



UPPSALA
UNIVERSITET

Lab 2

Programming LEGO Mindstorms NXT

Jakaria Abdullah
19 September 2018





Lab Overview

Lab goals

- Real-time programming on an embedded device
- Problem solving using real-time tasks

Schedule

- Slot 1: 20.9.18 13-15:00 1312, 1313
- Slot 2: 26.9.18 10-12:00 1312, 1313
- Report Deadline: 01.10.18 23:59

Lab preparation

- Get your lego box (after this class)
- Check lab web page:

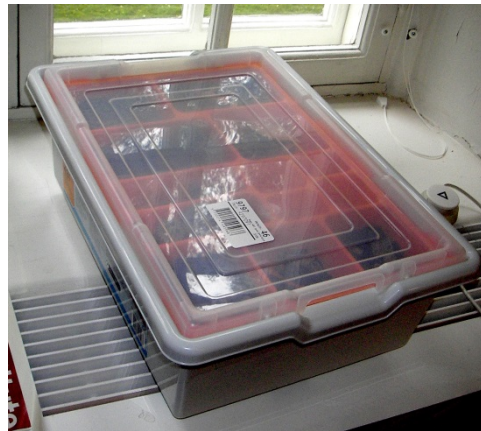
<http://www.it.uu.se/edu/course/homepage/realtid/2017-368/labs/lab2>



UPPSALA
UNIVERSITET

LEGO Box

Each group gets one LEGO Mindstorms box:



Follow the instruction in the box to rearrange before return

Deadline: 01.10.18, Monday



UPPSALA
UNIVERSITET

Lab Environment

LEGO toolchain +
Libusb +
Fwexec (program uploader)

LEGO VM (Debian 64-bit)



VirtualBox VM manager



Lab PC



Your robot



Username: lego

PW: lego

Root PW: lego



UPPSALA
UNIVERSITET

Toolchain

- Download VM from this link:
https://drive.google.com/file/d/1j7j9JH6VbRoziMcB10dE_CQmorXekQ_3/view?usp=sharing
- Import it in Virtualbox of lab computer and start
- GNAT GPL 2011 ARM for Linux (installed in VM)
<http://www.dit.upm.es/~str/proyectos/mindstorms/2011/index.html>
- Adacore GNAT GPL 2012 Windows:
<https://www.adacore.com/download/more>
- Need to install libusb driver
- Installation instruction available, non-trivial



UPPSALA
UNIVERSITET

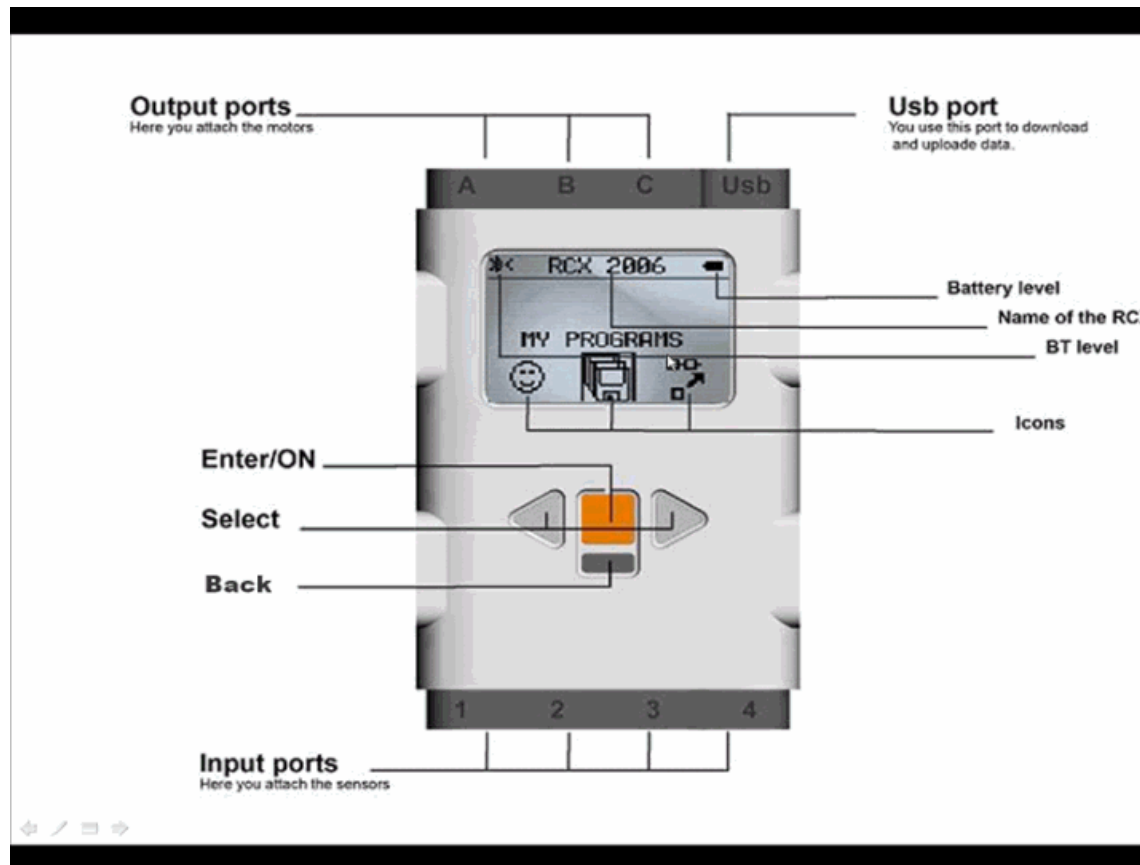
LEGO NXT

- One main unit with display, speaker, bluetooth
- 3 x motors
- Sensors:
 - Light
 - Distance (ultrasound)
 - 2 x touch
 - Sound



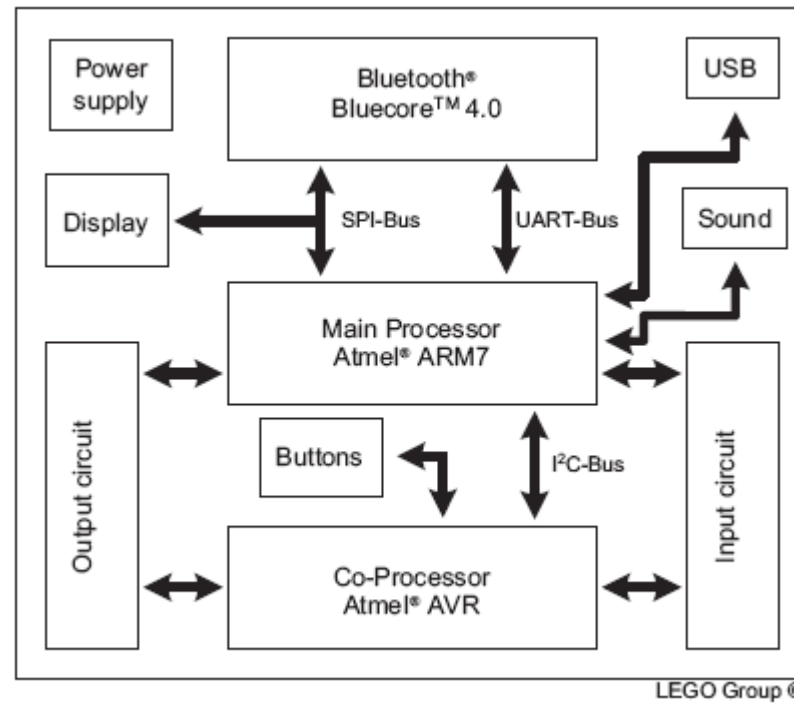


NXT Main Unit





NXT Main Unit



32-bit ARM7 main processor + 8-bit AVR co-processor,
64k RAM, 256k Flash, 48MHz clock



UPPSALA
UNIVERSITET

NXT Sensors



Light sensor:
Passive
Measures intensity



Ultrasonic sensor:
Digital sensor
Range: 255 cm
Precision: +/- 3 cm

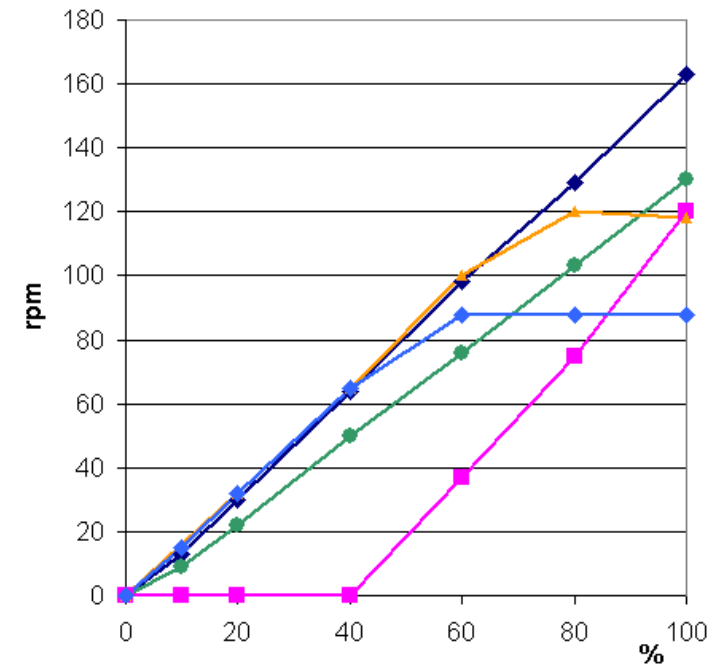
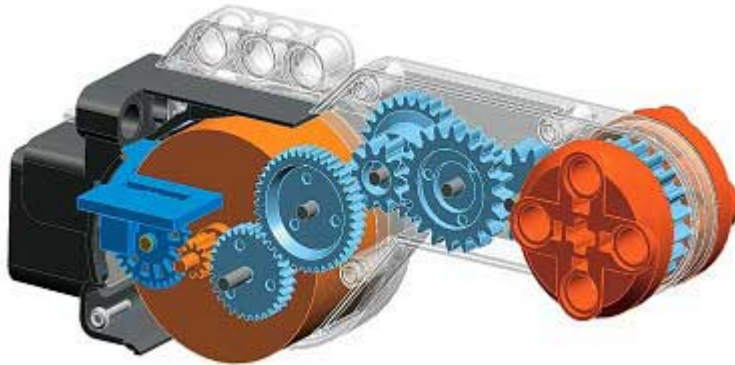


Touch sensor:
Passive
Push button



UPPSALA
UNIVERSITET

NXT Motor

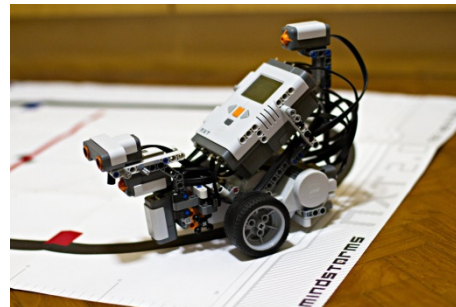


<http://www.philohome.com/nxtmotor/nxtmotor.htm>



UPPSALA
UNIVERSITET

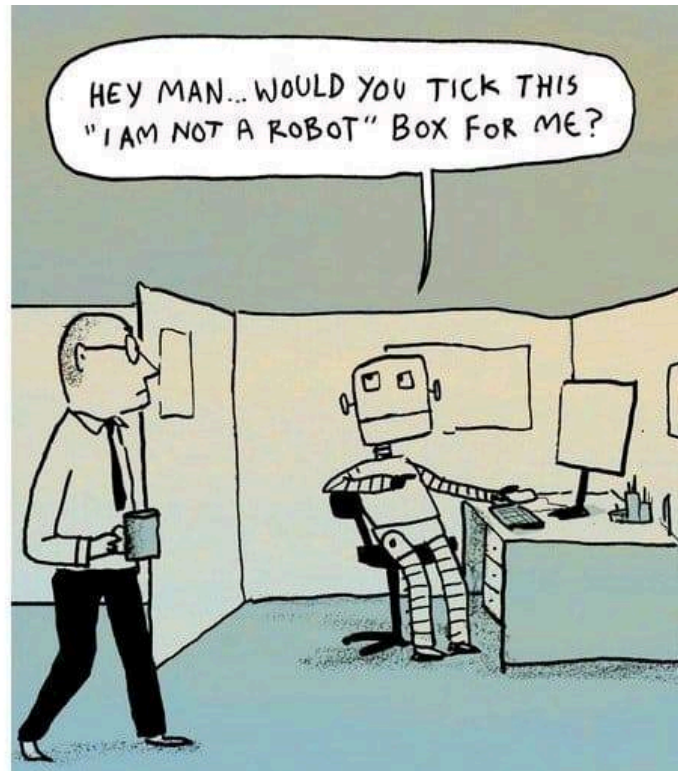
Make Your Robot!





UPPSALA
UNIVERSITET

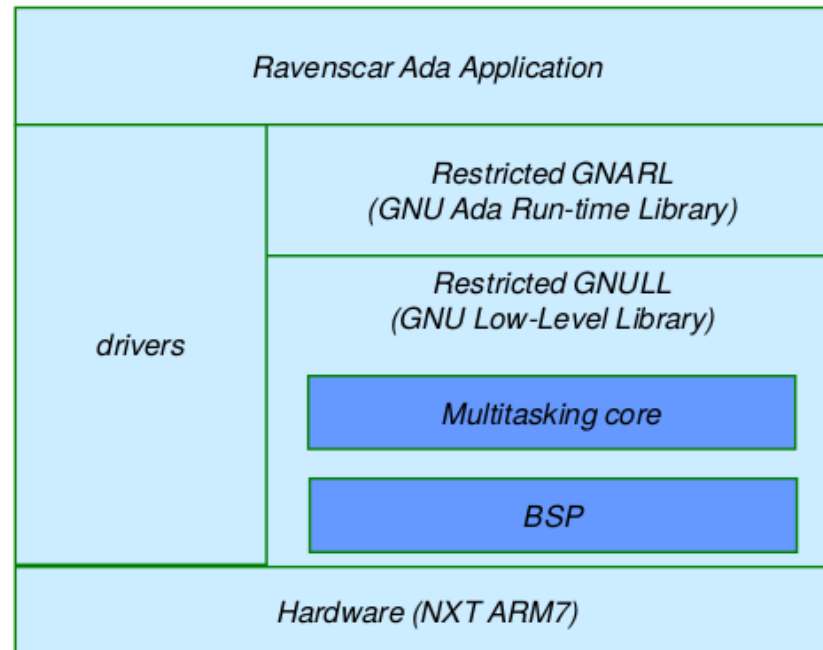
Help It To Run!



source: me.me



Ada NXT Runtime



Only 4186 lines!

Program + driver + runtime -> in RAM!

Based on Ravenscar small footprint profile (SFP)

<http://www.it.uu.se/edu/course/homepage/realtid/2017-368/lab2/runtime>



UPPSALA
UNIVERSITET

Ravenscar Profile

- Ada subset for safety-critical system
- Important features:
 - No relative delay
 - No select
 - Only one entry per protected object
 - Task declarations in library/package level
 - Only Ada.Real_Time
 - many more ...

Check:

<http://www.it.uu.se/edu/course/homepage/realtid/2017-368/lab2/ravenscar>



UPPSALA
UNIVERSITET

Helloworld Again!

Main Procedure

Helloworld.adb

Task Package Specification

Tasks.ads

Task Package Implementation

Tasks.adb



Main Procedure

```
-- File : hello_world.adb  
with System ; -- default package, always include  
with Tasks ; -- Integrate tasks package  
  
procedure hello_world is  
    pragma Priority (System.Priority'First);  
begin  
    Tasks.Background;  
end hello_world;
```

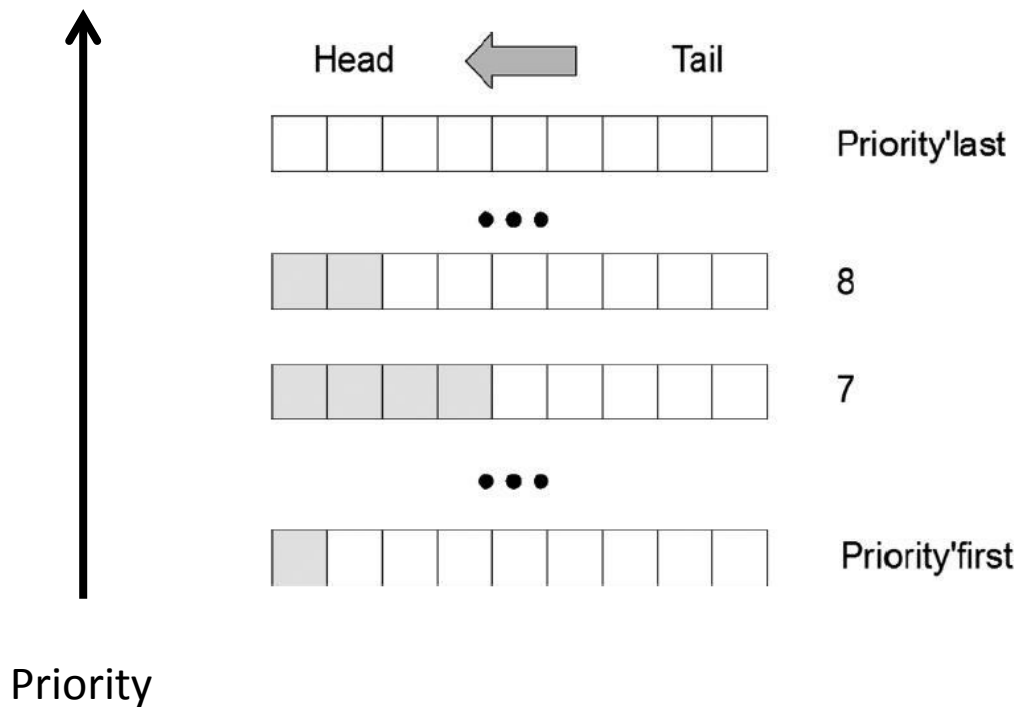
Calling main procedure of tasks package

Assigning lowest priority



Ada Priority Model

FIFO queue within same priority





Tasks Package

```
-- File : Tasks. ads
with Ada.Real_Time ; use Ada.Real_Time;
with NXT;           use NXT;
-- main NXT package

package Tasks is
  procedure Background;
  private
    -- Define periods and times
    -- Define used sensor ports
    ...
end Tasks;
```



Tasks Implementation

-- File : Tasks. adb

```
Package body Tasks is
  procedure Background is
  Begin

  end loop

  task HelloworldTask is
    -- Define priority
    pragma Storage_Size(4096);
  end Helloworldtask;

  task body HelloworldTask is

  begin

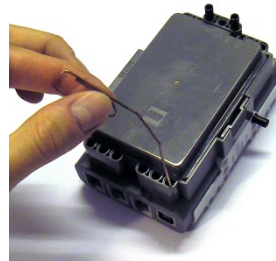
  ....
```



UPPSALA
UNIVERSITET

Compilation and Upload

- Check step by step instructions in this file:
http://www.it.uu.se/edu/course/homepage/realtid/2017-368/labs/lab2/Compile_Upload_VM.pdf
- Don't forget to charge your robot's battery!
- For first time, you need to put NXT into reset mode:



- If you stop the robot, everything will be gone



HW Interfacing

Example: Initializing sensors and motors

```
1 Touch_Sensor_Id : Touch_Sensor (Sensor_1);  
2  
3 Light_Sensor1 : Light_Sensor := Make(Sensor_2, True);  
4  
5  
6 Right_Motor_Id : constant Motor_Id := Motor_A;  
7 Left_Motor_Id  : constant Motor_Id := Motor_B;
```

Check: programs in `low_level_tests.zip`
API in `drivers.zip`



Motor Control

Example: Using motor control API

```
1 procedure Forwards is
2 begin
3     Control_Motor (Right_Motor_Id, Speed_Full, Forward);
4     Control_Motor (Left_Motor_Id, Speed_Full, Forward);
5 end Forwards;
6
7 procedure Turn_Left is
8 begin
9     Control_Motor (Right_Motor_Id, Speed_Full, Backward);
10    Control_Motor (Left_Motor_Id, Speed_Half, Backward);
11 end Turn_Left;
```

Check: motor_test.zip examples
drivers.zip for API descriptions



Synchronization

- Synchronization between tasks by protected objects
- Ravenscar restriction: **only one entry** per protected object
- Protected object itself should have **priority at least as high** as the maximum priority of the user tasks
- Access to this protected object is controlled by Immediate Ceiling Locking Protocol (ICPP)



UPPSALA
UNIVERSITET

Lab Assignments

- **Part 1: Warm-up**
Extend helloworld to display light values
- **Part 2: Event-driven scheduling**
Detect a sensor event
- **Part 3: Periodic task scheduling**
Add obstacle avoidance
- **Part 4: Line tracker**
Complete lap following another robot



UPPSALA
UNIVERSITET

Part 4



Follow **it** on the **this** track