Robotics with LEGO™

Adrian Dabrowski
dabrowski@spengergasse.at

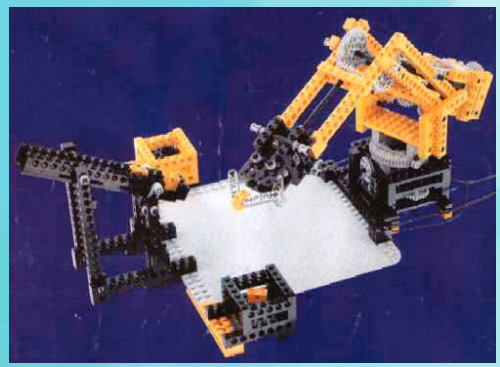
Robotic Summer School, Centrobot
V1.1 STU, Bratislava, June 30th 2010

Outline

- History
- NXT Brick
- NXT-G
- NXC, Brixcc
- LeJOS
- Constructing Hints

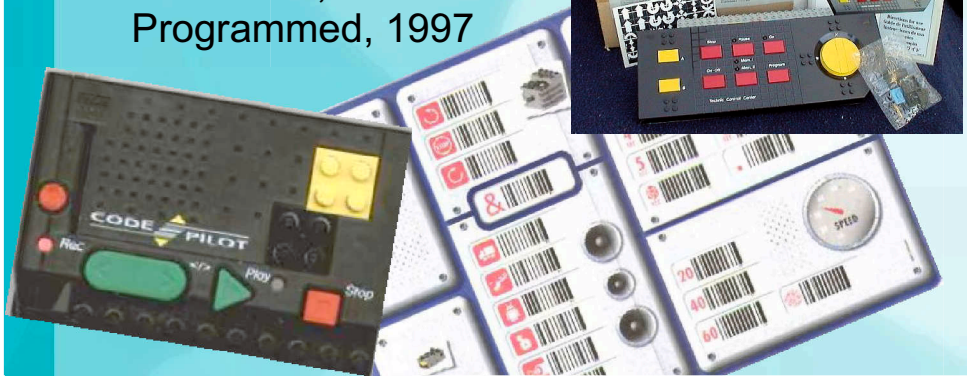
LEGO Technic

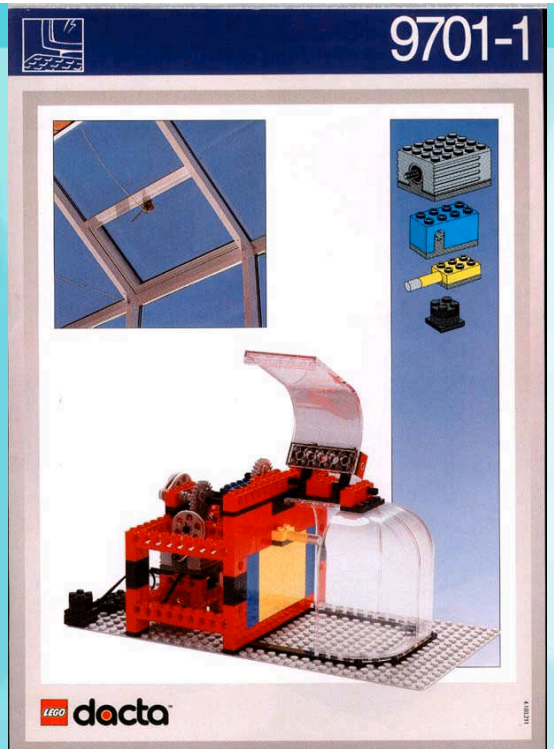
- Technic Control Center
- 1994
- Record Sequences



History – programming LEGO

- LEGO Technic Computer Control, 1986
- 90es: Lego Dacta; Physics Education
- Code Pilot, Barcode Programmed, 1997

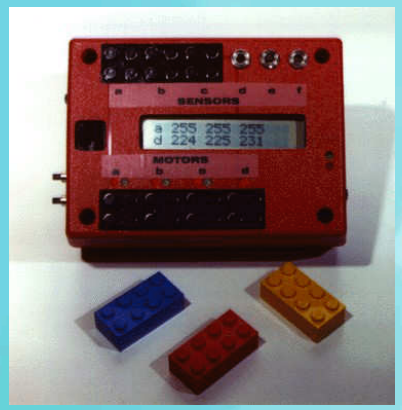




History

- Mindstorms RCX (1998) based on 8-bit Microcontroller, 32KB RAM, IR serial Communication, iconic programming language
  - with parallel events: used in many competitions (First LEGO League, RoboCup, LineFollowing, Micromouse, etc.)
  - Many alternative programming environments;

- Origin: MIT Programmable Brick, 1996



- LEGO Mindstorms RCX, 1998

The LEGO NXT "Brick" (2006,2009)

- 32 bit ARM CPU, 48 Mhz, 256kb flash, 64kb RAM, USB, Bluetooth
- 100x64 LCD, 3 Motor Ports, 4 Sensor Ports
  - Analogue and I2C
- Graphical Programming Language NXT-G (LabView based)
- 3rd party firmware (C, JAVA, …)
  - NBC Assembly
  - NXC C-Compiler, Robot C
  - LeJOS (JAVA Micromachine)

NXT Features

- Compatible to RCX sensors and motors via adapter cable
- Bluetooth radio communication
- I2C bus/sensors
- Programmable over USB (and BT)
- Rechargeable Li-ion battery available
- Public full documentation of specifications

Port A,B,C: Motors

Port 1,2,3,4: Sensors

NXT Motors

- Powerful motor with build in rotation sensor
- Control by
  - **Power** or **Speed** or **Angle**
- Its a Servo and a DC Motor at once!

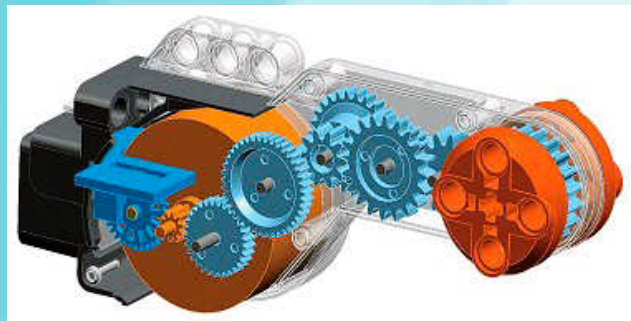
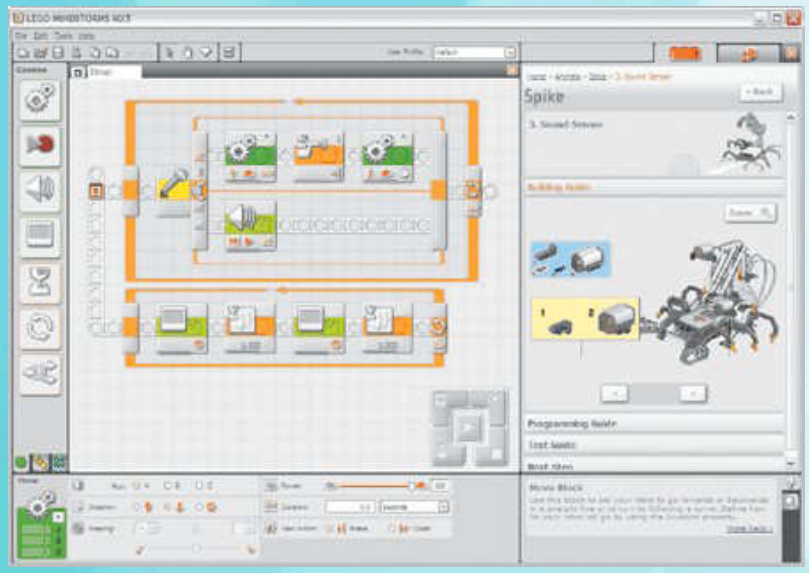


Illustration of the motor's inner workings

Advanced Sensors

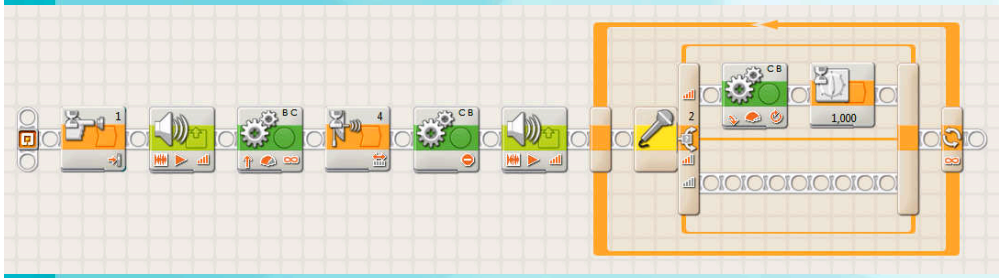
- Accelerometer
- Gyroscope
- Compass
- Line Sensor
- Camera
- Distance
- RFID
- Generic I2C
- GPS (via BT)

NXT-G Iconic Language



NXT-G Demo

- Robot is started with a „kick“ (pressing the touch sensor), then it says „good morning“, moves until it comes closer to a wall (or a person), says hello, and sits. When the hands clap, robot turns at the spot.



Source: “Introduction to LEGO Mindstorms NXT Workshop” by Pavel Petrovič RSS2009

NBC / NXC

- NBC (Next Byte Codes)
  - Assembly like language
- NXC (Not eXactly C)
  - C Compiler build on top of NBC
- BricxCC: GUI for multiple RCX/NXT languages
- Download:
  - http://bricxcc.sourceforge.net/nbc/

NBC - Overview

```
add         sub         neg         mul
div         mod         and         or
xor         not         cmp         tst
index       replace     arrsize     arrbuild
arrsubset   arrinit     mov         set
flatten     unflatten   numtostr    strtonum
strcat      strsubset   strtoarr    arrtostr
jmp         brcmp       brtst       syscall
stop        exit        exitto      acquire
release     subcall     subret      setin
setout      getin       getout      wait
gettick     thread      endt        subroutine
follows     precedes    segment     ends
typedef     struct      db          byte
sbyte       ubyte       dw          word
sword       uword       dd          dword
sdword      udword      long        slong
ulong       void        mutex       waitv
call        return      abs         sign
strindex    strreplace  strlen      shl
shr         sizeof      compchk     compif
compelse    compend     valueof     isconst
asl         asr         lsl         lsr
rotl        rotr        start       stopthread
priority    cmnt        fmtnum      compchktype
```

NXC: Tasks

- Organized in “tasks”
  - `task main() { … }`
  - Similar to threads
- Start multiple tasks simultaneously
  - `Precedes(task1, task2, taskn, … );`
- Stop(), Follows(task,...), ExitTo(task)
- Functions are outside of tasks

```
task main()
{
  SetSensorTouch(S1);       // configure the sensors
  SetSensorSound(S2);
  //SetSensorLight(S3);
  SetSensorLowspeed(S4);

  while (!SENSOR_1);              // wait for button

  PlayFile("Goodmorning.rso");    // say good morning

  Wait(1000);                     // wait 1 second

  OnFwd(OUT_BC, 75);              // go forward
  while (SensorUS(S4) > 35);      // until obstacle

  Float(OUT_BC);                  // stop

  PlayFile("Hello.rso");          // say hello

  // forever repeat turning on clapping
  do {
    if (SENSOR_2 > 90)            // if sound > 90 db, turn
    {
      OnFwd(OUT_B, 75);
      OnRev(OUT_C, 75);
      Wait(400);
      Float(OUT_BC);
      Wait(1000);                 // wait 1 second
    }
  } while (true);
}
```

NXC: Other Changes

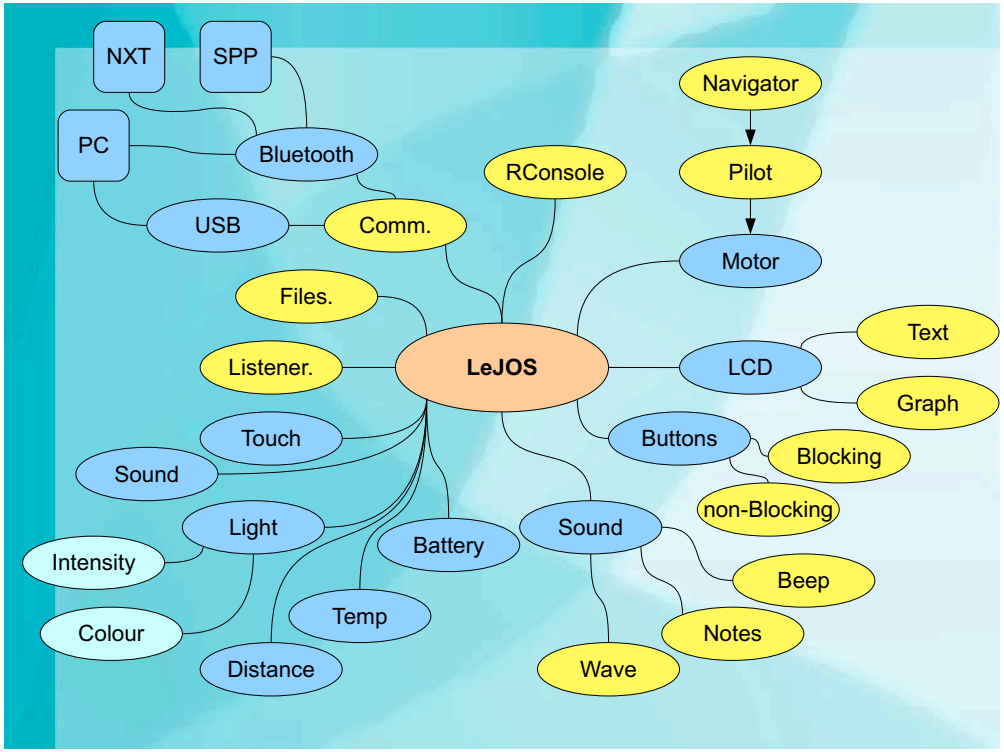
- Array handling is a bit different
- “String” has its own type
- float, but no double
- Inline assembler (NBC) via `asm {…}`
- Multidimensional Arrays only with beta firmware

LeJOS: JAVA on LEGO

- For NXT and RCX
- PC Libraries available
- Download:
  - http://lejos.sourceforge.net/
  - Tutorials and FAQs

LeJOS

- Motors
- LCD
- Buttons
- Bluetooth
- Sound (Wave)
- Sensors
  - Touch
  - Ultrasonic
  - Reflection
  - Color
  - GPS
  - Compass
  - Accelerometer
  - I2C



JAVA Micromachine

- Similar to JAVA on mobile Phones
- Preemptive threads & Synchronization
- Arrays, including multi-dimensional
- Recursion
- Exceptions
- Java types including float, long, and String
- Most of the java.lang, java.util and java.io classes
- A Well-documented Robotics API

LeJOS Advanced Robotics API

- Navigation
- Subsumption Architecture / Behavioral Programming
- Kalman Filters
- Monte Carlo Localisation

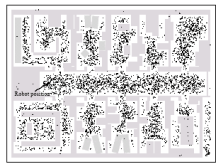


Fig. 3: Global localization: Initialization.

Fig. 4: Ambiguity due to symmetry.

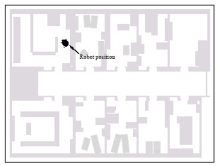


Fig. 5: Achieved localization.

LeJOS Advanced Robotics API

- Navigation
- Subsumption Architectu
  Programming
- Kalman Filt
- Mon tion

DO NOT CONFUSE IT WITH A TOY
ITS NOT A PLAYTHING ANYMORE

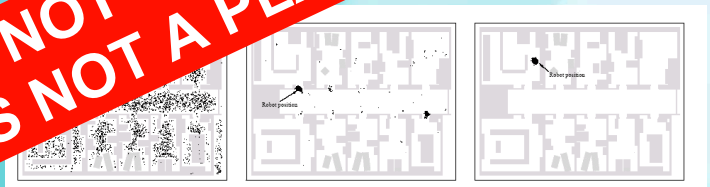
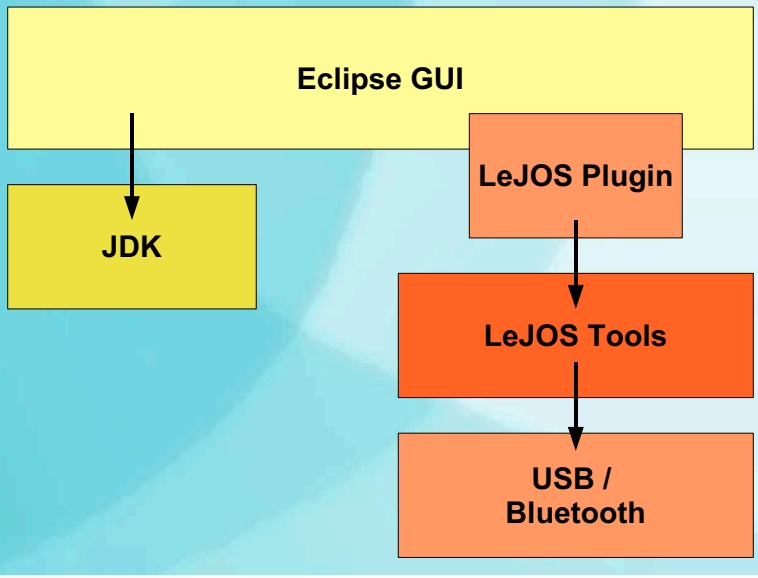


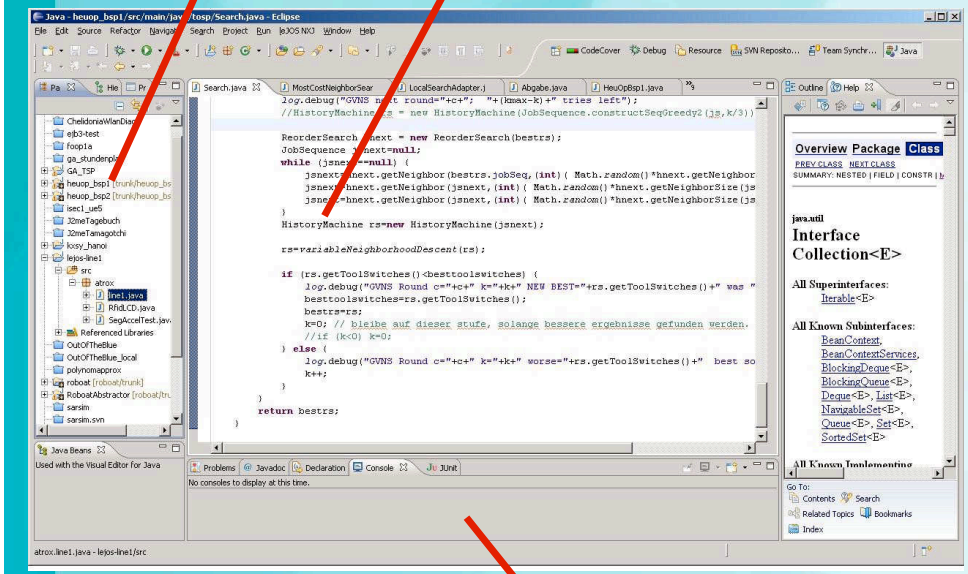
Fig. 3: Global localization: Initialization.

Fig. 4: Ambiguity due to symmetry.

Fig. 5: Achieved localization.

LeJOS Software





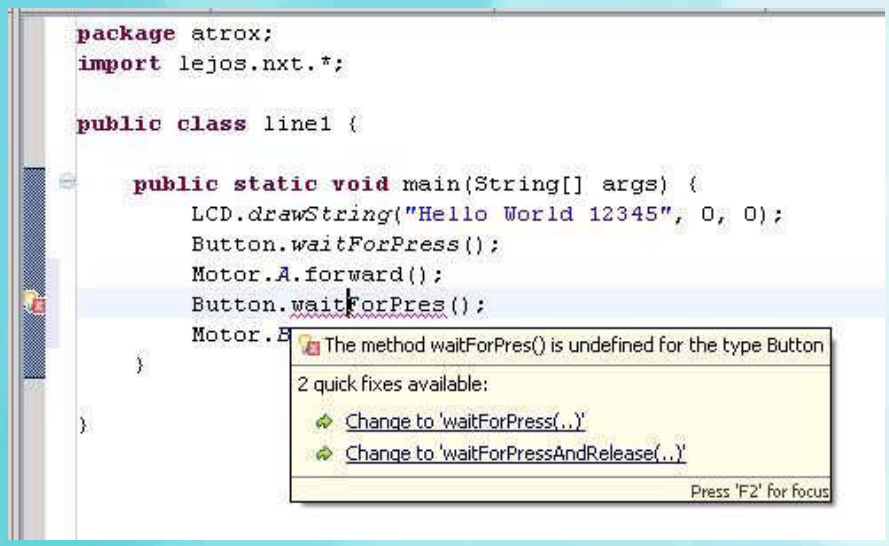
Auto-Proposals

- While Typing or by pressing CTRL-Space

```
public static void main(String[] args) {
    LCD.drawString("Hello World 12345", 0, 0);
    Button.waitForPress();
    Motor.A.
}
```

_port : TachoMotorPort - Motor
_rampUp : boolean - Motor
regulator : Motor.Regulator - Motor
backward() : void - Motor
equals(Object aOther) : boolean - Object
flt() : void - Motor
forward() : void - Motor
getBasePower() : float - Motor
getClass() : Class<?> - Object
Press 'Ctrl+Space' to show Template Proposals

Errors



JAVA in 15 Minutes

- Object Oriented
  - Static methods and classes
  - Create instances with
    `var = new Classname();`
  - Method / function Call
    `var.doSomething(param);`
    `returnValue=var.getValue();`
- Program Start at
  `main(...)`

Types

- Numeric:
  - int
  - long
  - float / double
- String
  - String
- Boolean
  - boolean
- Belong to (Scope)
  - Class
  - Method
  - Block

Block

```
{
   Statement;
   Statement;
}
```

Methods

```
public int addNumbers (int x, int y) {
    int z;
    z=x+y;
    return z;
}
```

Control Structures

```
while (expression) {
    Statement(s);
}

do {
    Statements;
} while (expression);
```

Control Structures

```
for ( initialisation ; termination ; increment ) {
    ....
}

for (int i=0; i<5; i=i+1) {
    .....
}
```

Control Structures

```
if ( expr ) {
    DoSomething();
} else {
   DoSomethingElse();
}

if (a<0) a=0;
```

Boolean (Logic) Expressions

- Equal a==b
  - a=b is an assignment (!)
- Unequal a!=b
- Not !a
- And a&&b
- Or a||b
- Example:
  - if ((a<5) || (a>10)) { … }
  - return a<5;

Numeric Operations

- Add/Subtract + -
- Multiply / Divide * /
- Modulo %
- Shortcuts:
  - a = a+1;
  - a += 1;
  - a++;

HelloWorld.java

```
import lejos.nxt.*;

public class HelloWorld {

    public static void main(String[] args) {
        System.out.println("Hello World");
        Button.waitForPress();
    }

}
```

NXT Motors

- Control by
  - **Power** or **Speed** or **Angle**

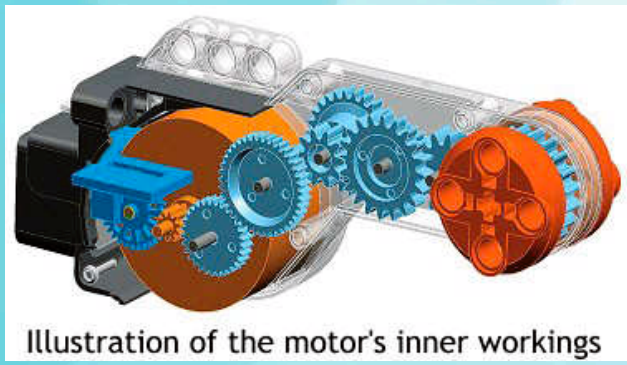


Illustration of the motor's inner workings

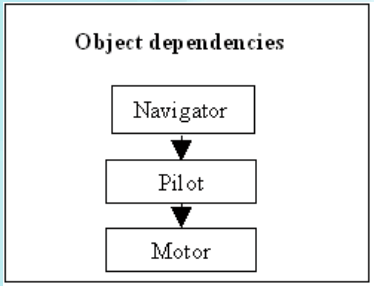
Motor.X.... (x=A..C)

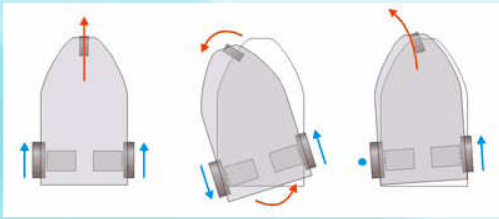
- Motor.X.forward()
- Motor.X.backward()
- Motor.X.stop()
- Motor.X.flt()
- Motor.X.setSpeed(speed)

- Motors are static objects, and do not need to be created.

Advanced Movement

- For wheeled devices (differential drive)
- **Pilot** keeps track of movement
  - Distance
  - Orientation
- **Navigator** keeps track of position





Buttons

- Blocking
  - waitForPress()
- For all Buttons
- For a specific Button
  - ENTER
  - ESCAPE
  - LEFT
  - RIGHT
- Non Blocking
  - IsPressed()

```
Button.waitForPress();

Button.ENTER.waitForPress();
while ( !Button.ENTER.isPressed() ) {}
```

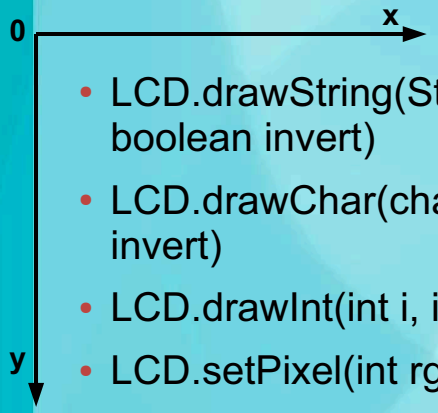
Buttons

Why is this not a good Idea?

```
public static void main(String[] args) {
    init();
    while(!Button.ENTER.isPressed()) {
        do_something();
    }
}
```

- Hint: Try starting this program from the on-screen menu.

LCD



- LCD.drawString(String str,int x, int y, boolean invert)
- LCD.drawChar(char c, int x, int y, boolean invert)
- LCD.drawInt(int i, int x, int y)
- LCD.setPixel(int rgbColor, int x, int y)

Sensors

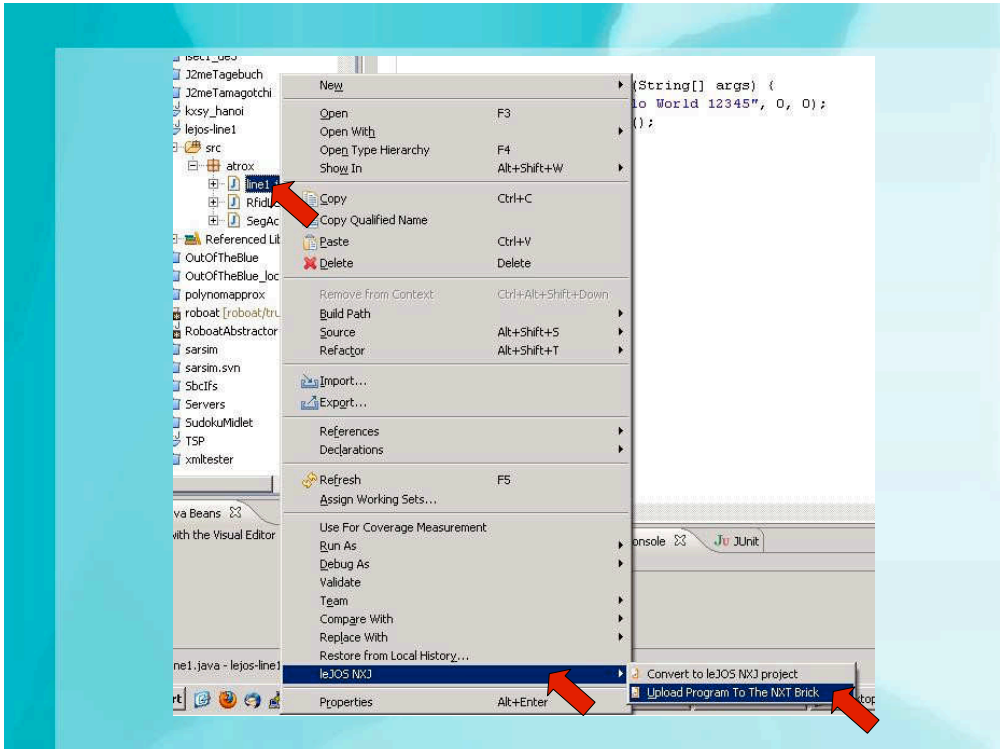
- Sensor-objects have to be created using the “new” operator

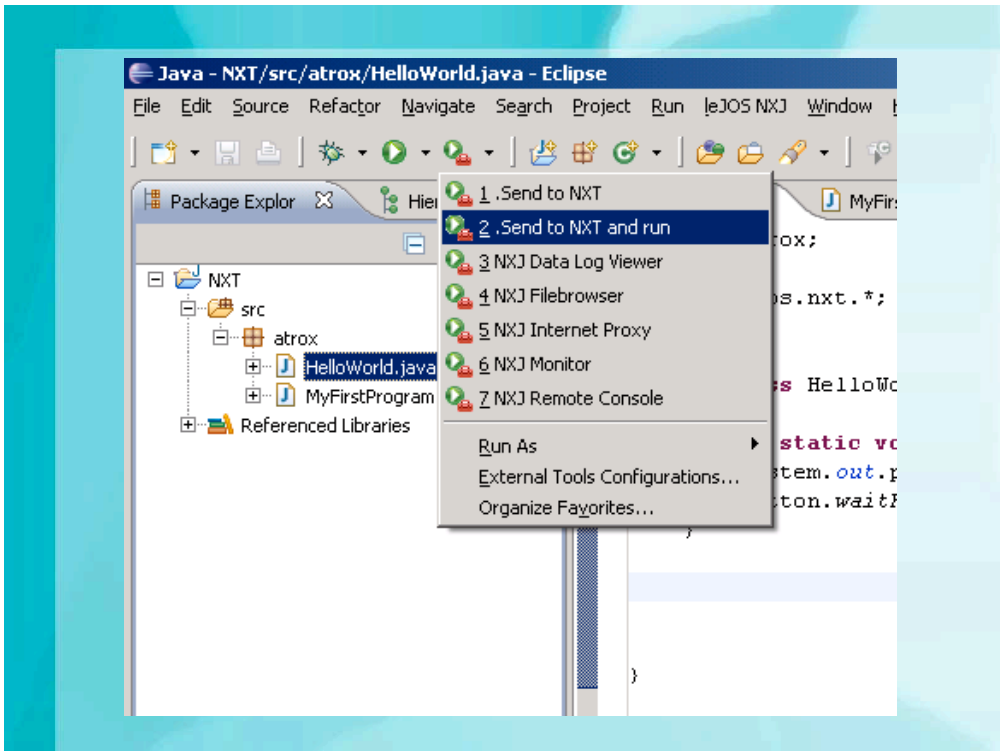
```
int var1;
LightSensor left = new LightSensor(SensorPort.S1);
LightSensor mid = new LightSensor(SensorPort.S2);
LightSensor right = new LightSensor(SensorPort.S3);

var1=left.readValue();

LCD.drawString("Left: "+var1,0,0);
```

Light Sensors

- Only light and dark
- Light, dark and Color

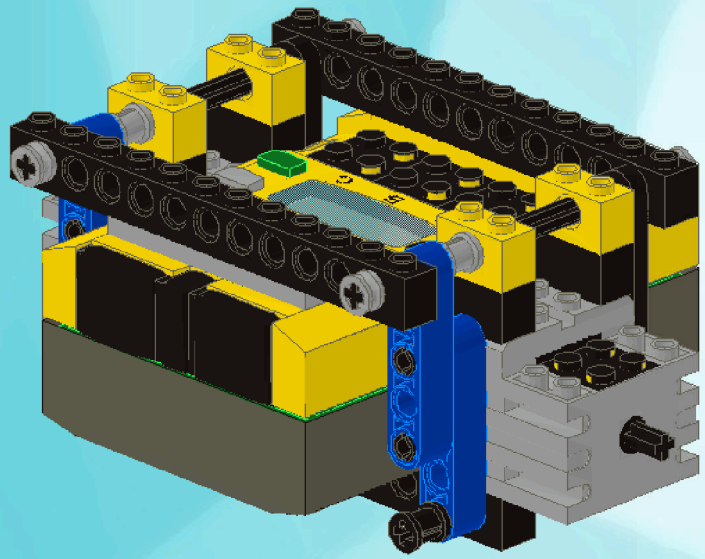
```java
import lejos.nxt.*;

public class MyFirstProgram {

    LightSensor light;

    public static void main(String[] args) {
        MyFirstProgram hw2=new MyFirstProgram();
        try {
            hw2.run();
        } catch (InterruptedException e) {}
    }

    public MyFirstProgram() {
        light=new LightSensor(SensorPort.S1);
    }

    public void run() throws InterruptedException {
        int i=0;  // some variable
        // repeat while the ENTER Button is NOT pressed
        while ( !Button.ENTER.isPressed() ) {
            if (Button.LEFT.isPressed()) { // when the left button is pressed, decrement i
                i=i-1;
            }
            if (Button.RIGHT.isPressed()) { // when the right button is pressed, increase i
                i=i+1;
            }
            LCD.clearDisplay();
            LCD.drawInt(i,1,1);
            Thread.sleep(250); // sleep 250 ms = 1/4 sec
        } // end of while
    }

}
```





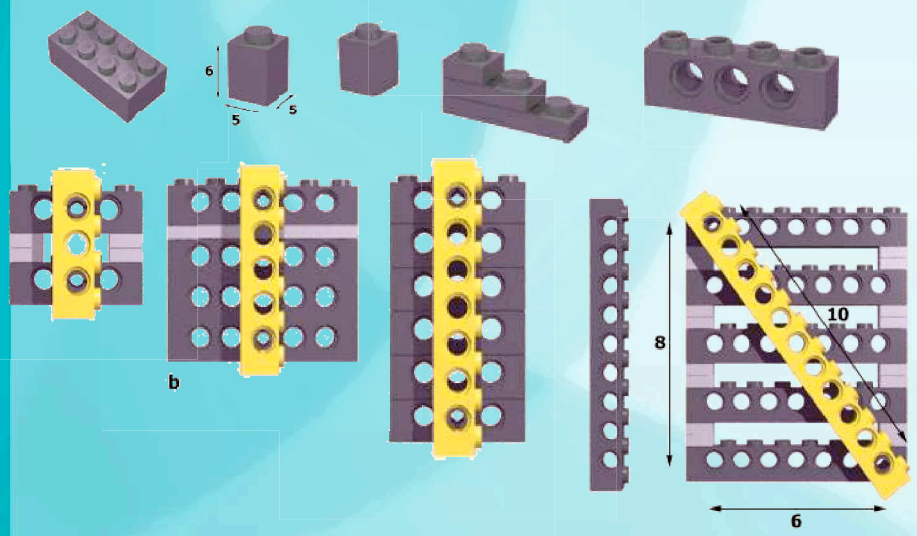
Contructing with LEGO Technic

From: **"Guide to LEGO™ Geometry"** By Gustav Jansson



From: **"Guide to LEGO™ Geometry"** By Gustav Jansson

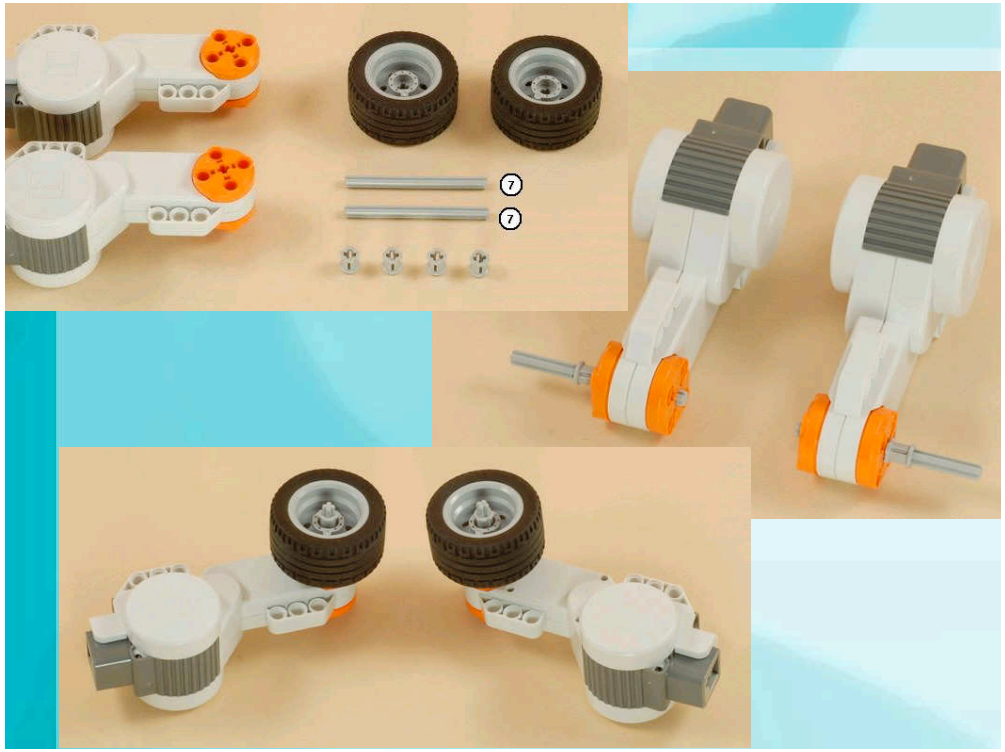
LEGO Geometry

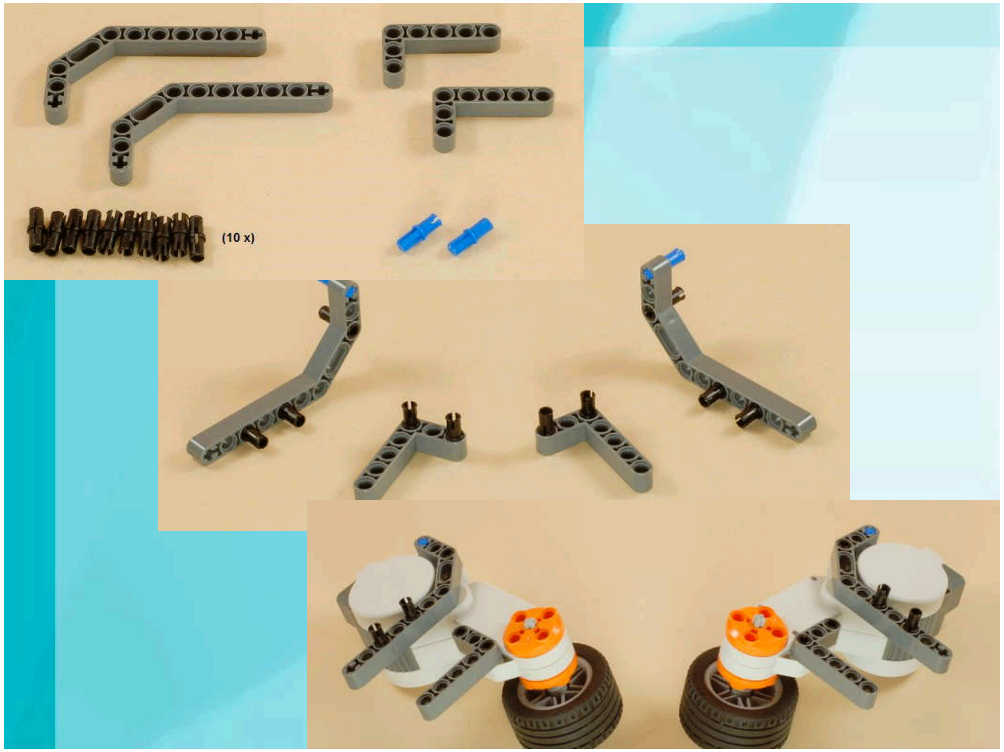


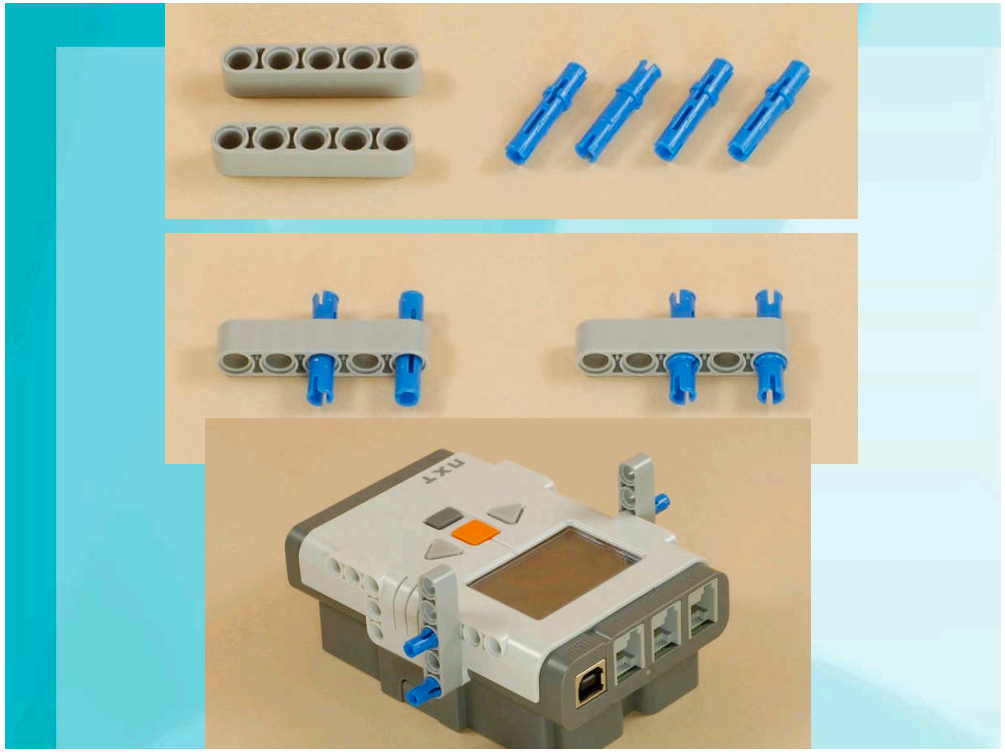
LEGO Digital Designer

5 Min Robot

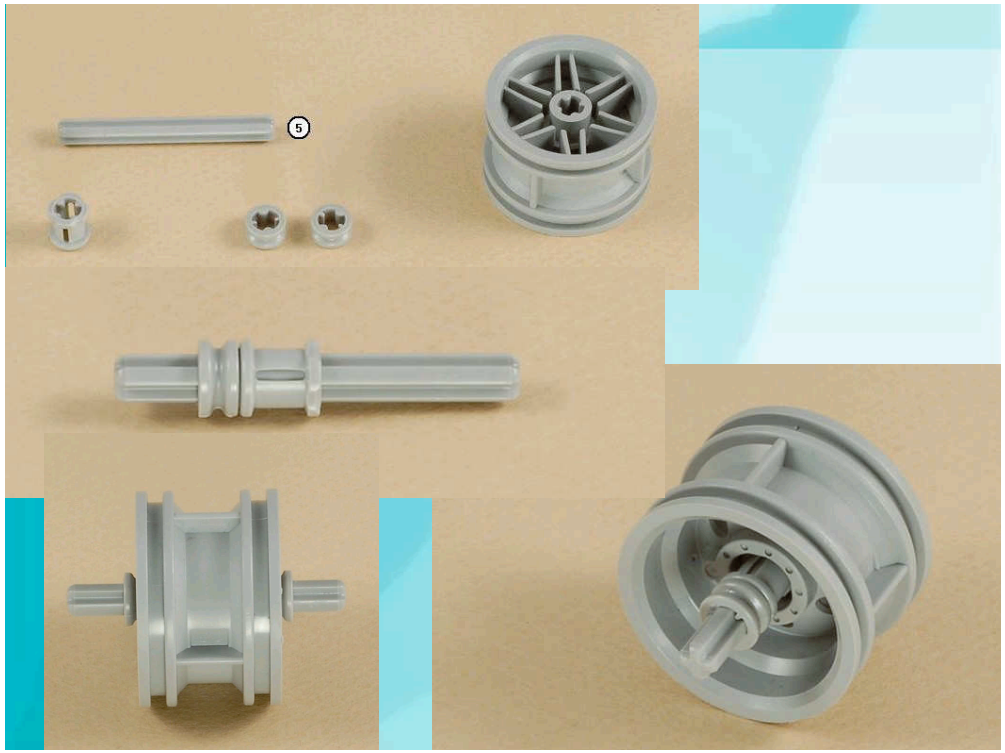
- http://www.nxtprograms.com/NXT2/five_minute_bot/index.html
- http://wiki.atrox.at/index.php/NXT-JAVA
- http://robotika.sk/nxt/

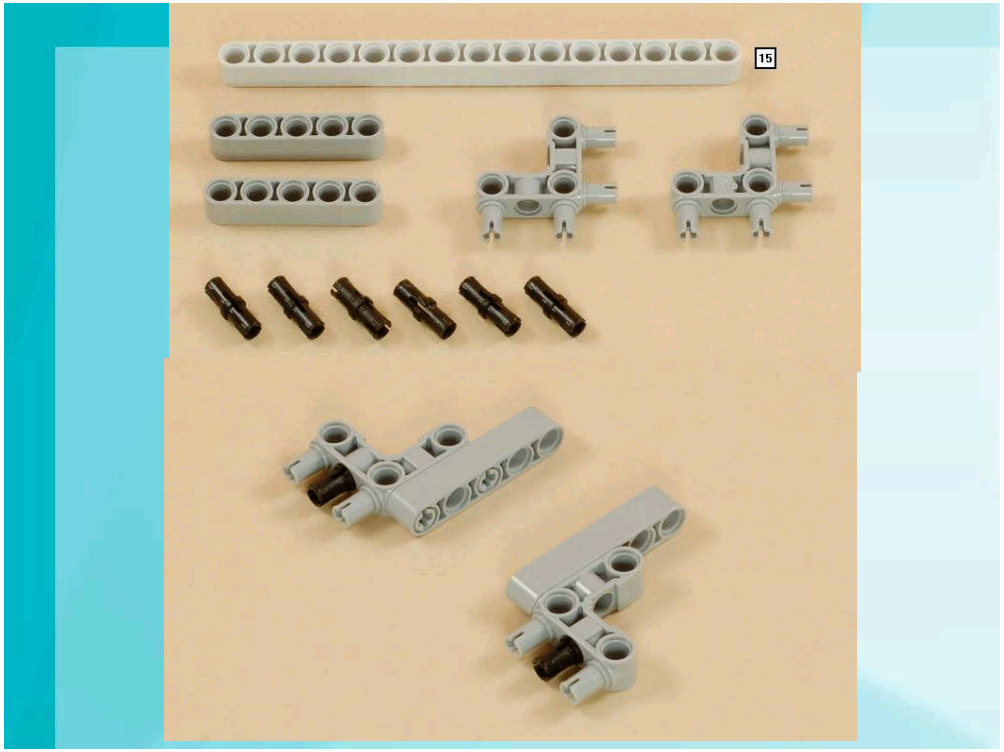


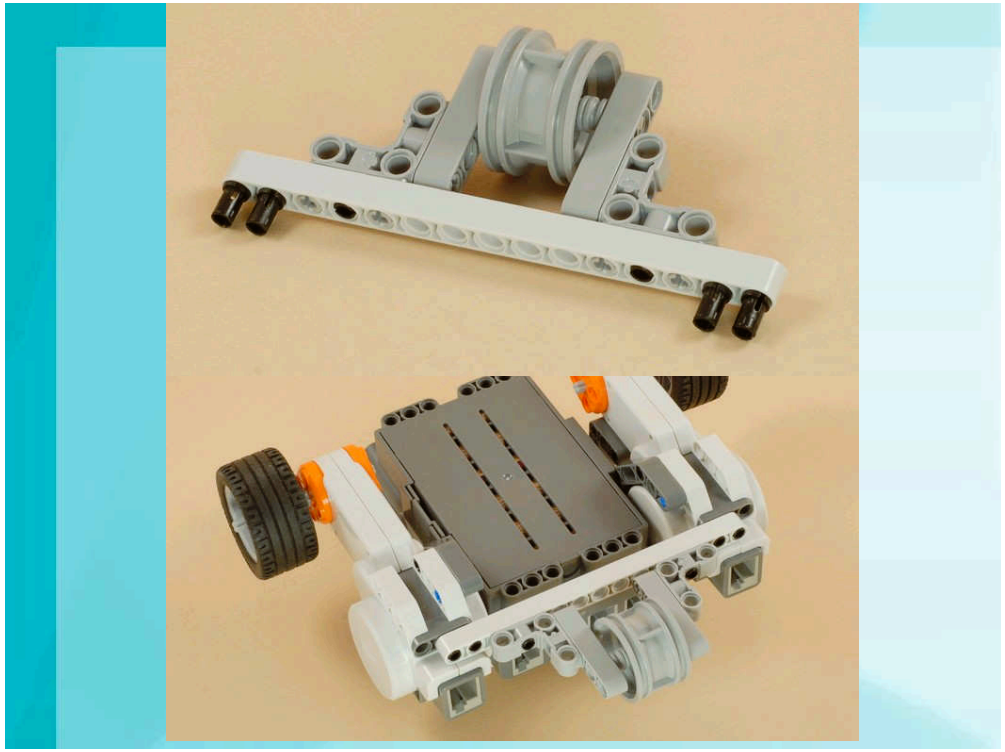


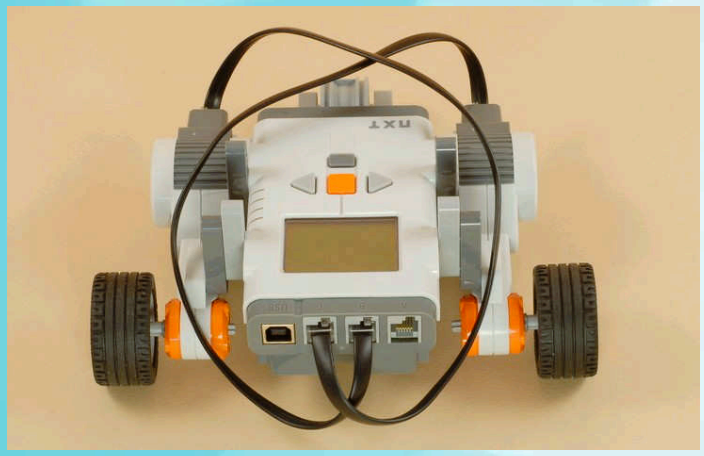


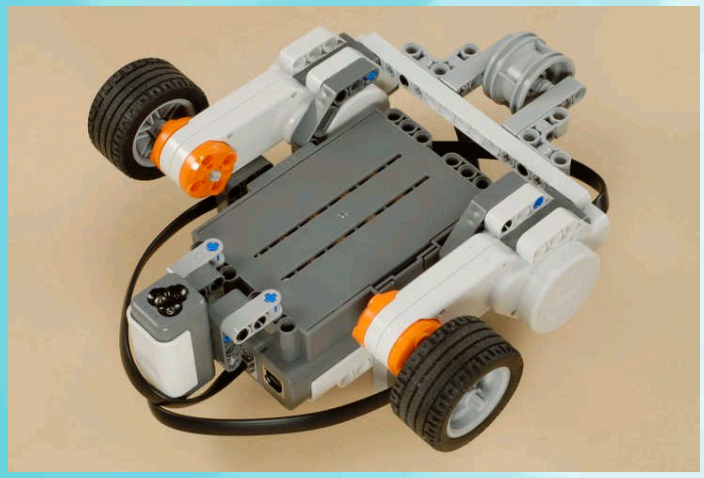

NXT

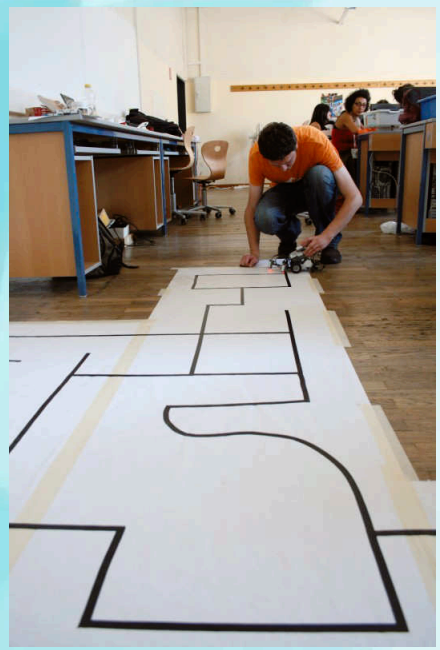

5


15









Robotics with LEGO™

Adrian Dabrowski
dabrowski@spengergasse.at

June 30th 2010