Design of Embedded Systems (DES)

Jozef Hooman

http://www.cs.ru.nl/~hooman/DES/

Course 8
22 October 2014
Notes

• Last person which leaves the lab asks Harco / Simone / .. to close the lab

• New PCs in lab: login with “labuser” and well-known real-time password

• Better version of USB driver available: http://www.cs.ru.nl/lab/downloads/fantomdriver-1.1.3.zip
Planning

• 29 Oct & 5 Nov: Autumn break
  when needed: **finalize Xenomai part**
  *I am available for help on 29 Oct*
  *abroad 3 – 6 Nov*

• 12 Nov: discuss Rover results +
  explanation Domain Specific Language (DSL)
  **NOTE**: in **HG 01.028**

• 19 Nov: apply DSL to small Lego Rover

• 26 Nov: discuss DSL proposal +
  requirements Mars Rover

• 3 Dec: start development Mars Rover using DSL
Observations

- Not easy to stop immediately with DifferentialPilot – first had to increase acceleration
- Detect bottle; only top part is seen by sensor
- Motor control: "true" field has to be used with care
- Continuous reading of sonar might be useful
- Sound: always blocking; maybe use separate thread
- Dangerous to drive back
- Not all hardware is the same …
  … influence of environment (surface, light, ..)
Color Sensor

**NOT**: ColorHTSensor & other classes for HiTechnic color sensor  
**Use**: **ColorSensor** for LEGO color sensor  
Replaces light sensor, so connected to port S1

```java
LCD.drawString("ID:", 0, 2);
LCD.drawString("RGB:", 0, 3);

int colorValue = color.getColor();
LCD.drawString("rRGB:", 0, 3);
```
Bluetooth (1)

Preparation:

• See Bluetooth menu in:
  http://www.lejos.org/nxt/nxj/tutorial/MenuSystem/MenuSystem.htm

  Go to Bluetooth menu on NXT brick(s), make sure
  – power is on
  – visibility is on

• See samples project (File > Import > General)

  Optional: see info via
  – src / org.lejos.sample.bluestates
    Download and run on NXT: BlueStats.java
Example programs

• See Communications:
  http://www.lejos.org/nxt/nxj/tutorial/Communications/Communications.htm

• Example receive, see samples project
  – src / org.lejos.sample.btreceive
    BTRReceive.java
    ➔ runs on slave

• On slave brick:
  – check PIN
  – check name
Bluetooth (3)

• On **master**:  
  – also here Bluetooth power on, visibility on  
  – add slave to Bluetooth devices of master, see  
  Select: Search/Pair  
  – device searches other BT devices and lists them  
  – select slave you want to connect to and enter PIN of slave  
  – then it should pair successfully  

• Example Sender, see samples project  
  – src / org.lejos.sample.btconnecttest  
    BTConnectTest.java  
    ➔ Change string “name” to name of slave  

• Start BTReceive on slave;  
  **next** BTConnectTest on master
Bluetooth (4)

http://www.lejos.org/nxt/nxj/tutorial/Communications/Communications.htm

Notes:

• Methods to read the stream are blocking; they do not return until data is read
  ➔ to be able to do other tasks use Thread
  see “Threads, Listeners and Events”
  http://www.lejos.org/nxt/nxj/tutorial/ListenersAndEvents/Listeners_Eventso.htm

• Flush output stream to be sure data is transmitted
  Transmission might fail without throwing exception
LeJOS GUI Tools

See **Setting up the leJOS GUI Tools**

- [http://www.lejos.org/nxt/nxj/tutorial/Preliminaries/UsingEclipse.htm#9](http://www.lejos.org/nxt/nxj/tutorial/Preliminaries/UsingEclipse.htm#9)

Use External tools configuration

- [nxjflashg.bat: button to flash firmware already available]
- nxjbrowse - an Explorer for NXJ files
- nxjconsoleviewer - GUI viewer for RConsole debug output
- nxjmonitor - Remote monitoring of programs running on the NXT
- nxjdataviewer - GUI tool to download data logs from NXT
- **nxjcontrol** - a GUI tool that combines the function of all of the above tool, and adds a few more functions.
  
  → *might allow changing the name of the brick (not tested)*

Note: after changes in Java (e.g. using jre instead of jdk) it might be needed to adjust the environment variable LEJOS_NXT_JAVA_HOME
Memory usage

- Java programs execute from flash memory
- Static read-only data is held in flash memory
- Static read-write data is copied to RAM
- Objects are created in a heap that starts at the top of the RAM and grows downwards.
- The Java stack starts at the bottom of free RAM memory and grows up
- A garbage collector frees memory used by unreferenced objects when the heap becomes full
Guideline(s)

• http://sourceforge.net/p/lejos/wiki-nxt/Virtual%20Machine%20Issues/
  In leJOS each use of a string literal will create a new String Object

• Sample “BTRReceive” uses variables initialized with a string constant in LCD.drawString to avoid frequent garbage collection

• The leJOS VM does not implement any memory defragmentation. So it may happen, that memory allocation for large arrays or large object fails, even though the overall free memory seems to be enough.
Limitations

max 255 threads

The following are some of the known limits of the leJOS VM

```
MAX_CLASSES = 256;
MAX_FIELDS = 255;
MAX_METHODS = 255;
MAX_PARAMETER_WORDS = 16;
MAX_SIGNATURES = 4096;
MAX_OPERANDS = 255;
MAX_LOCALS = 255;
MAX_EXCEPTION_HANDLERS = 255;
MAX_CODE = 0xFFFF;
MAX_CONSTANTS = 1024;
MAX_STATIC = 1024;
MAX_FIELD_OFFSET = 4095;
MAX_STRING_CONSTANT_LENGTH = 255;
MAX_DIMS = 7;
```

NXT has 256Kb of Flash
limit programs: 64Kb

65535
Assignment 12 November 2014

• Develop LeJOS programs for two NXT robots such that
  – they stay within black border
  – together they detect all colors
  – give signal by sound/lamp on both rovers when a color is detected
  – both rovers stop and give signal when all colors have been detected
  – collision avoidance and detection (including other robot)

Mail source files + description before Tuesday 11 November 18:00