddress hostAddress = InetAddress.getByAddress(hostIP);
                    System.out.println("The Aq audio has started at "+ date);
                    int number_of_packets =999;
    String code = audio + encode + option + number_of_packets;
    File gen = new File(path + "/aq.wav");
    File deigmata = new File(path + "/aq_samples.csv");
    File differences1 = new File(path + "/aq_differences.csv");
    FileOutputStream sample_stream = new FileOutputStream(deigmata);
    FileOutputStream differences_stream = new FileOutputStream(differences1);
    File average = new File(path + "/mean.csv");
    File step1 = new File(path + "/step.csv");
    FileOutputStream mean_stream = new FileOutputStream(average);
    FileOutputStream step_stream = new FileOutputStream(step1);

    txbuffer = code.getBytes();
    DatagramPacket p = new DatagramPacket(txbuffer, txbuffer.length, hostAddress, server_port);
    receive.setSoTimeout(8000);
    byte[] rxbuffer = new byte[2048];
```

```
socket.send(p);
//metavlites
int packages = 0;
int length1, temp_niddle1, temp_niddle2, Sample1, Sample2, check;
String audio_samples = "", audio_differences = "", average2 = "", step = "";
ArrayList<Byte> Samples1 = new ArrayList<>();
ArrayList<Byte> Samples2 = new ArrayList<>();
byte[] data;
byte[] temp = new byte[4];
byte pos1;
do {
   DatagramPacket q = new DatagramPacket(rxbuffer, rxbuffer.length);
   try {
      check = 0;
      receive.receive(q);
      length1 = q.getLength();
      data = q.getData();

      pos1 = (byte)( ( data[1] & 0b10000000) !=0 ? 0xFF : 0x00);
      temp[3] = pos1;
      temp[2] = pos1;
      temp[1] = data[1];
      temp[0] = data[0];
      int average3 = ByteBuffer.wrap(temp).order(ByteOrder.LITTLE_ENDIAN).getInt();
      pos1 = (byte)( ( data[3] & 0b10000000) !=0 ? 0xFF : 0x00);
      temp[3] = pos1;
      temp[2] = pos1;
      temp[1] = data[3];
      temp[0] = data[2];
      int quan_step = ByteBuffer.wrap(temp).order(ByteOrder.LITTLE_ENDIAN).getInt();
      average2 += average3 + ","; step += quan_step + ",";

      for (int j = 4; j < length1; j++) {
```

```
            temp_niddle1 =  (0x0000000F & data[j]) - 8 ;
            temp_niddle2 = ((0x000000F0 & data[j]) >> 4 ) - 8;
            audio_differences += temp_niddle1 + ","; audio_differences += temp_niddle2 + ",";
            Sample1 = (temp_niddle2 * quan_step) + check + average3;
            Sample2 = (temp_niddle1 * quan_step) + (temp_niddle2 * quan_step) + average3;
            check = temp_niddle1 * quan_step;
            Samples1.add((byte) ( Sample1 & 0x000000FF));
            Samples1.add((byte) ((Sample1 & 0x0000FF00) >> 8));
            Samples1.add((byte) ( Sample2 & 0x000000FF));
            Samples1.add((byte) ((Sample2 & 0x0000FF00) >> 8));
        }
        for (int i = 0; i <  Samples1.size(); i++){
            Samples2.add(Samples1.get(i));
            audio_samples += Samples1.get(i) + ",";
        }
        Samples1.clear();
        packages++;
    }
    catch (Exception x) {
        System.err.println("Audio package didn't arrive.");
        exit(-1);
    }
} while (packages < number_of_packets);

AudioFormat adaptive = new AudioFormat(8000, 16, 1, true, false);

SourceDataLine lineOut = AudioSystem.getSourceDataLine(adaptive);
lineOut.open(adaptive,32000);
lineOut.start();
lineOut.write(convertBytes(Samples2),0,256 * 2 * packages);
lineOut.stop();
lineOut.close();
```

```
        try {
            sample_stream.write(audio_samples.getBytes());
            sample_stream.close();
            differences_stream.write(audio_differences.getBytes());
            differences_stream.close();
            mean_stream.write(average2.getBytes());
            mean_stream.close();
            step_stream.write(step.getBytes());
            step_stream.close();
            ByteArrayInputStream Audio_Data = new ByteArrayInputStream(convertBytes(Samples2));
            AudioInputStream Audio = new AudioInputStream(Audio_Data, adaptive, Samples2.size());
            AudioSystem.write(Audio, AudioFileFormat.Type.WAVE, gen);
        } catch (IOException ioe) {
            throw new IllegalArgumentException(ioe);
        }

        receive.close();
        System.out.println("The packages are arrived.");
    }


    private static byte[] convertBytes(ArrayList<Byte> bytes) {
        byte[] k = new byte[bytes.size()];
        Iterator<Byte> iterator = bytes.iterator();
        for (int i = 0; i < k.length; i++) {
            k[i] = iterator.next().byteValue();
        }
        return k;
    }

//-----------------------------------------------------Ithakicopter-----------------------------------------------------------

    public static void IthakiCopter(int level) throws IOException {
```

```
            date = LocalDate.now();
            System.out.println("The helicopter starts at " + date);
        File CopterTelemetry = new File(path + "/CopterTelemetry.csv");
        FileOutputStream copter = new FileOutputStream(CopterTelemetry);

        byte[]hostIP = { (byte)155,(byte)207,(byte)18,(byte)208};
        InetAddress hostAddress = InetAddress.getByAddress(hostIP);
        Socket console = new Socket(hostAddress, 38048);
        BufferedReader Input = new BufferedReader(new
InputStreamReader(console.getInputStream()));
        DataOutputStream Output = new DataOutputStream(console.getOutputStream());

        String telemetry, teleconsole = "";
        String LLL, RRR, AAA, TTTT, PPPP;
        int left = 200,right = 200;
        try {
          for(int times = 0; times < 300; times++) {
            telemetry = Input.readLine();
            Output.writeBytes("AUTO FLIGHTLEVEL=" + level + " LMOTOR=" + left + " RMOTOR=" + right
+ " PILOT \r\n");
            if (telemetry.contains("ITHAKICOPTER")) {
              LLL = telemetry.substring(20, 23);
              RRR = telemetry.substring(31, 34);
              AAA = telemetry.substring(44, 47);
              TTTT = telemetry.substring(60, 66);
              PPPP = telemetry.substring(76, 83);
              teleconsole += LLL + "," + RRR + "," + AAA + "," + TTTT + "," + PPPP + "\r\n";
            }
          }
        }catch (Exception e){
          System.out.println(e);
        }
        try {
```

```
            copter.write(teleconsole.getBytes());
            copter.close();
        } catch (IOException x) {
            System.out.println("Ops something went wrong.");
        }
        System.out.println("The data ended.");
    }
//----------------------------------------------------------Vehicle-----------------------------------------------------------

    public static void Vehicle(int selection,String vehicle) throws  IOException{

            DatagramSocket socket = new DatagramSocket();
            DatagramSocket receive = new DatagramSocket(client_port);
            date = LocalDate.now();
            byte[] txbuffer ;
            byte[]hostIP = { (byte)155,(byte)207,(byte)18,(byte)208};
                    InetAddress hostAddress = InetAddress.getByAddress(hostIP);

        File Vehicle_OBD = null;
        String code;
        code = vehicle + "OBD=" + vehicleoption[selection];
         //Menu
              if (selection==0) {
            System.out.println("The packages with code " + code + " has started at "+ date);
            Vehicle_OBD = new File(path + "/vehicle_runtime.csv");
             }
             else if(selection==1) {
                    System.out.println("The packages with code " + code + " has started at "+ date);
            Vehicle_OBD = new File(path + "/vehicle_IntakeTemp.csv");
          }
          else if (selection ==2) {
            System.out.println("The packages with code " + code + " has started at "+ date);
            Vehicle_OBD = new File(path + "/vehicle_throttle.csv");
```

```
    }
    else if(selection==3) {
        System.out.println("The packages with code " + code + " has started at "+ date);
        Vehicle_OBD = new File(path + "/vehicle_RPM.csv");
    }
    else if(selection==4) {
        System.out.println("The packages with code " + code + " has started at "+ date);
        Vehicle_OBD = new File(path + "/vehicle_speed.csv");
    }
    else if(selection==5) {
        System.out.println("The packages with code " + code + " has started at "+ date);
        Vehicle_OBD = new File(path + "/vehicle_CoolantTemp.csv");
    }

FileOutputStream vehicle_stream = new FileOutputStream(Vehicle_OBD);

txbuffer = code.getBytes();
DatagramPacket p = new DatagramPacket(txbuffer, txbuffer.length, hostAddress, server_port);
receive.setSoTimeout(8000);
byte[] rxbuffer = new byte[2048];

ArrayList<Double> vehicle_packages = new ArrayList<>();
String nibble1, nibble2, vehicle_value = "";
int value_XX = 0;
int value_YY = 0;
long start = System.currentTimeMillis();
while (System.currentTimeMillis() - start < 240000) {
    socket.send(p);
    DatagramPacket q = new DatagramPacket(rxbuffer, rxbuffer.length);
    try {
        receive.receive(q);
        int length1 = q.getLength();
        byte[] data = q.getData();
```

```
if (length1 == 11) {
   nibble1 = "" + (char) data[6] + (char) data[7];
   nibble2 = "" + (char) data[9] + (char) data[10];

   value_XX = Integer.parseInt(nibble1, 16);
   value_YY = Integer.parseInt(nibble2, 16);
}else if (length1 == 8){
   nibble1 = "" + (char) data[6] + (char) data[7];

   value_XX = Integer.parseInt(nibble1, 16);
}else {
   System.err.println("Unexpected input from OBD.");
   exit(-1);
}

   if(selection==0) {
            vehicle_packages.add((double) (256 * value_XX + value_YY));
   }
   else if(selection==1) {
            vehicle_packages.add((double) (value_XX - 40));
   }
   else if(selection==2) {
            vehicle_packages.add((double) (value_XX * 100 / 255));
   }
   else if(selection==3) {
            vehicle_packages.add((double) (( (value_XX * 256) + value_YY) / 4));
   }
   else if(selection==4) {
            vehicle_packages.add((double) value_XX);
   }
   else if(selection==5) {
            vehicle_packages.add((double) (value_XX - 40));
   }
```

```
            } catch (Exception x) {
                System.err.println("OBD package didn't arrive.");
            }
        }
        for (Double v_package : vehicle_packages) {
            vehicle_value += v_package + ",";
        }
        try {
            vehicle_stream.write(vehicle_value.getBytes());
            vehicle_stream.close();
        } catch (IOException x) {
            System.out.println("Ops something went wrong.");
        }
        receive.close();
        System.out.println("The packages are arrived");
    }
}
```