Robotics and Autonomous Systems

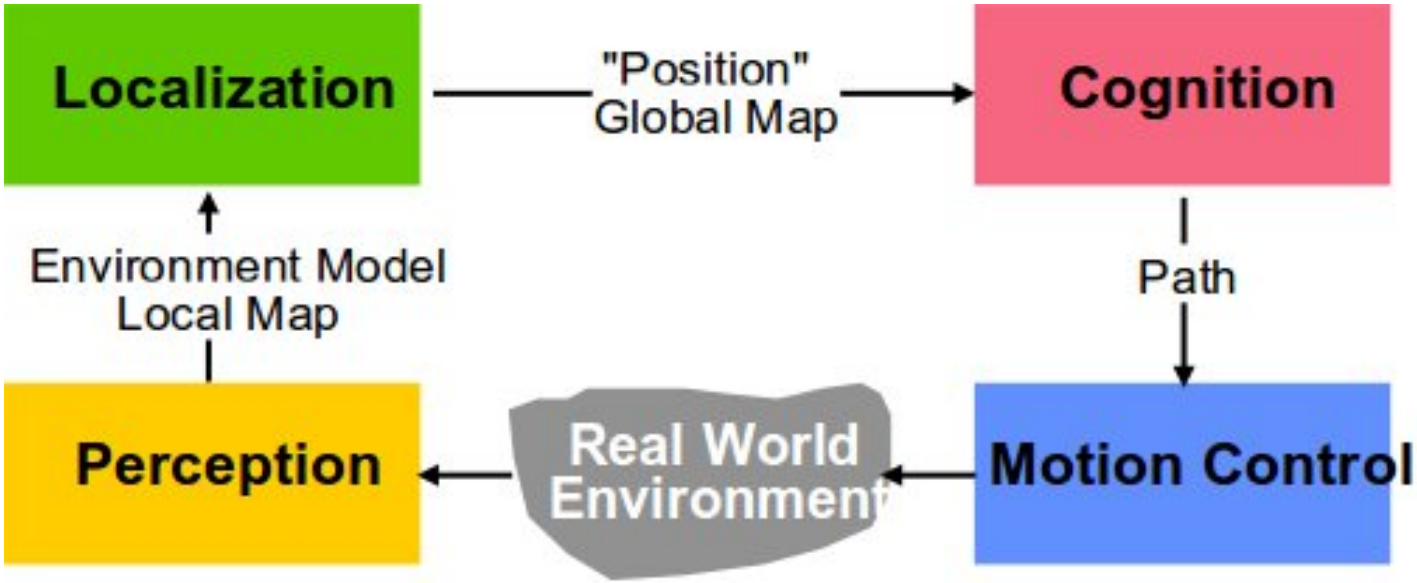
Lecture 14: Communication and other useful things

Richard Williams

Department of Computer Science
University of Liverpool

UNIVERSITY OF LIVERPOOL

Today



- We will cover things from all parts of robotics

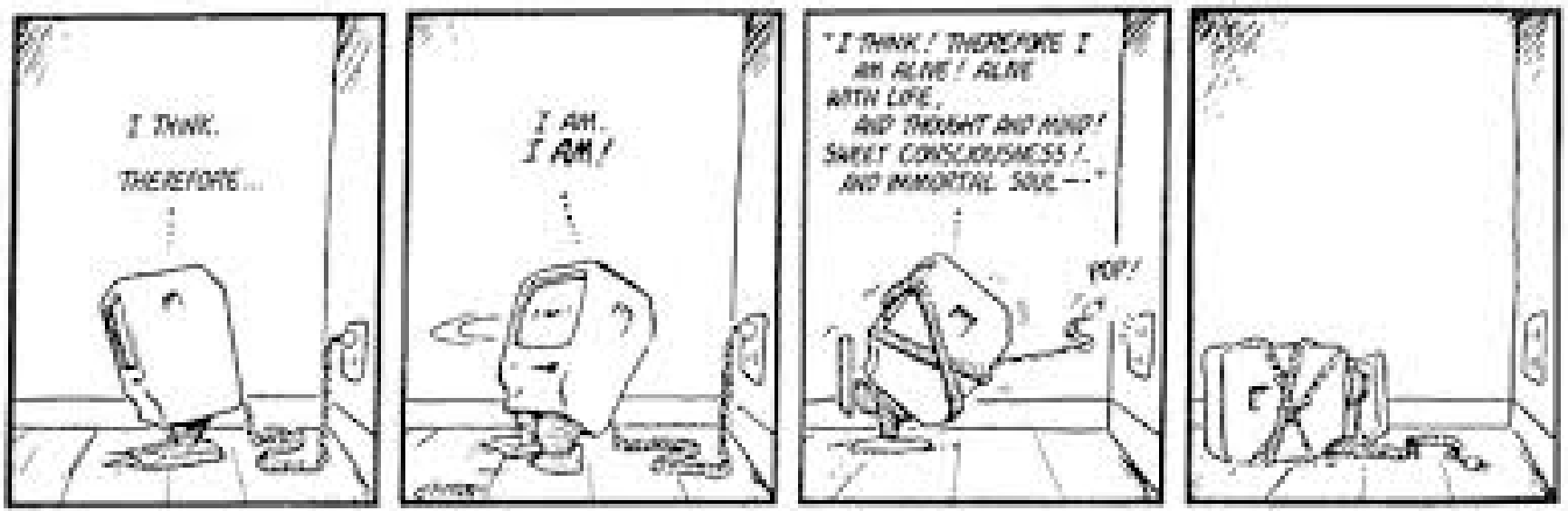
Today

- Basically a bit of a ragbag of topics
  - All LeJOS stuff
- Things that will help with the assignments.

Communication on the NXT

- Connectivity:
  - USB cable
    Fast, reliable but limited by cable range.
  - Bluetooth
    Slow, unreliable but flexible.
- Connectivity via Bluetooth is achieved through the BlueCove Java library for Bluetooth.

Relying on the cable causes problems



- Bloom County, Berkeley Breathed

Bluetooth

- Bluetooth was originally conceived as a wireless alternative to RS-232 data cables.
- Developed by Eriksson in 1994.
- Bluetooth technology exchanges data over short distances using radio transmissions.
- Bluetooth technology operates at 2.4 to 2.485 GHz.
- The 2.4 GHz ISM band is available and unlicensed in most countries.
  - Unlicensed = heavily used

Named after

- Harald “Bluetooth” Gormsson (Haraldr blátǫnn Gormsson, Old Norse)
- King of Denmark around 960-980.
- Unified several Scandinavian kingdoms.

Named after

Bluetooth

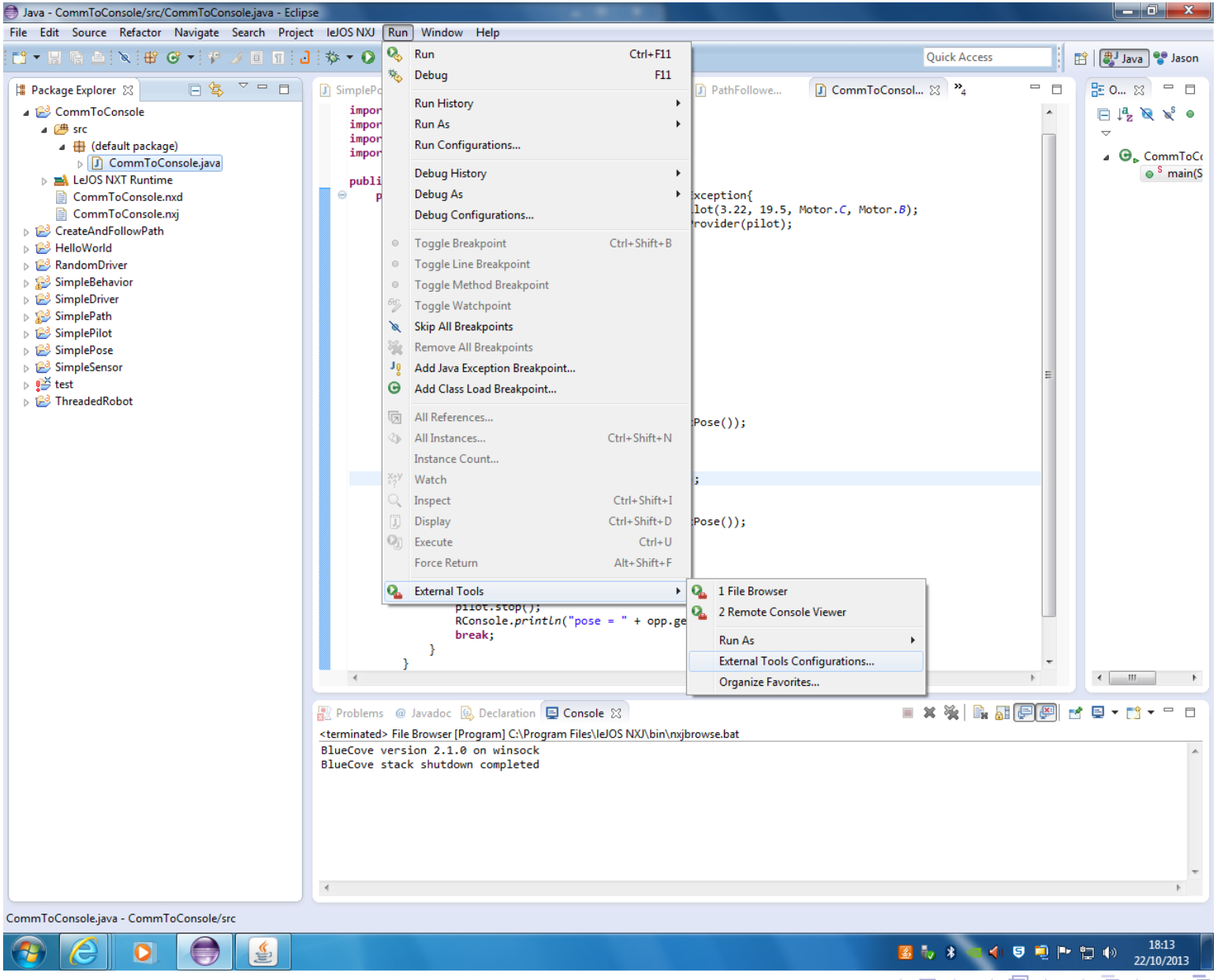
- Communication works as follows:
  - A connection is established by an Initiator to a Receiver (who waits for a connection to be established).
  - Once a connection is established, both ends can open input and output streams, read and write data.
  - The Initiator program is usually assumed to run on a PC
- Here, we will focus on how the NXT brick can communicate to the console on your PC via Bluetooth
- Can also connect two NXTs

LeJOS communication tools

- LeJOS comes with a number of tools for communicating between a PC and the NXT brick.
  - `nxjbrowse`—a browser for nxj files on the NXT brick
  - `nxjconsoleviewer`—GUI viewer for RConsole output
  - `nxjcontroller`—extends the ConsoleViewer with monitoring functionalities for `.nxj` programs
- These can all be installed and run from Eclipse (install via the "External Tool Configuration" toolbar)

```
Run > External Tools
      > External Tools Configuration
```

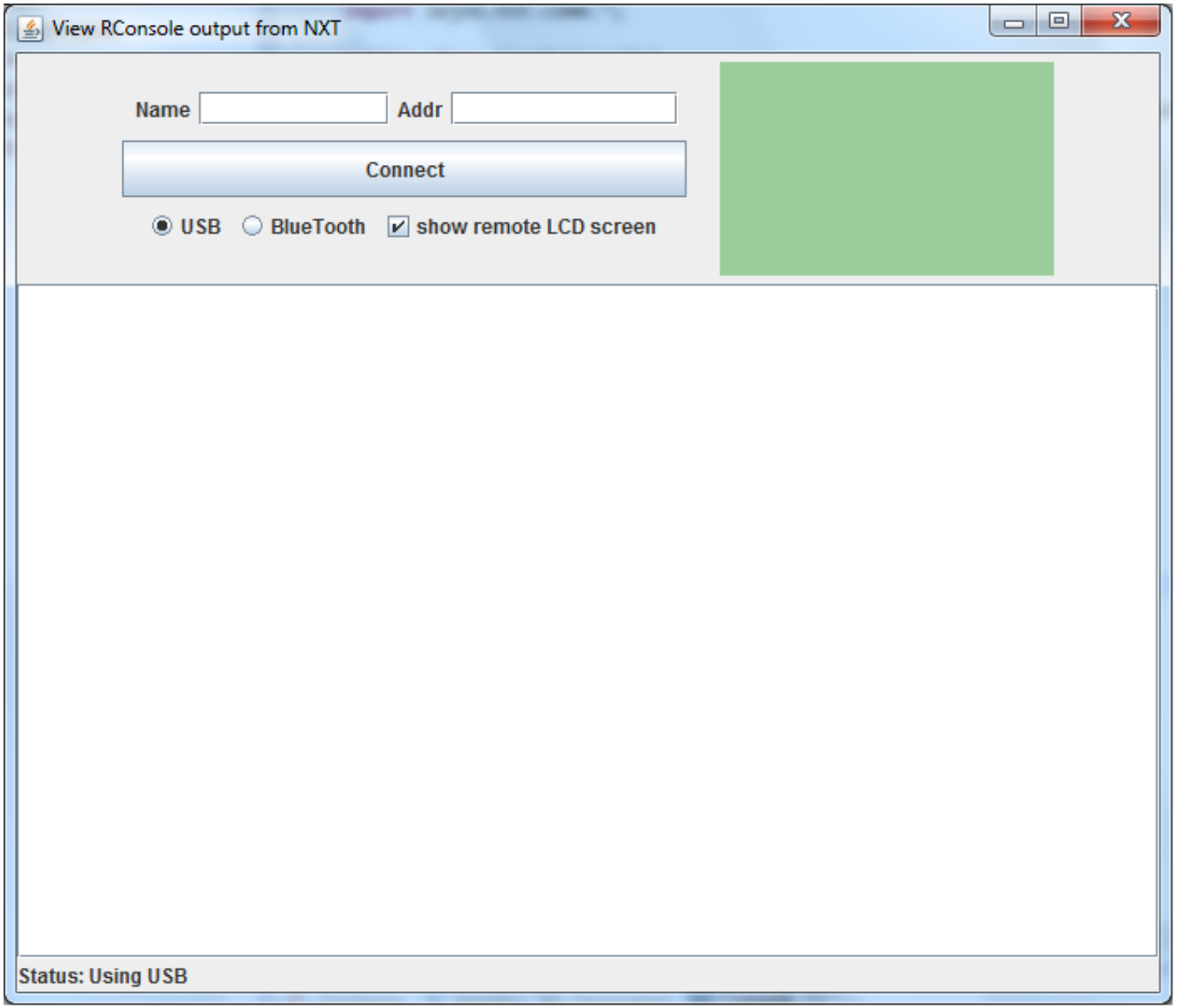
LeJOS communication tools



LeJOS communication tools

- When you do this, you will need to browse for the relevant files: `Program File > leJOS NXJ > bin`
- The files are `.bat` files.
- Then you can run these tools from the `Run` menu.

LeJOS communication tools



- For example, the remote console

LeJOS communication tools

- Can’t do anything until you connect to the robot.
- PS. Can connect through the USB and run all the tools.

Setting up to communicate by Bluetooth

- Setting up a connection between PC and robot is a pain.
- There are four steps before you do this for the first time:
  1. Get NXT Bluetooth PIN
  2. Turn on Bluetooth on NXT
  3. Pair Bluetooth and PC
  4. Open up a connection.

Step 1

- First you have to mess with the connection from the NXT side.
- Go to
  `Bluetooth > Change PIN`
  on the NXT menu system.
- This will display a number.
- On my robot this was:
  1 2 3 4
  which looks like the default to me.
- Look this up and write it down.

Step 2

- Turn on Bluetooth on the robot.
- Go to
  `Bluetooth`
  on the NXT menu.
- You may have to repeat this periodically.

Step 3

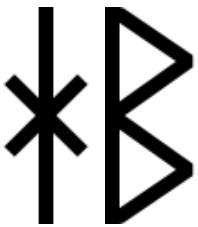
- Set up the PC to talk to the robot.
- Find the bluetooth symbol on the menubar at the bottom of the screen.

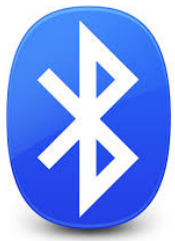
- Click it and pick the “Add a device” option.
- The dialogue should pick up your robot (if the bluetooth is on).
- It will also need a “pairing code”.
- This is the PIN you wrote down before.

Aside

- The blue tooth logo is the merge of pair of runes.

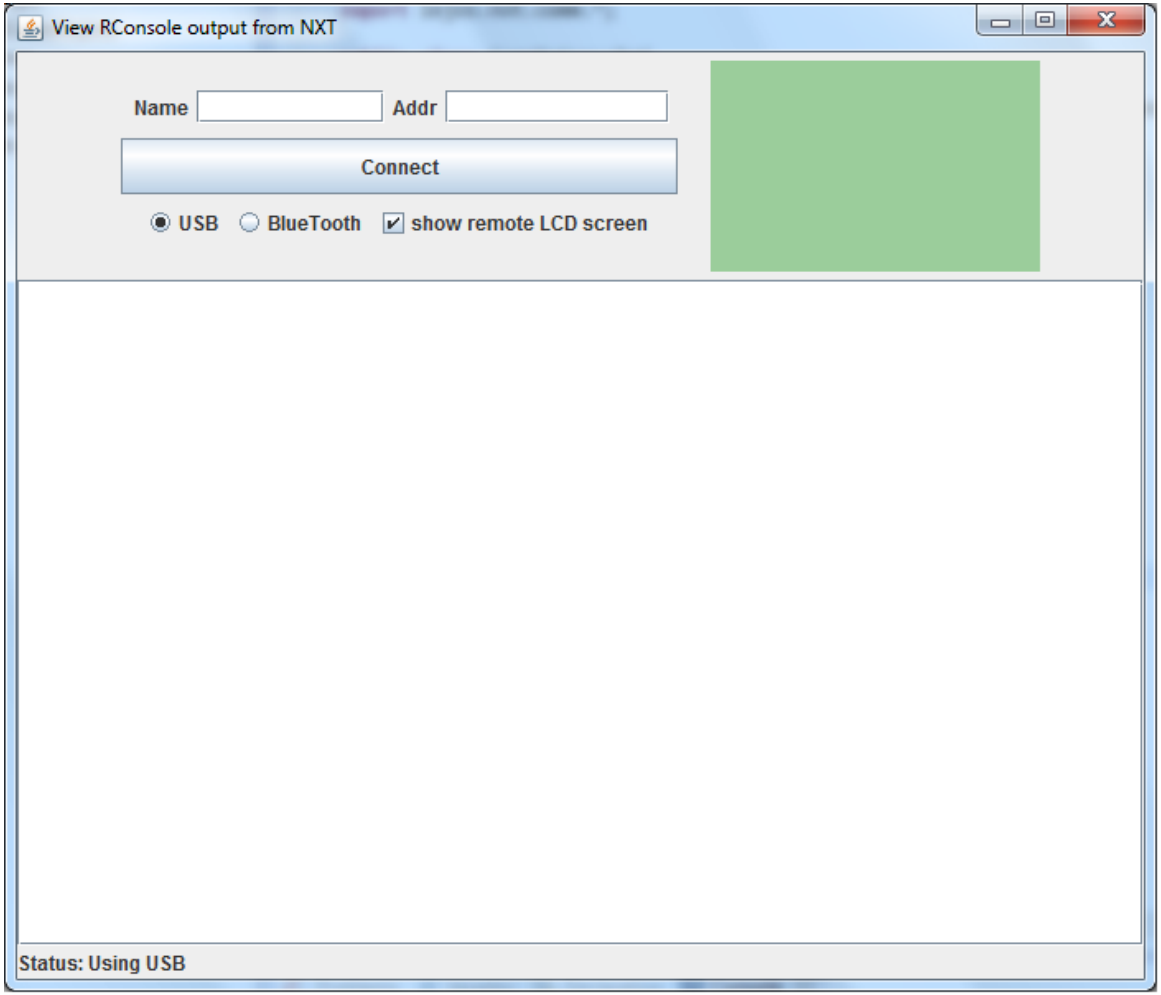
- The rune for h, and the rune for b



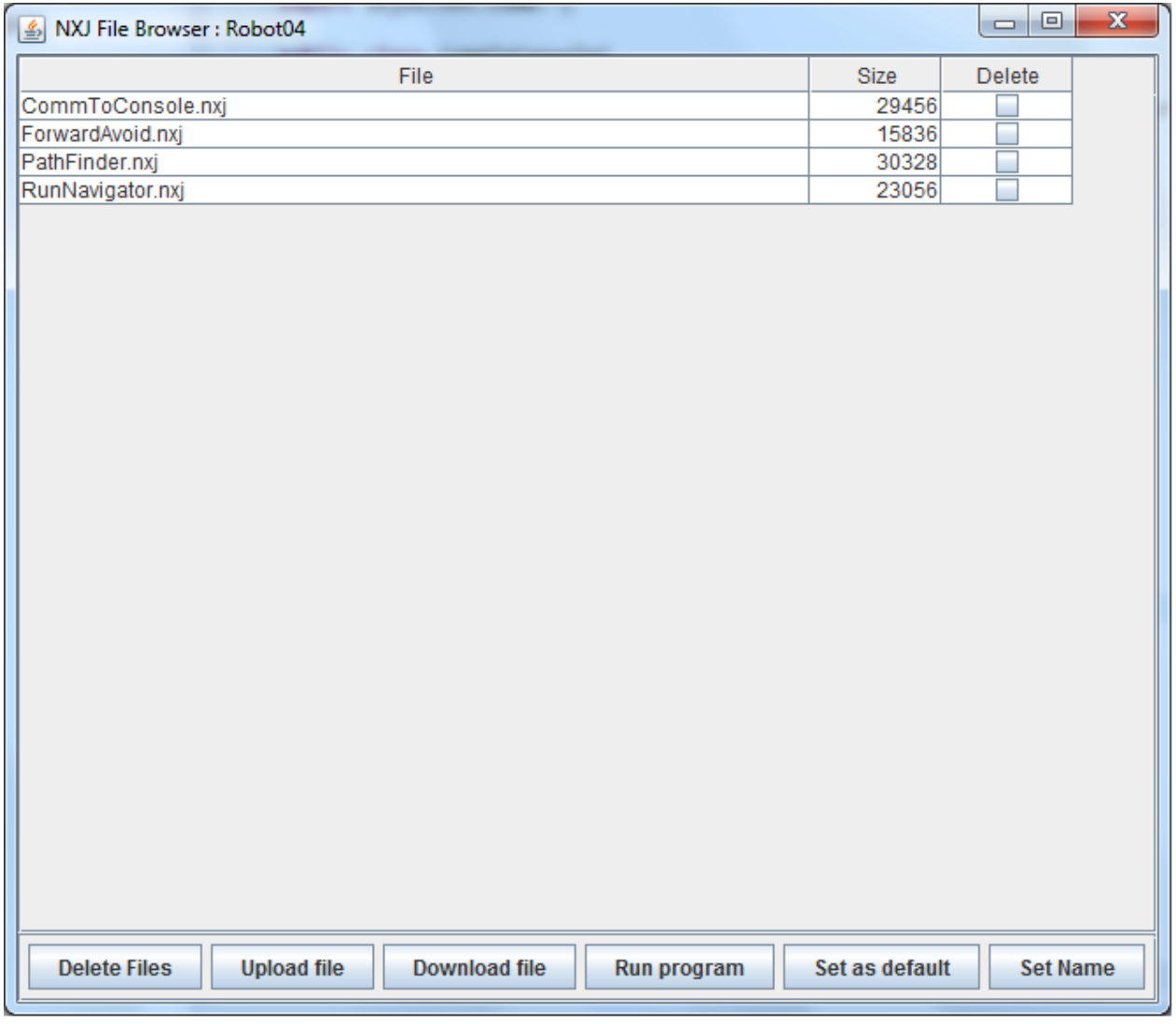


Step 4

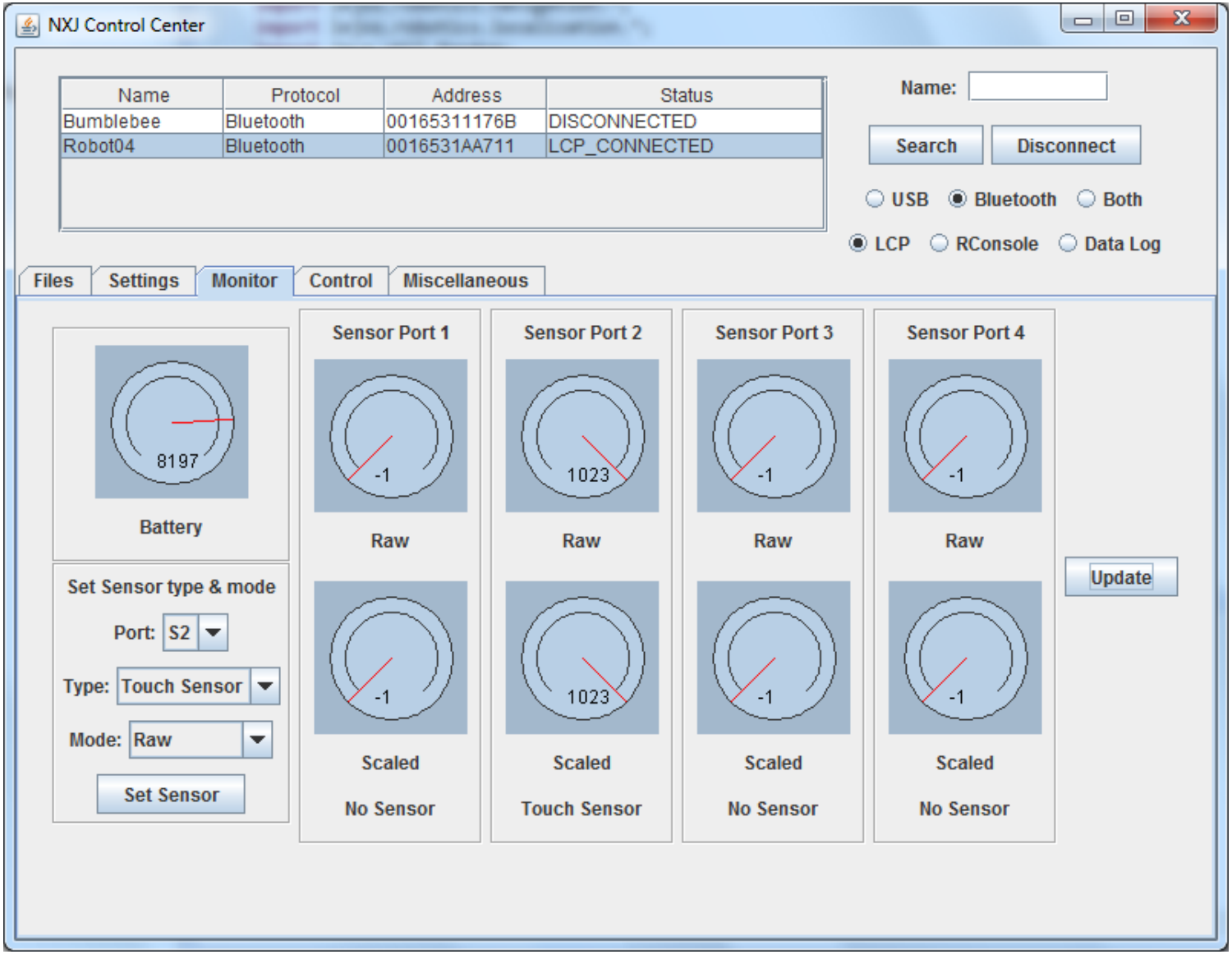
- Now you can open up a connection.



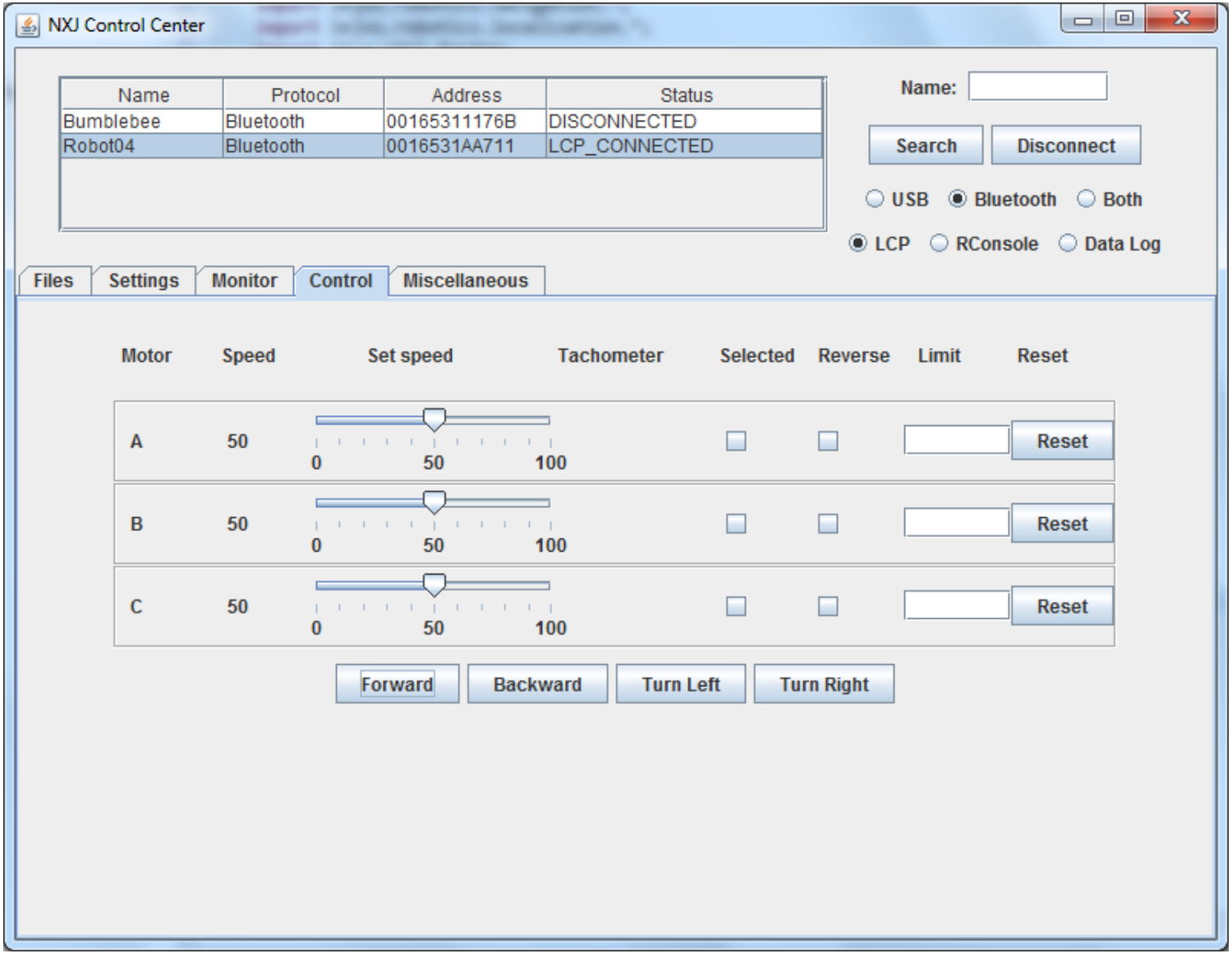
File Browser



Controller


NXJ Control Center
Name
Protocol
Address
Status
Bumblebee
Bluetooth
00165311176B
DISCONNECTED
Robot04
Bluetooth
0016531AA711
LCP_CONNECTED
Name:
Search
Disconnect
USB
Bluetooth
Both
LCP
RConsole
Data Log
Files
Settings
Monitor
Control
Miscellaneous
8197
Battery
Set Sensor type & mode
Port:
S2
Type:
Touch Sensor
Mode:
Raw
Set Sensor
Sensor Port 1
Sensor Port 2
Sensor Port 3
Sensor Port 4
-1
1023
-1
-1
Raw
Raw
Raw
Raw
-1
1023
-1
-1
Scaled
Scaled
Scaled
Scaled
No Sensor
Touch Sensor
No Sensor
No Sensor
Update

Controller


NXJ Control Center
Name
Protocol
Address
Status
Bumblebee
Bluetooth
00165311176B
DISCONNECTED
Robot04
Bluetooth
0016531AA711
LCP_CONNECTED
Name:
Search
Disconnect
USB
Bluetooth
Both
LCP
RConsole
Data Log
Files
Settings
Monitor
Control
Miscellaneous
Motor
Speed
Set speed
Tachometer
Selected
Reverse
Limit
Reset
A
50
0
50
100
Reset
B
50
0
50
100
Reset
C
50
0
50
100
Reset
Forward
Backward
Turn Left
Turn Right

Issues with Bluetooth

- Pairing: You may have to go through the setup procedure everytime you use a robot — NXTs change.
- Pairing: You may find your robot is paired with someone else's PC.
- Pairing: You may find your PC is paired with someone else's robot.
- Interference: many bluetooth devices close together:
  - multiple robots,
  - phones,
  - computers,
  - tablets

  will impede communication.
  802.11 also uses the 2.4 GHz band.

Programs that communicate

- Once again, LeJOS (and Java) make doing complicated things quite easy.
- The `RConsole` class gives us a way to communicate from the NXT to the remote console program running on the PC.
- Open a connection, no timeout
  `open()`
- Close a connection
  `close()`
- Send text to the console
  `println(String s)`

Programs that communicate

- Here's part of a program `CommToConsole.java` that you can get from the module website.

```
public class CommToConsole{
  public static void main (String[] args)
     throws Exception{
  DifferentialPilot pilot
    = new DifferentialPilot(3.22, 19.5,
                        Motor.C, Motor.B);
  OdometryPoseProvider opp
    = new OdometryPoseProvider(pilot);

  Random randomGenerator = new Random();
  int move;

  RConsole.open();
```

- Only the last line is necessary to start communication.

Programs that communicate

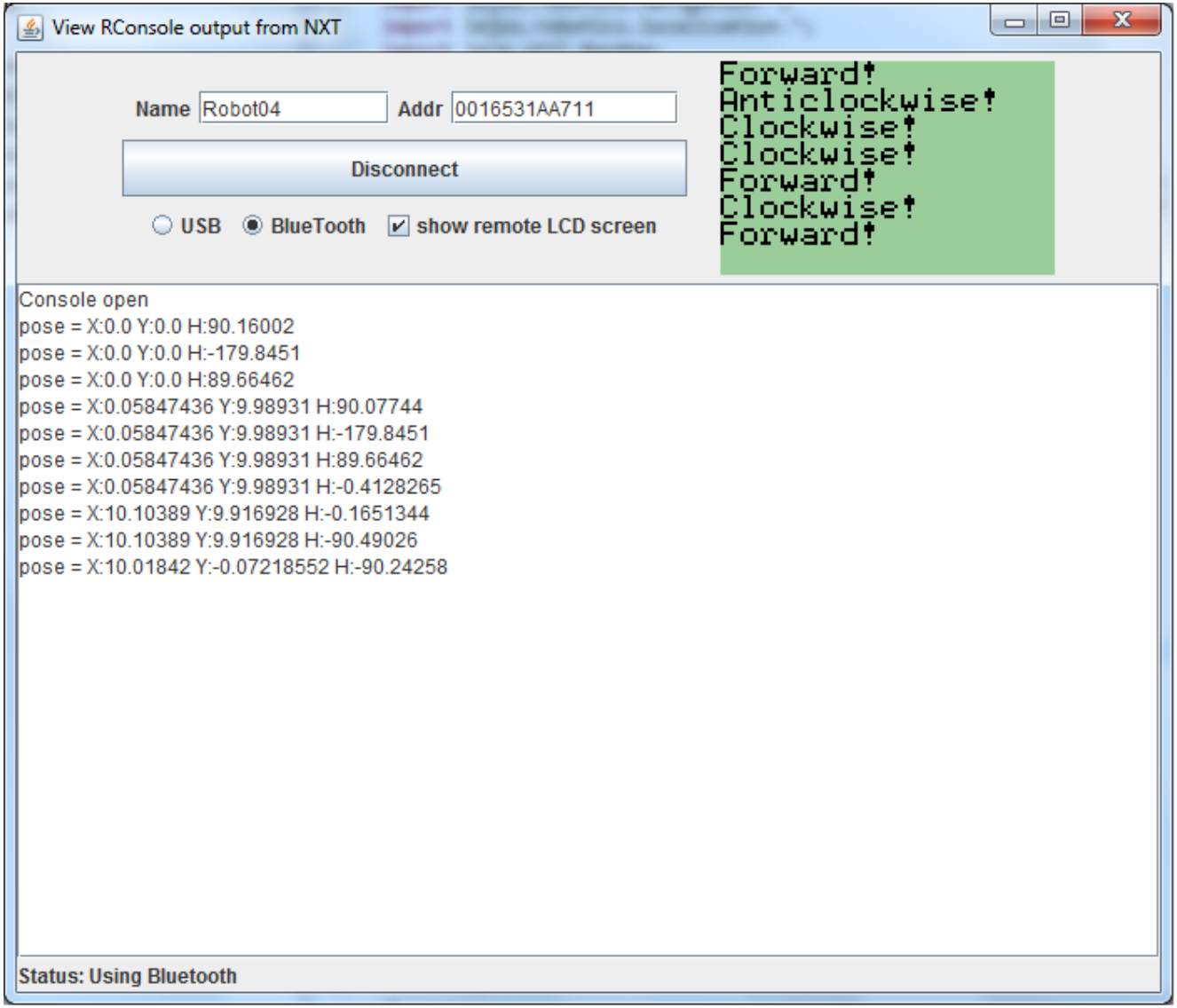
- Here's more of the program:

```
for(int i=0; i<10; i++){
  move = randomGenerator.nextInt(3);
  switch(move){

  case 0:
    System.out.println("Forward!");
    pilot.travel(10);
    pilot.stop();
    RConsole.println("pose = " + opp.getPose());
    break;
```

- Where do the two pieces of text appear?

Programs that communicate

- Here's what a run looks like:



Other functions

- Check if the connection is open:
  `boolean isOpen()`
- Open connection with timeout:
  `openAny(int timeout)`
- Open bluetooth connection
  `openBluetooth(int timeout)`
- Open USB connection:
  `openUSB(int timeout)`

Other functions

- Get a print stream from the console:
  `PrintStream getPrintStream()`
- This allows you to write a program (using the PC API) that runs on the PC and talks to the NXT.
- Just like Remote Console Viewer, Controller, etc.

Listeners

- In addition to communication, there is one major piece that we haven't covered.
- Listeners
- Think of this as allowing us to make full use of concurrency in LeJOS.
  - No longer need to keep a busy watch on the hardware
- Instead, have the hardware tell us when somegthing changes.
- Exactly the same kind of event-driven programming that we have in GUIs.

Listeners

- In a GUI, pressing a button typically leads to an action.
- On the robot:
  - Pressing a button
  - Having a sensor change state

  can lead to an action.

Listeners

- Here is a button listener in LeJOS

```
import lejos.nxt.Button;
import lejos.nxt.ButtonListener;
import lejos.nxt.Motor;
import lejos.nxt.Sound;

public class myButtonListen
     implements ButtonListener{
  public void buttonPressed(Button b){
    Motor.B.stop();
    Motor.C.stop();
  }

  public void buttonReleased(Button b){
    Sound.beepSequence();
  }
}
```

Listeners

- Allows you to interrupt the execution of another thread.

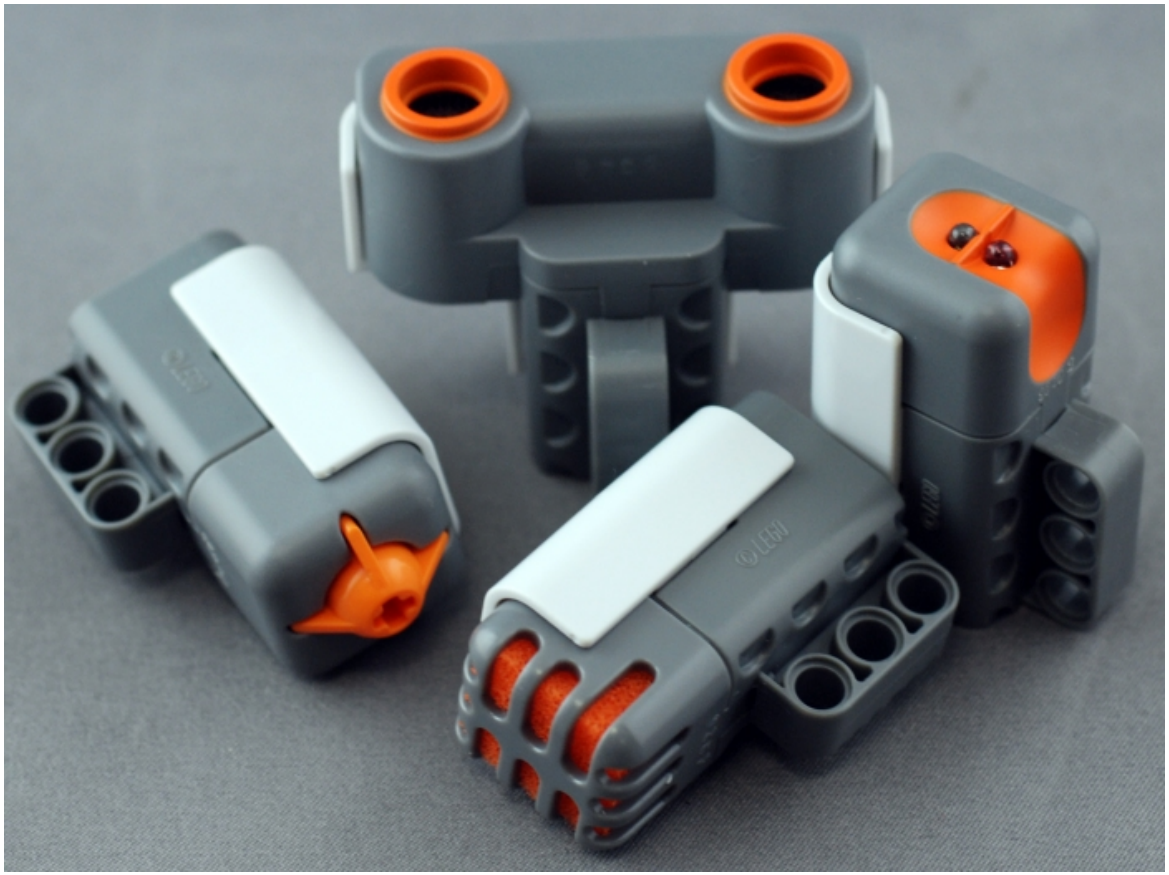
```
public class SimpleDriveWBL{
  public static void main(String[] args){

  Button.ENTER.addButtonListener(new myButtonListen());

  System.out.println("Press any button to start robot");
  Button.waitForAnyPress();
  LCD.clear();
  System.out.println("Press ENTER to stop robot");
  Motor.B.forward();
  Motor.C.forward();
  try{
    Thread.sleep(10000);}
    catch(Exception e){
      // Nothing
    }
}}
```

Listeners

- You can write separate button listeners for each of the four buttons.
- You can write a general button listener that tests the ID of the calling button and acts appropriately.

Listeners

- You can set up listeners for sensors also.



Listeners

- More corectly, you set up a listener for a sensor port:

```
public class DriveWTListen{
  public static void main(String[] args){
  TouchSensor leftBump = new TouchSensor(SensorPort.S2);

  SensorPort.S2.addSensorPortListener(new LTouchListener());

  System.out.println("Press any button to start robot");
  Button.waitForAnyPress();
  LCD.clear();
  System.out.println("Press ENTER to stop robot");
  Motor.B.forward();
  Motor.C.forward();
  try{
    Thread.sleep(10000);}
    catch(Exception e){
      // Nothing
}}}
```

Listeners

- Here the listener is the following:

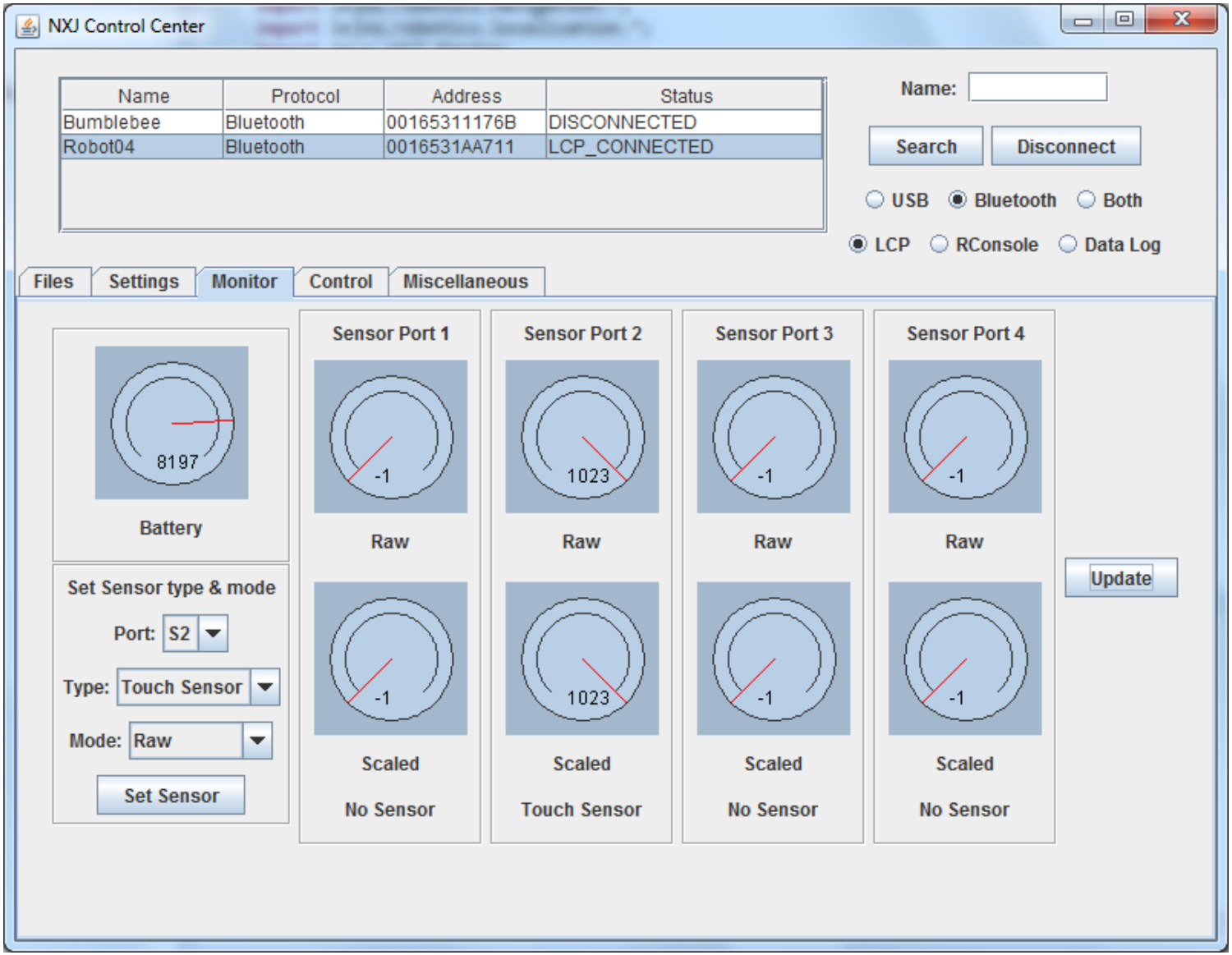
```
public class LTouchListener
     implements SensorPortListener{
  public void stateChanged(SensorPort port,
                   int oldValue, int newValue){
    if(newValue <500){
      Motor.B.stop();
      Motor.C.stop();
    }
  }
}
```

- `stateChanged` is the function that gets called when something happens to the sensor.

Listeners

- The values handled by `stateChanged` are raw values.
- Output of the ADC connected to the sensor.
- Value between 0 and 1023.
- You have to figure out what is a good threshold value.
- The remote controller is handy for this.

Controller


NXJ Control Center
Name
Protocol
Address
Status
Bumblebee
Bluetooth
00165311176B
DISCONNECTED
Robot04
Bluetooth
0016531AA711
LCP_CONNECTED
Name:
Search
Disconnect
USB
Bluetooth
Both
LCP
RConsole
Data Log
Files
Settings
Monitor
Control
Miscellaneous
8197
Battery
Set Sensor type & mode
Port: S2
Type: Touch Sensor
Mode: Raw
Set Sensor
Sensor Port 1
Sensor Port 2
Sensor Port 3
Sensor Port 4
-1
1023
-1
-1
Raw
Raw
Raw
Raw
-1
1023
-1
-1
Scaled
Scaled
Scaled
Scaled
No Sensor
Touch Sensor
No Sensor
No Sensor
Update

Advantages of listeners

- You don't have to think so much about the effect of concurrency.
- No need to write a control loop that checks for, or waits for, a button press to stop motors.
- Write the control loop to do its thing.
- Have the button listener stop the motors.
- Particularly useful if you are using behaviors.
- Listeners turn on and off the semaphores that determine which behaviors can take control.
- Arbitrator then does its selection.

PS

- Most threads set up by LeJOS
  - Pilots
  - Navigation
  - etc

  can be equipped with Listeners.
- Check the API documentation for details.

Summary

- This lecture covered a miscellany of topics that will be useful in the assignment.
- Bluetooth communication.
- Listeners for buttons, sensors etc.
- You should now have all the information you need to complete the first assignment.
- There is still more information at `www.lejos.org`