# Lego Mindstorms

Radek Pelánek

Tento projekt je spolufinancován Evropským sociálním fondem a státním rozpočtem České republiky.

EVROPSKÁ UNIE

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

OP Vzdělávání pro konkurenceschopnost

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

# Lego Mindstorms

- produced by Lego, `http://mindstorms.lego.com`
- history
  - 1998: RCX brick
  - 2006: NXT brick
- why?
  - simple, easy to use
  - illustrates main typical features of embedded real-time system

# Lego Mindstorms

# NXT brick

- 3 output motor ports
- 4 input sensor ports
- USB port, bluetooth
- loudspeaker
- buttons
- display
- 32-bit ARM7 microcontroller, 256 Kbytes FLASH, 64 Kbytes RAM
- power source: 6 AA bateries

# Sensors

| | |
|---|---|
| touch | pressed/released |
| sound | measures sound pressure in decibels |
| light | measures the light intensity |
| ultrasonic | measures distance form an object; 0 to 255 cm, (declared) precision +- 3cm |
| third-party | color, compass, temperature, pressure, ... |

# Motors

- setting: direction, speed
- motor synchronization
- built-in rotation sensor (accuracy +- 1 degree)

# Programming Environments

| type of environment | product |
|---|---|
| visual | Mindstorms NXT software<br>RoboLab |
| C-like | BrickCC, NXC (Not eXactly C)<br>RobotC |
| Java | leJOS NXJ |

Comparison:

`http://www.teamhassenplug.org/NXT/NXTSoftware.html`

# leJOS

- `http://lejos.sourceforge.net/`
- Java based firmware replacement and programming environment
- Eclipse support
- Java API for the brick (sensors, motors, ...)

# Brick Command Center

- `http://bricxcc.sourceforge.net/`
- integrated development environment
- only under Windows
- supports all Mindstorms bricks
- programing for NXT brick:
  - Next Byte Codes (NBC) – similar to assembler
  - Not eXactly C (NXC) – similar to C
- runs over standard firmware, own firmware for advanced features

# Brick Command Center: Basic Demo

Tools:

- Find Brick
- Watching the Brick
- Brick Joystick/Piano

# Not eXactly C (NXC)

- basic syntax similar to C
- variables, expressions, control flow
- using the Brick Command Center you can compile and download the code to the brick (Menu $\rightarrow$ Compile)

# NXC: Sensors

Sensors = input from environment

- setting sensor type:
  - `SetSensor(IN_1,SENSOR_TOUCH)`
  - `SetSensorTouch(IN_1)`
  - you can specify more detail (mode)
- reading sensor information:
  - basic reading: `Sensor(IN_1)`
  - ultrasonic sensor: `SensorUS(IN_1)`
  - more: `SensorNormalized, SensorBoolean, ...`

# NXC: Outputs

- motors (changes to the environment):
  - `OnFwd(OUT_AB, 70)`
  - `OnRev(OUT_A, 30)`
  - `Off(OUT_A)`
  - `RotateMotor(OUT_A,speed, angle)`
  - many more involved commands
- sounds: `PlayTone, PlayFile`
- display: `NumOut, TextOut, GraphicOut`

# NXC: Tasks, concurrency

- task = unit of concurrency
- explicitly declared, up to 255 tasks
- start = only task `main` is running
- task activation, termination: `StartTask`, `StopAllTasks`, `Precedes`, `Follows`
- access to motors, sensors — critical sections
- mutual exclusion support: data type `mutex`, operations `Acquire`, `Release`

# NXC: Time

- NXT has timer with granuality 1/1000 second
- `CurrentTick` — the current value of the timer
- `Wait` — waits for the specified time
- `SleepTime, SleepTimer`, ...

# Project: Organization

- groups of 3-4 students
- development environment: **Brick CC** or leJOS
- deadline: April 30th
- presentation during the lecture (with slides and demo)
- documentation ($\sim$ 2 pages)

# Project: Task

- there is no exact specification
- it is part of your task to make up an interesting problem
- preferably: one hardware architecture, two programs for different behaviours

# Minimal Requirements

- the robot should do something meaningful and understandable (not a sequence of random movements)
- the robot uses at least: two motors, two sensors, display or sound
- the implementation uses concurrency (at least two tasks)
- the implementation has real-time aspects (i.e., the behaviour depends on correct timing)

# Project: Suggestions

- find a ball, pick a ball, take a ball to some destination
- finding a path through a maze
- line following with navigation through sound
- hunter: tries to touch (shoot) a 'pray' (e.g., another robot or your hamster)

look for inspiration on the web (e.g. "youtube lego mindstorms")

# Teamwork: Advice

Divide team role's among team members, e.g.:

- boss
- designer
- hardware (Lego) engineer
- software engineer
- presenter, documentator

final report: specify contribution of individual members

# What Should You Do Now?

(in teams, using concurrency)

- install Lego Mindstorms and Brick Command Center
- try few experiments with the brick (viewing sensor values, running motors, ...)
- look at tutorial examples, try to compile some of them and run them
- try to write some simple code of your own
- build a simple robot according to the tutorial
- discuss the project