

```

/*
 * To change this license header, choose License Headers in Project
Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package seafarm;

import seafarm.RovReceiveDataObservable;
import java.io.IOException;
import java.net.*;
import java.util.Arrays;

/**
 * Gets data from Raspberry throu UDP Connection
 * @author Jørgen
 */
public class RovUDPreceiver implements Runnable {

    private RovReceiveDataObservable observer;
    private DatagramSocket receiveSocket;
    private int port;
    private Thread t;

    /**
     * Sets ports for connection
     * @param receiveDataObs
     * @param port
     */
    public RovUDPreceiver(RovReceiveDataObservable receiveDataObs, int
port) {
        this.observer = receiveDataObs;
        this.port = port;
    }

    /**
     * create thread and starts it
     */
    public void start() {
        t = new Thread(this, "UDPreceiverThread");
        t.start();
    }
}

```

```

/**
 * getting information on the UDP connection reading the sensor values
sendt from Raspberry
 */
public void run() {
    if (observer != null) {
        try{
            Thread.sleep(10000);
        }catch (Exception e) {
            System.out.println(e);
        }
        try {

            receiveSocket = new DatagramSocket(port);
            DatagramPacket receivePacket = new DatagramPacket(new
byte[25], 25);

            while (observer.shouldChildOfThisRun()) {
                receiveSocket.receive(receivePacket);
                observer.setData(receivePacket.getData());
                //System.out.println(Arrays.toString(receivePacket.getDa
ata()) + " Fra Rasberry");

            }
        } catch (IOException e) {
            System.out.println(e);
        } finally {
            receiveSocket.close();
            System.out.println("receivesocket closed");
        }
    } else {
        System.out.println("receive datahandler not created in
udpreceiver thread");
    }
}
}

```