

<http://membres-liglab.imag.fr/donsez/cours>

Lego MindStorms

Didier Donsez

Université Joseph Fourier (Grenoble 1)

PolyTech'Grenoble LIG/ADELE

Didier.Donsez@imag.fr

Didier.Donsez@ieee.fr

Motivation

■ Histoire

- Né du projet « Programmable Brick » du MIT MediaLabs (95)
- Commercialisation du RIS (Robotics Invention System) par Lego (98)

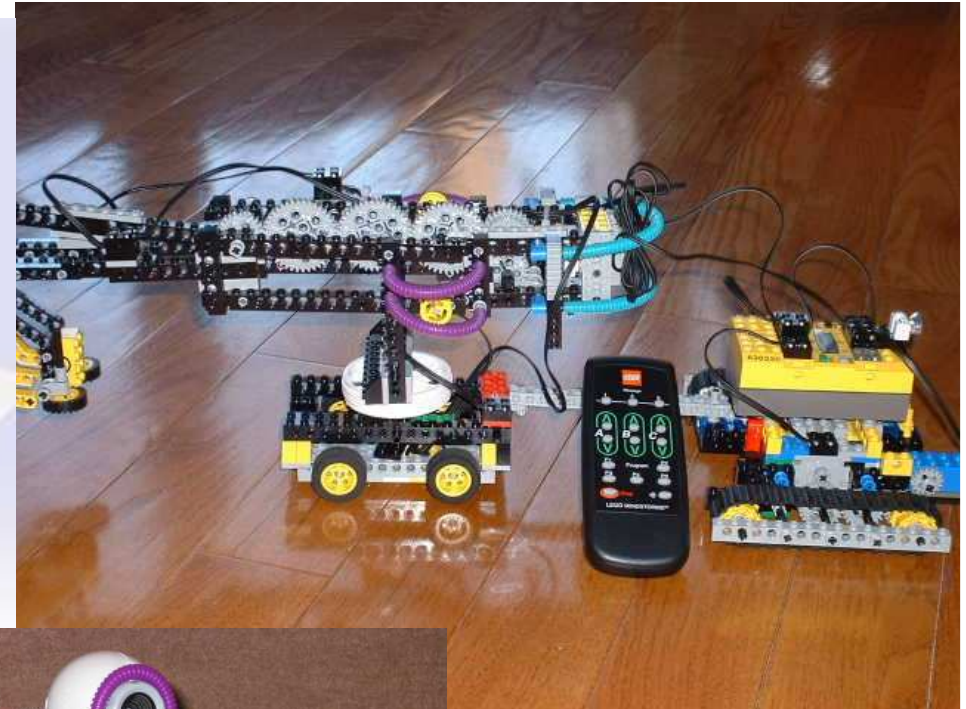
■ Motivation

- RRAD : Rapid Robotic Application Development
 - Briques Lego standard pour les parties mécaniques et la motorisation
 - Contrôleur RCX + moteur + senseurs



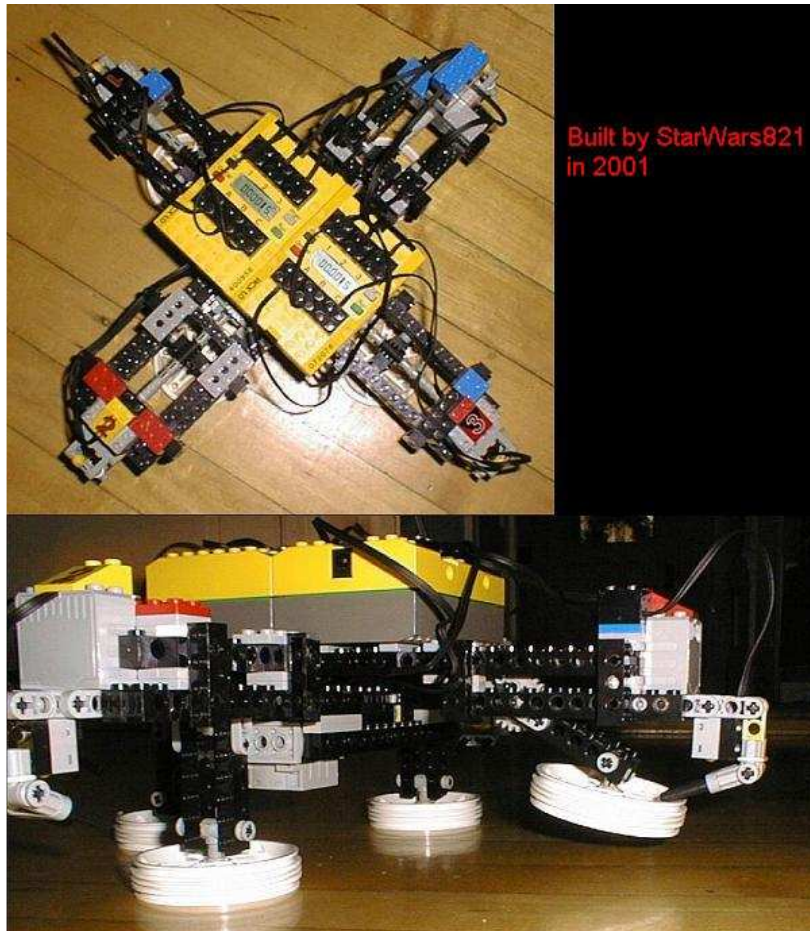
Exemples de Réalisation

<http://www.legomindstorms.com>



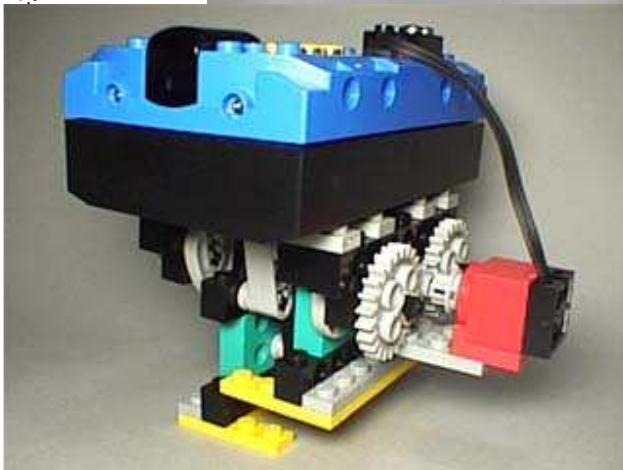
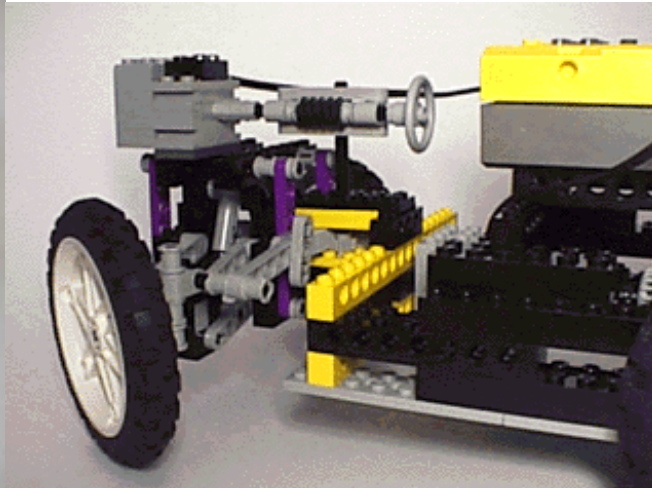
Exemples de Réalisation

<http://www.legomindstorms.com>



Exemples de Réalisation

<http://member.nifty.ne.jp/mindstorms/index.html>



2009

RCX (Robot Control)

■ Matériel

- Détails dans <http://www.crynwr.com/lego-robotics/>
- μ C Hitachi 8bits H8/3292 16 MHz
 - <http://semiconductor.hitachi.com/products/pdf/h3ltp001d1.pdf>
 - <http://semiconductor.hitachi.com/products/pdf/h33th014d2.pdf>
- 6 Ko ROM, 32Ko RAM
- 3 ports entrées
 - température
 - vitesse rotation moteur
 - détection de choc
- 3 ports sortie
 - Moteurs
- Transceiver IrDA (sert en autre au chargement du firmware)
- Clavier 4 touches
- Affichage LCD 5 caractères

RCX (Robot Control)

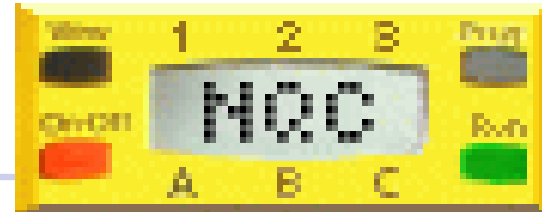
- Logiciel
 - En ROM (6Ko)
 - Fonctions bas niveau de pilotage des périphériques
 - Fonction de chargement du « firmware » par le transceiver IrDA
 - i.e. OS + Application
 - En RAM (32Ko)
 - Système d'exploitation (i.e. schedule de taches, event, drivers, ...)
 - Application (i.e Comportement robotique)

Langages de développement

Under Construction
En Construction

- Lego RCX code + Lego Dacta RoboLab
- NQC (Not Quite C)
- pbForth
- Langages synchrones
 - ESTEREL/LUSTRE
 - <http://www.emn.fr/x-info/lego//>

NQC (Not Quite C)



- Langage de programmation du RCX
 - Syntaxe proche du C
- Types
 - Entiers, Tableau []
- Variables
 - locales et globales
- Structures de contrôle
 - `if (condition) statement [else statement]`
 - `while (condition) [statement]`
 - `do [statement] while (condition)`
 - `repeat (expression) [statement]`
- Subroutine et Inline
 - `sub nomsub(int paramval, int& paramref){ statement }`
 - Passage des paramètres par référence ou par valeur
 - `void nominline(){ statement }`

NQC (Not Quite C)

■ Attente

- `Wait(dureeEnTicks);` // 1 tick par 1/100 sec
- `until(condition) [statement]`
 - mise en attente de la tache jusqu'à ce que la *condition* soit *true*

■ Tache

- `task nomtache() {statement}`
 - une des 10 taches concurrentes (ie. thread)
 - Au démarrage, seule la tache *main* est démarré (start)
 - Une tache peut démarrer (**start** *nomtache*) ou arrêter (**stop** *nomtache*) une autre tache

NQC (Not Quite C)

- Contrôle d'accès concurrents aux ressources
 - Ressources : moteurs, bipeur, ...
 - `SetPriority(expression)` fixe la priorité de la tâche
 - `acquire(listeressources) body_statement`
 `catch handler_statement`
 - tente d'acquies les ressources de la liste
 - exécute *body_statement* puis relance les ressources
 - *handler_statement* est exécuté si les ressources sont déjà prises ou perdues lors de l'exécution du *body_statement*
- Événements
 - Changement d'état sur un des senseurs de la tâche
 - `SetSensor(sensornum, sensorlist)`
 - `SetEvent(eventnum, sensornum, eventtype)`
 - `monitor(listeevenements) body_statement`
 `catch handler_statement`
 - exécute normalement *body_statement*
 - *handler_statement* est exécuté lors de l'arrivée d'un événement

NQC (Not Quite C)

- Les sorties

NQC (Not Quite C)

- Communication IrDA
 - SendMessage(expression) : envoie un octet
 - Message() : dernier octet reçu
 - ClearMessage() : efface le dernier octet reçu

pbForth

<http://www.hempeldesigngroup.com/lego/pbFORTH>

- Interpréteur Forth (en remplacement du firmware)
 - Compilé en NCQ
- Chargement dynamique/interactif du programme

- <http://www.lugnet.com/news/display.cgi?lugnet.robotics.rcx.pbforth>

OS et environnements alternatifs pour RCX

- LegOS
- LejOS
- brickOS
- Think/RCX
- RCX Java Operating Systems

LegOS

<http://www.noga.de/legOS/>

- OS très complet
 - Preemptive multitasking
 - Sémaphore et IPC
 - Dynamic memory management
 - Drivers for all RCX subsystems
 - Dynamic loading of programs and modules
 - Jusqu'à 8 programmes/modules relogeables
 - Full IR packet networking
 - Empreinte 26 Ko (soit 6Ko pour l'application)
- Langage
 - C (cross compiler GCC)
- Emulateurs
 - emulegOS, LegoSim

LejOS

<http://lejos.sourceforge.net>

- Motivation
 - Programmer les Robots (RCX) en Java
- JVM (TinyVM) pour le RCX
 - Limitations
 - pas de GC, pas de flottant, pas de switch, pas de constants String
 - Empreinte 10Ko
 - Optimisation de la mémoire
 - 1 octet pour 1 byte, 4 octets pour un objet...
- Runtime Environnement
 - java.io, java.lang, java.net, java.util
 - javax.servlet.http (car Lejos embarque un serveur Web)
 - josx.platform.rcx, josx.rcxcomm, josx.robotics, josx.util
- Chargement
 - Les classes et l'environnement sont pré-résolues au chargement

brickOS

<http://brickos.sourceforge.net>



- alternative operating system for the Lego Mindstorms RCX Controller.
- Tools
 - C/C++ development environment using gcc and g++
 - program loader (to the RCX).

LEGO .NET

<http://www.dcl.hpi.uni-potsdam.de/research/lego.NET/>

- Two approaches
 - Interpreter
 - Tiny interpreter (code-named *Micro .NET*) which interprets ECMA CLI on the H8/300.
 - Compiler
 - GCC front-end which translates ECMA CLI into H8 native machine code
- OS
 - brickOS

Think/RCX

(<http://www.objectweb.org>)

- Principe des exo kernels
 - Limiter au maximum concepts et abstractions
- Basé sur la spec. v1 de Think
 - Modèle de composants Fractal
 - Composants
 - Interfaces
 - Reliés par des composants « liaison »
 - Composants composites
 - Composants reconfigurables
 - Interface de contrôle sur des méta données (Fractal)
 - Mini-ORB, Service de liaison, ...
- Langage
 - IDL : Java
 - Implementation : C

RCX Java Operating System

<http://www.tcm.hut.fi/~pnr/rcx>

- OS/Runtime et applications
 - écrits en Java
 - compilés en natif H8 pour le chargement/exécution
 - GNU GCJ (Java Compiler) + libgcj
- JRE proche de J2ME
- Portable vers d'autres cibles
 - uClinux SIMM, ...
- Voir
 - http://www.usenix.org/publications/library/proceedings/usenix2000/fr/eenix/nikander/nikander_html/index.html

BicOS

Comparaison OS

- Application de test : PathFinder
 - Recherche de chemins avec détection d'obstacles
- Empreinte mémoire
 - Think+PathFinder = 6Ko
 - LegOS+PathFinder =28Ko
 - LejOS+PathFinder =25Ko
- Développement
 - 1 jour pour le portage du noyau (PowerPC vers RCX)
 - 1 semaine pour le comportement PathFinder

JINI sur RCX

- See <http://pandonia.canberra.edu.au/java/jini/tutorial/MindStorms.xml>

Alternatives au RCX



- Jstamp ?
 - Programmation Java
 - Profile J2ME

- Ajile ?

- TINI ?

Outils et IDE

- **Bricx Command Center (BricxCC)**
 - <http://bricxcc.sourceforge.net/>
 - IDE for programming the RCX (all versions), Scout, Cybermaster, and Spybot programmable bricks using NQC
 - **Compiler NBC and NXC**
 - See : Daniele Benedettelli, Programming LEGO NXT Robots using NXC,
http://bricxcc.sourceforge.net/nbc/nxcdoc/NXC_tutorial.pdf
- **Eclipse Plugins**
 - **leJOS Plugin for Eclipse**
 - <http://www2.info.ucl.ac.be/people/chp/projects/javarcx/eclipse/>

Lego Mindstorms NTX

- Nouvelle édition du Lego MindStorms

- Matériel

- ARM7 48 MHz, 256 KB Flash RAM, 64 KB RAM
- Atmel AVR 4 MHz, 4 KB FlashRAM, 512 bytes RAM
- 60x100 pixel LCD matrix , Bluetooth, 1 port USB 2.0, 4 input ports, 3 output ports
- Capteurs améliorés

- Firmware

- RIS 2.0
- Libéré en open source
 - <http://mindstorms.lego.com/Overview/>

- Autres

- leJOS / NXJ (tiny Java VM)
- BricxCC with NBC/NXC(Next Byte Codes and Not
- NTXGCC
- NXT# -.NET library for the Lego NXT
 - host-side DLL to control the NXT thru a BT connection <http://nxtsharp.fokke.net/>

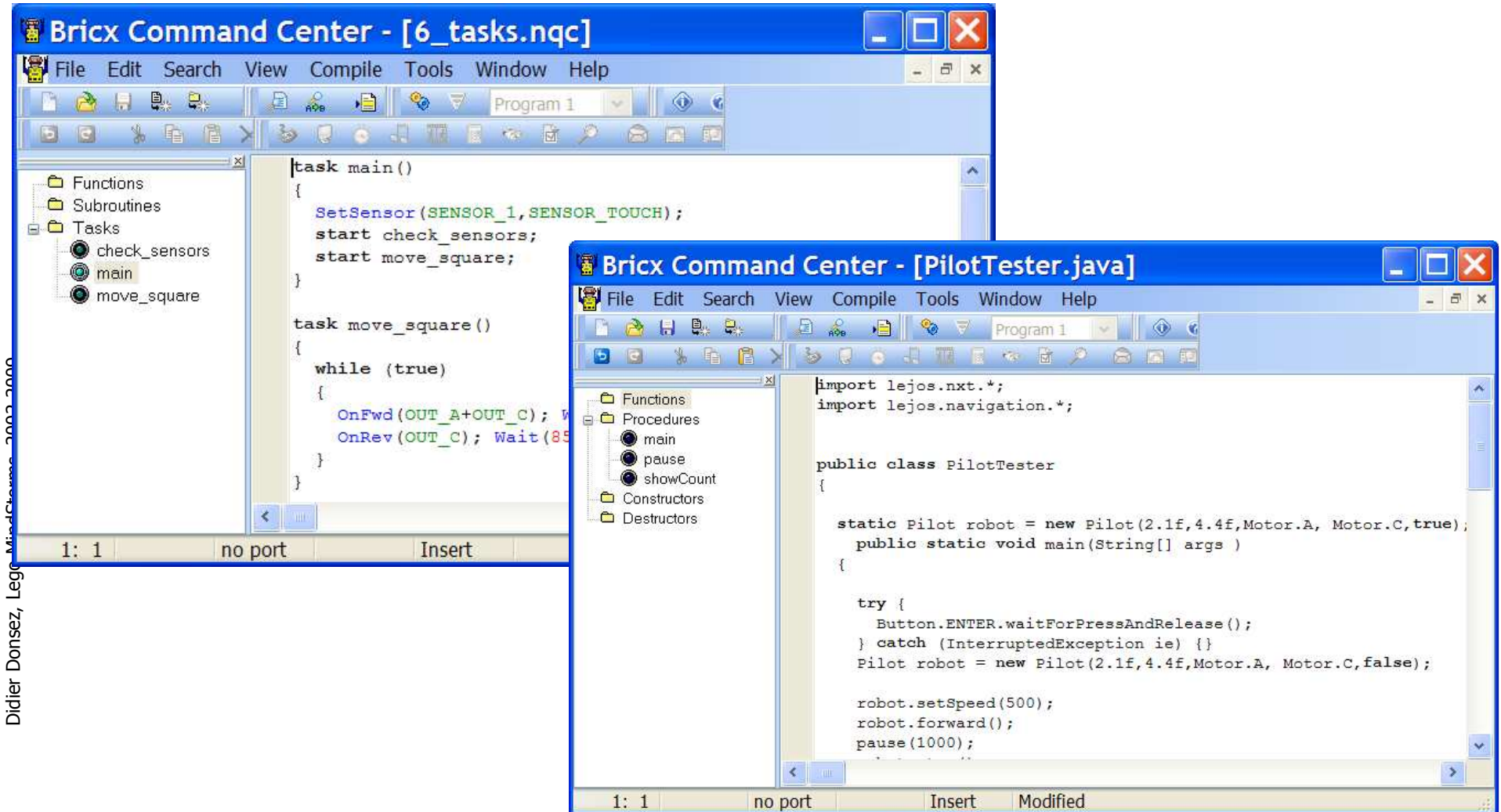
- Voir

- http://en.wikipedia.org/wiki/Lego_Mindstorms#Lego_Mindstorms_NXT



**Under Construction
En Construction**

BricX CC



NTXGCC Open Source Firmware Development Kit for NXT

<http://nxtgcc.sourceforge.net>

- **TODO**

Références RCX

- Site officiel

- <http://www.legomindstorms.com/>

- Livres

- Jonathan B. Knudsen , The Unofficial Guide to LEGO MINDSTORMS Robots, O'Reilly, 1st Edition October 1999, 1-56592-692-7, 300 pages, \$24.95
- Dave Baum, Michael Gasperi, Luis Villa, Extreme Mindstorms: An Advanced Guide to Lego Mindstorms, 250 pages (octobre 2000), Apress; ISBN : 1893115844
- Dirk Louis, Peter Müller, Programmez un robot LEGO, 330 pages (20 septembre 2002), en français.
- Baum, Lego mindstorms, (22 novembre 2000), Eyrolles; ; ISBN : 221202729X en français.

Microsoft® Robotics Developer Studio

- «Microsoft® Robotics Developer Studio 2008 (RDS) is a Windows-based environment for hobbyist, academic and commercial developers to create robotics applications for a variety of hardware platforms. RDS includes a lightweight REST-style, service-oriented runtime, a set of visual authoring and simulation tools, as well as tutorials and sample code to help get started. »
- Targetting the NXT
- C#, VPL, VB.NET

RDS – Visual Programming Language

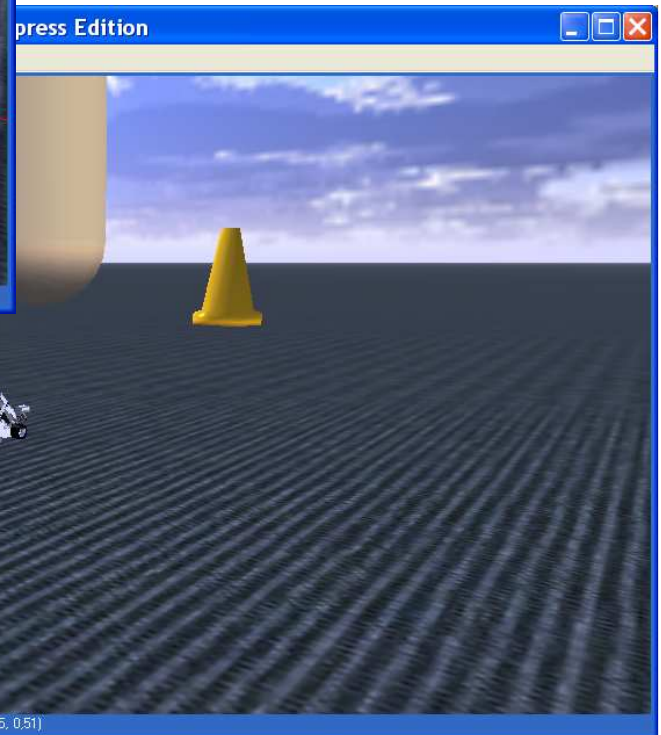
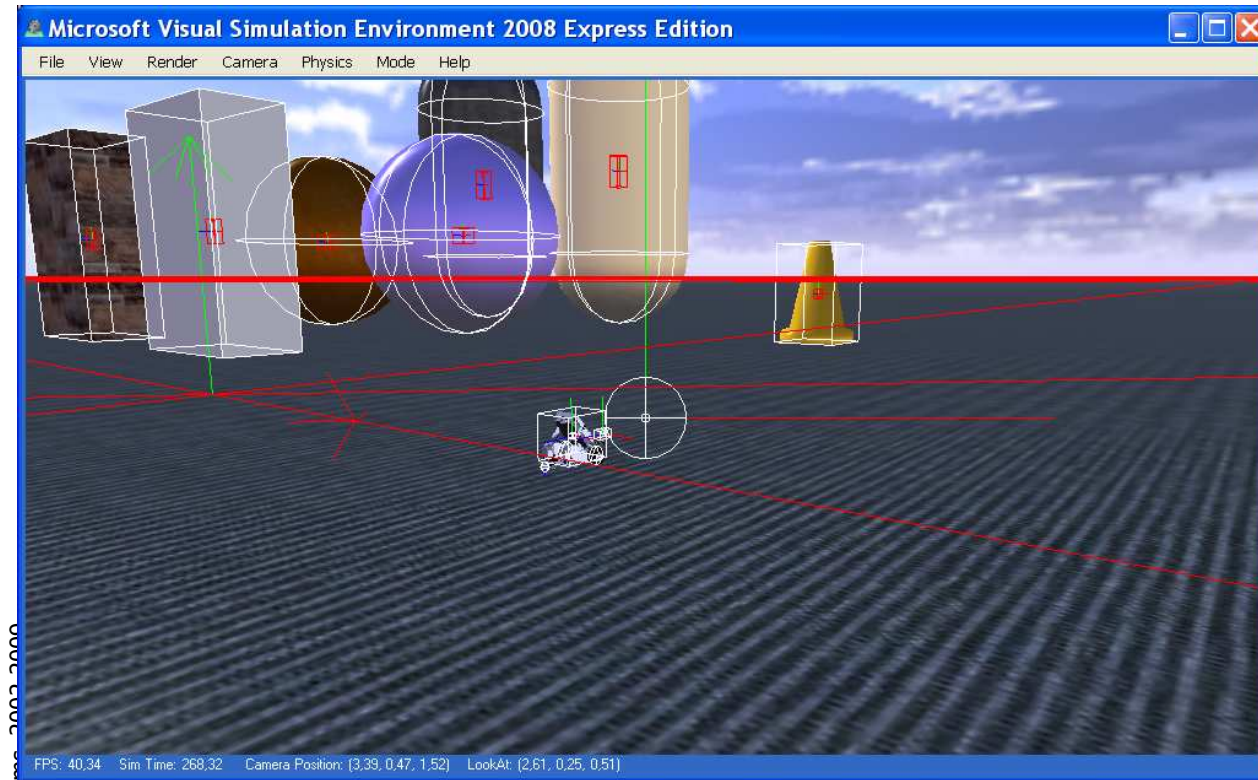
The screenshot displays the Microsoft Visual Programming Language Express Edition interface. The main workspace shows a visual programming diagram with the following components and connections:

- Data** block (int) with value 0, connected to **HiTechnicCompassSensor**.
- HiTechnicCompassSensor** connected to **Activity** (gear icon).
- Activity** connected to **LegoNXTMotorv2**.
- LegoNXTMotorv2** connected to **LegoNXTLightSensorv2**.
- LegoNXTLightSensorv2** connected to **LegoNXTMotorv2**.
- LegoNXTMotorv2** connected to **LegoNXTDrivev2**.
- LegoNXTDrivev2** connected to **LegoNXTTouchSensorv2**.
- LegoNXTTouchSensorv2** connected to **LegoNXTBatteryv2**.
- LegoNXTBatteryv2** connected to **LegoNXTUltrasonicSensorv2**.
- LegoNXTUltrasonicSensorv2** connected to **LegoNXTBrickv2**.
- LegoNXTBrickv2** connected to **LegoBrickIOv2**.

The interface includes the following panels:

- Basic Activities:** Activity, Variable, Calculate, Data, Join, Merge, If, Switch, List.
- Services:** nxt, All Found, HiTechnic Acceleration, HiTechnic Compass, Lego NXT Battery (v2), Lego NXT Brick (v2), Lego NXT Buttons (v2), Lego NXT Contact Set, Lego NXT Drive (v2), Lego NXT Light Sens, Lego NXT Motor (v2), Lego NXT Sound Ser, Lego NXT Touch Ser, Lego NXT Ultrasonic, Lego NXT Brick I/O (v2), MindSensors Accelerat, MindSensors Compa.
- Project:** Diagrams, Diagrams.xml.comp.
- Properties:** Diagrams, CompileSettings, NamespacePrefix: Robotics, ContractPrefix: http://schemas.ten, UseSpecificContra, ContractMonth: 12, ContractYear: 2009, KeyLocation: C:\Documents a, SourceLocation, IncludeDebugInfo: checked, ExcludeMainDiagr, TargetFramework: .NET CF v2.0, VisualStudioVersion: Visual Studio 20, PortSettings, HttpPort: 50000, TcpPort: 50001, RemoteDeploySettings, PackageDeployer: 55555.

RDS – Simulator (Tribot)



Références RCX

- Ressources
 - <http://www.oreilly.com/catalog/Imstorms/resources/>
- L'intérieur du RCX
 - <http://www.crynwr.com/lego-robotics/>
- Des senseurs maison
 - <http://www.plazaeearth.com/usr/gasper/lego.htm#otherhomebrew>
 - <http://ex.stormyprods.com/lego/>
- Des programmes
 - <http://www.astolfo.com/home.asp>

Références NXT

- Tutorial

- http://www.ortop.org/NXT_Tutorial/index.html

- Livres

- Bagnall, Brian. Maximum LEGO® NXT: Building Robots with Java Brains Variant Press. 2007. ISBN 0-9738649-1-5
 - <http://www.variantpress.com/books/maximum-lego-nxt-building-robots-with-java-brains>