Communication, Bluetooth, WiFi and distributed programming

Henning Christiansen

Robotics course, March 21, 2018

Overview

Until now

- Run programs on EV3, written and compiled on PC
- Use sensors
- Threads and virtual sensors
- Behaviours

Today:

- Have EV3 to communicate with other devices
- Why and how
- Exercise: Add a behavior to your behavioral robot that depends on communication with another device

Bluetooth and Wi-Fi technologies

- Background: read Wikipedia pages (moodle) for background
- How to set op communication involving EV3: See chapters 12+13 in the book

- Bluetooth: Lejos, EV3 and most PCs support it, easy to set up
- Wi-FI: Lejos supports Wi-Fi dongle for EV3; most PCs support it.
  - Requires an additional device, a router (special device or you smartphone)

- Important: When once set up, the choice of Bluetooth/Wi-Fi is transparent for the programmer
- Discuss: Pros & cons? Security, reliability?? …??

EV3 <—> PC

- Computer takes role of computer brain, sends commands to EV3 and receives sensor data
  - Or better: let robot take care of elementary behaviours
    - e.g. computer says “move to position x,y”, robot does so, avoiding holes, big obstacles, get up if it falls, etc.; sends only summarized reports back to PC
  - With several EV3s, this may work as “one robot” with “distributed body”
- EV3 may be in control of the robot and use PC as service provider
- External localization machinery, e.g., Qualisys Motion Capture System
  - ***NBNBNB: We have a lab at RUC with this; a good idea for miniproject to get it to work with EV3 – requires you are a good programmer.***

Robot <—> robot

- Two or more robots may coordinate their effort exchanging messages about what they sense and which subtasks they believe to have accomplished.

- Two or more EV3s may be part of the same physical robot.
  - Use more motors and sensors
  - More computational power, e.g., have on EV3 allocated to one specific, critical behaviour

EV3 <—> Arduino

Use more sensors and actuators

- especially such that are not supported by LeJOS and-or can be hand tailored with Arduino

NB: For the price of one Mindstorms infrared sensor, you can get an Arduino and a whole array of infrared sensors

- But no click-click and you need to write some low-level code to interpret the sensors

A sample program (Bagnall pp. 205–6)

PC asking the EV3: *What is your current battery level?*

Notice roles:

EV3 is the ***server***

- receiving requests and answering them

PC is ***client***

- contacts the server, asks a question and receives the answer

NB: It worked for your teacher in very first attempt!!

Server program running on EV3

```
public class EV3Server {
  public static final int port = 1234;
  public static void main(String[] args) throws IOException {
    ServerSocket server=  new ServerSocket(port);
    System.out.println("Awaiting client..");
    Socket client = server.accept();
    System.out.println("CONNECTED");
    OutputStream out = client.getOutputStream();
    DataOutputStream dOut = new DataOutputStream(out);
    dOut.writeUTF("Battery: " + Battery. getVoltage());
    dOut.flush();
    server.close();
  }
}
```

Client program running on PC

```
public class PCClient
 {  public static void main(String[] args) throws IOException {
    String ip = "10.0.1.1"; // Bluetooth
    if(args.length > 0) ip = args[0];
    Socket sock = new Socket(ip, EV3Server.port);
    System.out.println("Connected");
    InputStream in = sock.getInputStream();
    DataInputStream dIn = new DataInputStream(in);
    String str = dIn.readUTF();
    System.out.println(str);
    Sound.twoBeeps();
    sock.close();
  }
}
```