

Project from Real-Time Systems – Lego Mindstorms EV3

March 13, 2017

Lego Mindstorms

- manufactured by LEGO, <http://mindstorms.lego.com>
- extension of LEGO *Technic* line
- history:
 - RCX, 1998
 - NXT, 2006; NXT 2.0, 2009
 - EV3, 2013
- why LEGO?

EV3 Brick

- 300 MHz 32-bit [ARM9](#) microcontroller by Texas Instruments

EV3 Brick

- 300 MHz 32-bit [ARM9](#) microcontroller by Texas Instruments
- 16MB FLASH memory, 64 MB RAM (cf. 64 kB flash in NXT!), microSDHC slot (up to 32 GB)

EV3 Brick

- 300 MHz 32-bit [ARM9](#) microcontroller by Texas Instruments
- 16MB FLASH memory, 64 MB RAM (cf. 64 kB flash in NXT!), microSDHC slot (up to 32 GB)
- runs a native version of Linux

EV3 Brick

- 300 MHz 32-bit [ARM9](#) microcontroller by Texas Instruments
- 16MB FLASH memory, 64 MB RAM (cf. 64 kB flash in NXT!), microSDHC slot (up to 32 GB)
- runs a native version of Linux
- USB host port, mini USB port, Bluetooth interface (Android and iOS connectivity and apps available)

EV3 Brick

- 300 MHz 32-bit [ARM9](#) microcontroller by Texas Instruments
- 16MB FLASH memory, 64 MB RAM (cf. 64 kB flash in NXT!), microSDHC slot (up to 32 GB)
- runs a native version of Linux
- USB host port, mini USB port, Bluetooth interface (Android and iOS connectivity and apps available)
- 4 and 4 output ports, connection via modified RJ12 cables

EV3 Brick

- 300 MHz 32-bit **ARM9** microcontroller by Texas Instruments
- 16MB FLASH memory, 64 MB RAM (cf. 64 kB flash in NXT!), microSDHC slot (up to 32 GB)
- runs a native version of Linux
- USB host port, mini USB port, Bluetooth interface (Android and iOS connectivity and apps available)
- 4 and 4 output ports, connection via modified RJ12 cables
- loudspeaker
- buttons, display, diodes
- timer doing **1000** ticks per second
- power source: 6 AA batteries

Motors

- **electric motors** (2 large and 1 medium-sized), built-in gear
- synchronization, built-in rotation sensor (accuracy ± 1 deg)
- typical use:
 - set power (0-100), direction (fwd, rev), set on/off, or
 - turn the shaft by a given angle at a given power.



Sensors

- touch sensor - analogue, pressed/not pressed

Sensors

- **touch sensor** - analogue, pressed/not pressed
- **colour sensor** - digital, 3 modes: colour recognition, measuring intensity of ambient/reflected light

Sensors

- **touch sensor** - analogue, pressed/not pressed
- **colour sensor** - digital, 3 modes: colour recognition, measuring intensity of ambient/reflected light
- **infrared sensor** - digital, 2 modes:
 - proximity mode: measures distance to objects (0 – 70 cm), accuracy depends on size, shape and composition of objects
 - beacon mode: measures distance and angle to *IR beacon* (up to 200 cm)
- **IR beacon** - acts as a beacon or an RC device, requires 2 AAA batteries

Digital sensors sample rate 1 kHz.

Sensors

- **touch sensor** - analogue, pressed/not pressed
- **colour sensor** - digital, 3 modes: colour recognition, measuring intensity of ambient/reflected light
- **infrared sensor** - digital, 2 modes:
 - proximity mode: measures distance to objects (0 – 70 cm), accuracy depends on size, shape and composition of objects
 - beacon mode: measures distance and angle to *IR beacon* (up to 200 cm)
- **IR beacon** - acts as a beacon or an RC device, requires 2 AAA batteries

Digital sensors sample rate 1 kHz.

Additional sensors (not bundled):

- **gyro sensor** (or use gyro-capable smartphone...)
- **ultrasound sensor**

Programming Languages and Environments

Visual:

- bundled visual language ([EV3-G](#), based on LabView's [G](#))

“Code-based”:

- [leJOS EV3](#)
 - a [Java](#) Virtual Machine for EV3 brick
 - comes with a well-documented API
 - plug-ins for Eclipse
 - microSDHC card needed for installation
 - EV3 version: 0.9 beta, **recommended**

Programming Languages and Environments

Visual:

- bundled visual language ([EV3-G](#), based on LabView's [G](#))

“Code-based”:

- [leJOS EV3](#)
 - a [Java](#) Virtual Machine for EV3 brick
 - comes with a well-documented API
 - plug-ins for Eclipse
 - microSDHC card needed for installation
 - EV3 version: 0.9 beta, **recommended**
- C-like alternatives:
 - [NXC](#) (Not eXactly C): originally for NXT, IDE for Win (BricxCC), development stopped in 2013, EV3 support experimental
 - [RobotC](#): proprietary (1-year license starts at \$49), IDE for Win.

Programming Languages and Environments II

ev3dev

- <http://www.ev3dev.org/>
- customized Debian (8.0) Linux Distribution
- allows access to EV3's native devices via standard file access
- provides rich a set of libraries and language bindings allowing their use in many standard languages: C++, Lua, Python, C...
- distro comes with several of these languages (Python, Lua), others can be installed (if ARM9 port exists)
- still in development, may require substantial tweaking

Project: Organization

- work in teams
- each team chooses a **leader**
- submit short project **abstract** (up to 1/2 page) by **March 30**
- work :)
- submit a **project report** (up to 5 pages) and **program source**
- **presentation** of results

Submission deadlines: TBA

Project: Requirements

- the robot performs a non-trivial, meaningful and a well-defined task.
- the robot uses at least 2 motors and 2 sensors
- the implementation uses **concurrency** in a meaningful way (at least 2 threads running in parallel)
- the resulting system is a hard real-time system (i.e., a successful completion of the defined task depends on a correct timing)
- implementation in **leJOS** is preferred

Project: Evaluation

Project report:

- up to 5 pages
- describes: the task performed, the implementation, deviations from abstract, **difficulties encountered during implementation**, use of concurrency and timing
- specifies the contribution of individual members (does not have to be equal, but team members may “kick out” work-avoiding colleagues)

Project: Evaluation

Project report:

- up to 5 pages
- describes: the task performed, the implementation, deviations from abstract, **difficulties encountered during implementation**, use of concurrency and timing
- specifies the contribution of individual members (does not have to be equal, but team members may “kick out” work-avoiding colleagues)

Source code:

- preferably non-visual, non-proprietary language
- well-documented source code

Project: Evaluation

Project report:

- up to 5 pages
- describes: the task performed, the implementation, deviations from abstract, **difficulties encountered during implementation**, use of concurrency and timing
- specifies the contribution of individual members (does not have to be equal, but team members may “kick out” work-avoiding colleagues)

Source code:

- preferably non-visual, non-proprietary language
- well-documented source code

Presentation:

- with **demo**, not necessarily by the team leader, **rehearse in advance!**

Project: Evaluation II

Golden rules:

- Purpose of the project is **learning**, not building the “best” robot or a “safe” robot that works well under every circumstance (be a bit ambitious!).

Project: Evaluation II

Golden rules:

- Purpose of the project is **learning**, not building the “best” robot or a “safe” robot that works well under every circumstance (be a bit ambitious!).
- Failures are ok, provided that you transform them into a lesson that can be **shared** with others (but it is nice to have a robot that does at least “something”).

Project: Evaluation II

Golden rules:

- Purpose of the project is **learning**, not building the “best” robot or a “safe” robot that works well under every circumstance (be a bit ambitious!).
- Failures are ok, provided that you transform them into a lesson that can be **shared** with others (but it is nice to have a robot that does at least “something”).
- Show us that you learned something new and that you are ready to discuss your project, and your project will pass.

Project: Evaluation II

Golden rules:

- Purpose of the project is **learning**, not building the “best” robot or a “safe” robot that works well under every circumstance (be a bit ambitious!).
- Failures are ok, provided that you transform them into a lesson that can be **shared** with others (but it is nice to have a robot that does at least “something”).
- Show us that you learned something new and that you are ready to discuss your project, and your project will pass.
- If the project passes, the final grade is not further influenced by it.

Project: Topic

- it is a part of the project to choose an interesting yet doable goal
- search the internet for inspiration (e.g., YouTube: Lego Mindstorms)
- all-time classics: finding, picking up and transporting an object; following a black line; navigation through a maze; Segway; 2-leg (or 4-leg) walker...
- **new ideas are appreciated!**

Quick Start Guide

- unbox :), put unboxed parts into some nicer box
- download and install Lego Mindstorms app, leJOS, or ev3dev
- play around for a while, test the sensors etc.
- read manuals and tutorials on the web (including the official ones)
- build a simple robot and try to run a simple program
- discuss the project, write an abstract