

# Integrated Modeling, Control Synthesis and Code Generation with Modelica and Dymola

Johan Åkesson

Department of Automatic Control, Lund University

Hilding Elmqvist

Ulf Nordström

Dynasim, Lund



# Project in Automatic Control

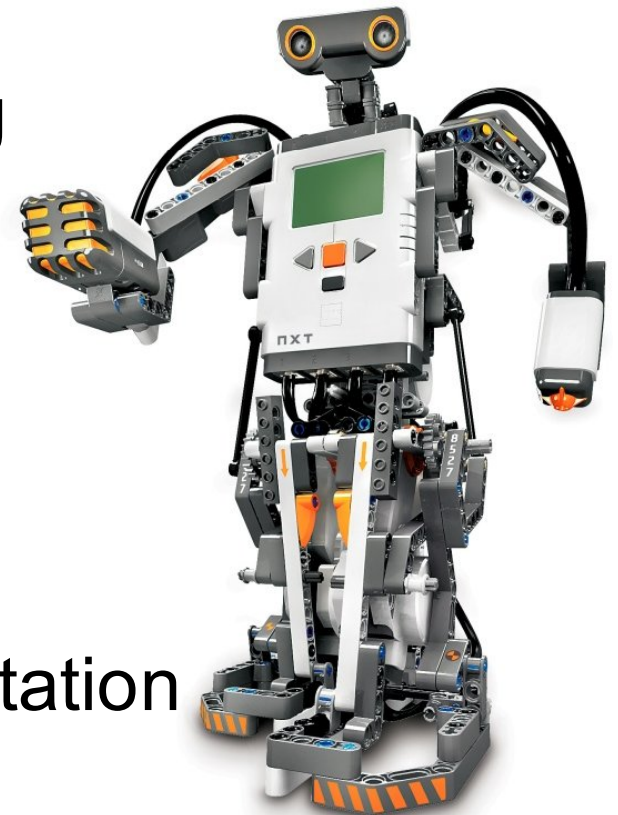
## FRT090

Johan Åkesson  
Department of Automatic Control  
Faculty of Engineering  
Lund University



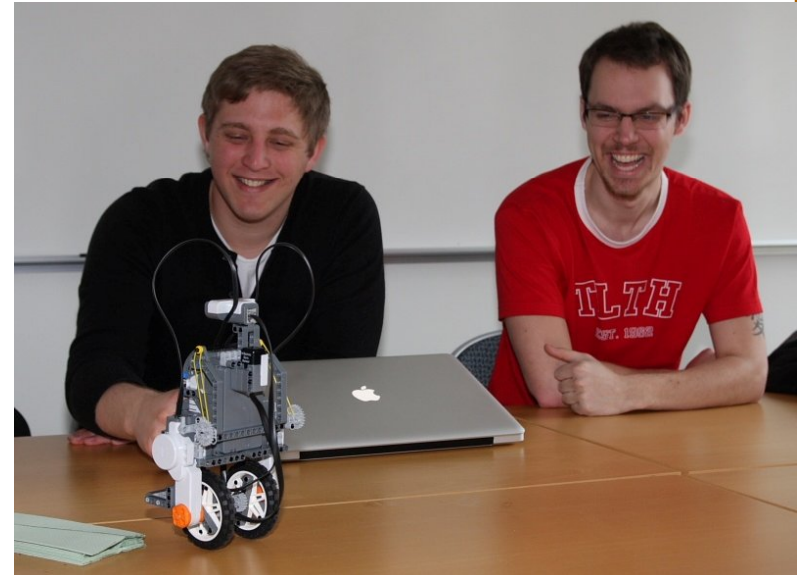
# Project in Automatic Control

- Advanced level course 7.5 ECTS units
- Course runs for seven weeks
- Team effort
  - Collaborative problem solving
- Get practical experience
  - Work in the lab
- Apply course knowledge
  - Modeling and identification
  - Control design and implementation



# Project in Automatic Control 2009

- 25 students
  - Mostly from Lund but some exchange students
  - Several disciplines
    - Engineering physics
    - Applied mathematics
    - Computer sciences
    - Chemical engineering
- Two groups working with Lego/Dymola
  - Five students in each group



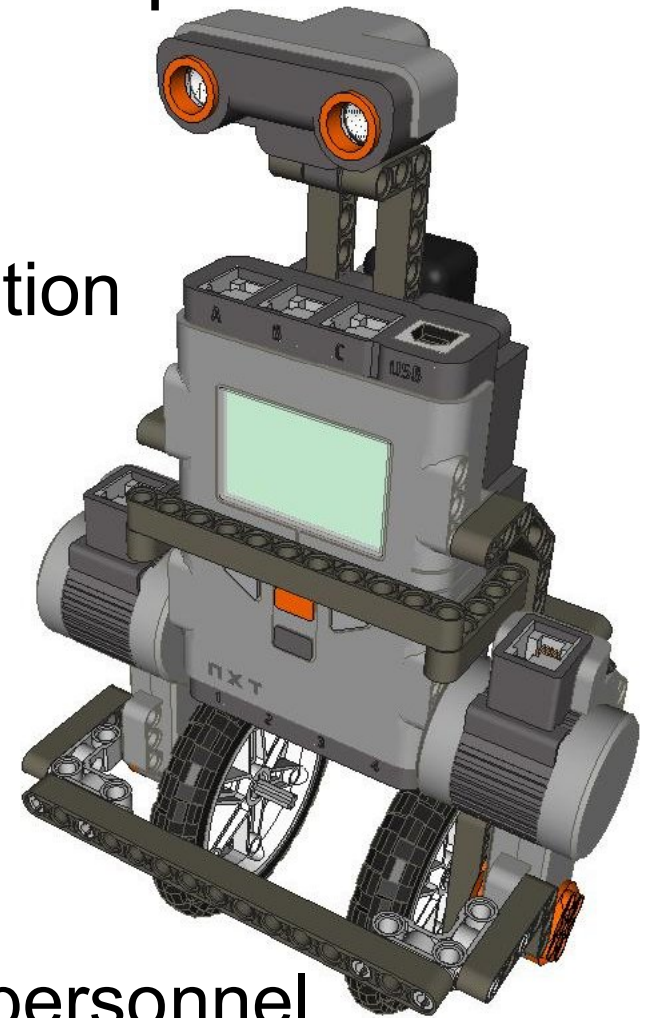
# Lego with Modelica/Dymola

- Build the NXTway two-wheel robot
- Physical modeling with Modelica
  - Multi-body dynamics
- Model calibration
  - Experiments
  - Dymola Calibration module
- Control design
  - Derive simple model
  - Develop control scheme

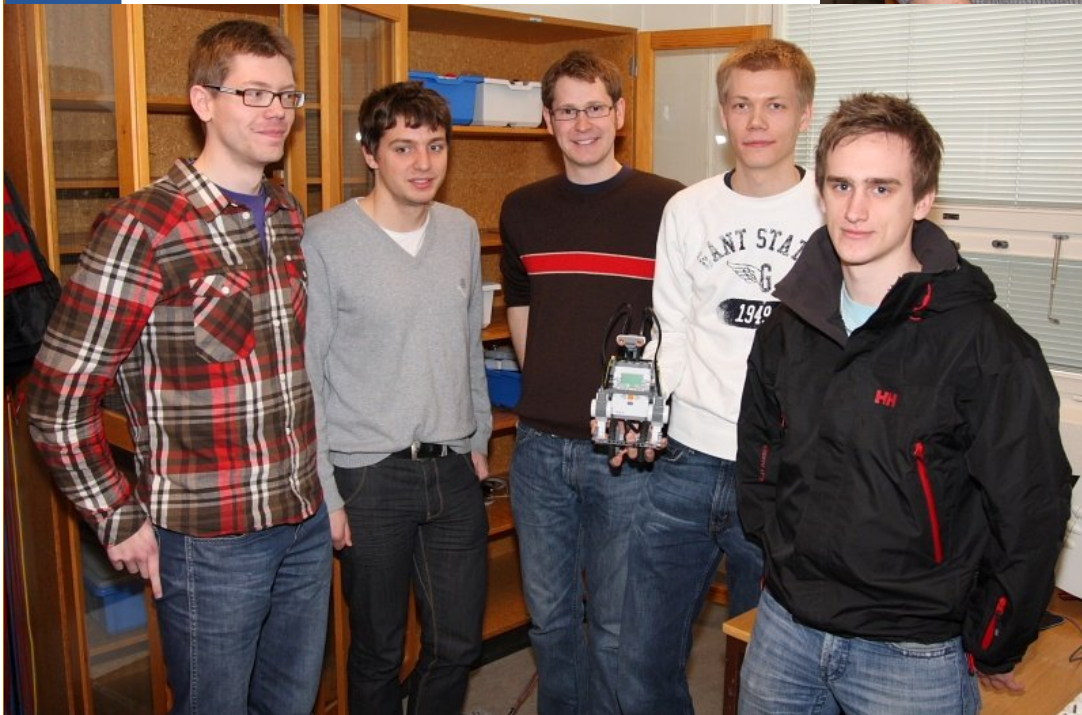
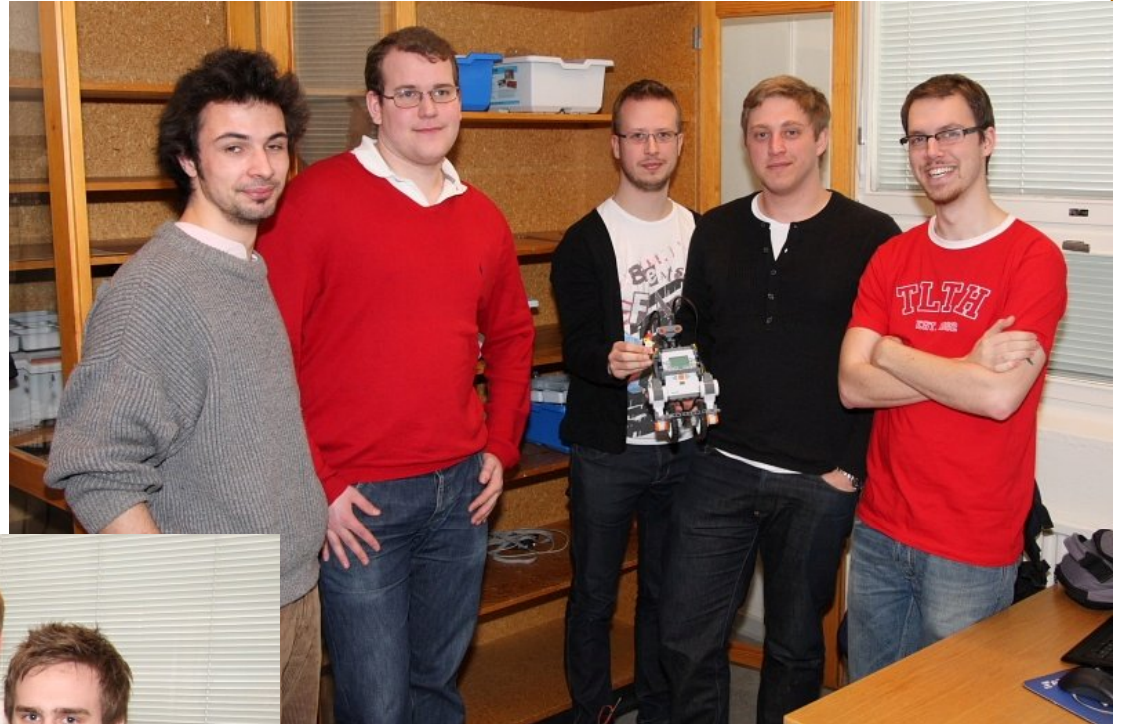


# Lego with Modelica/Dymola

- Automatic generation of fixed point controller code
  - No C programming
  - Software in the loop evaluation
- User interaction
- Deployment on NXTway
- Animation in Dymola
  - Real-time animation
- Get in touch with industry
  - Tutorials held by Dynasim personnel



# Lego Dymola Groups



# Course plan

- w1. Form groups and planning
  - Tuesday March 17<sup>th</sup> – group announcement
  - Friday March 20<sup>th</sup> – project plan dead line
- w2.-w3.
  - Tutorial
  - Weekly meetings with project supervisors
- w4.-w7
  - Weekly meetings with project supervisors
  - Presentation and demo in w7.





# Lego Dymola Tutorials

1. Introduction to Modelica (AC)
2. Multi-body modeling (Dynasim)
  - Wheel models (by Martin Otter)
3. Code generation with Dymola (Dynasim)
  - Modelica\_Embedded

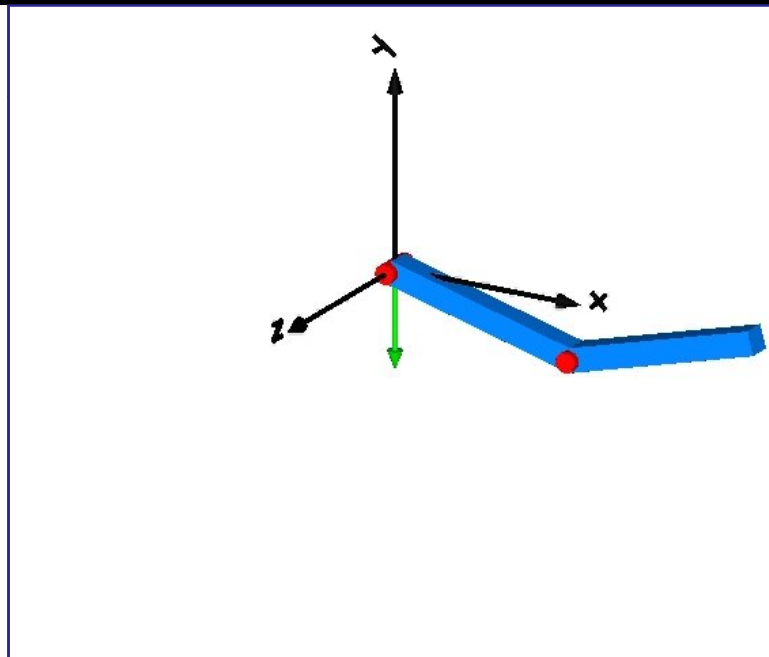


# Modelica/Dymola Modeling

Hilding Elmqvist

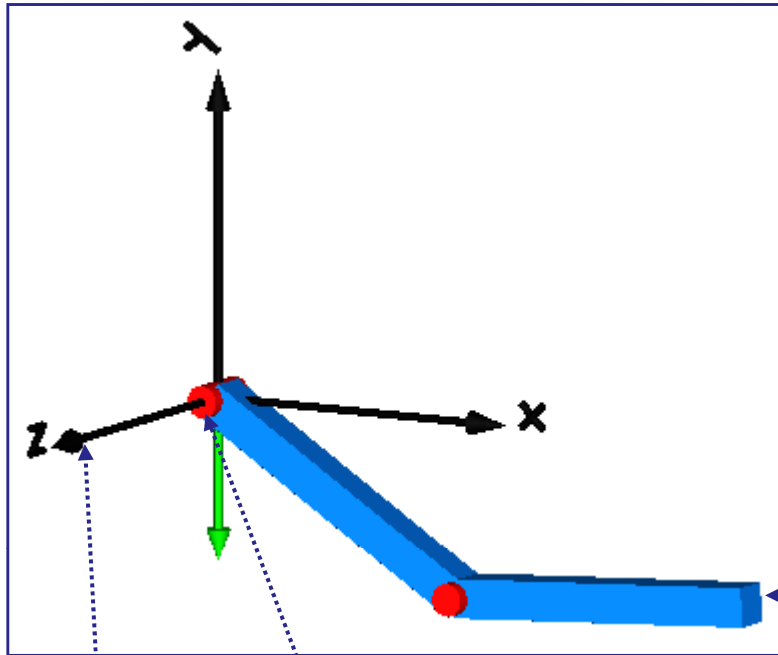
# MultiBody modeling

- **Modelica.Mechanics.MultiBody**
- **Bodies and Joints**
- **Automatic 3D animation**



- Modelica
- + Users Guide
- + Blocks
- + Constants
- + Electrical
- + Icons
- + Math
- Mechanics
  - MultiBody
    - + User's Guide
    - World
    - + Examples
    - + Forces
    - + Frames
    - + Interfaces
    - + Joints
    - + Parts
    - + Sensors
    - + Types
    - + Visualizers
    - + Rotational
    - + Translational
  - + Slunits
  - + StateGraph
  - + Thermal
  - + Utilities

# Example – Double pendulum



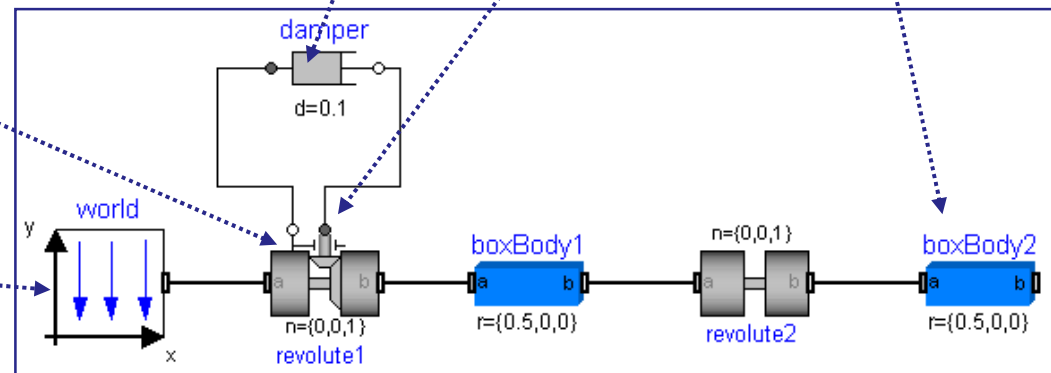
1D rotational damper

Bodies

1D rotational flange for drive

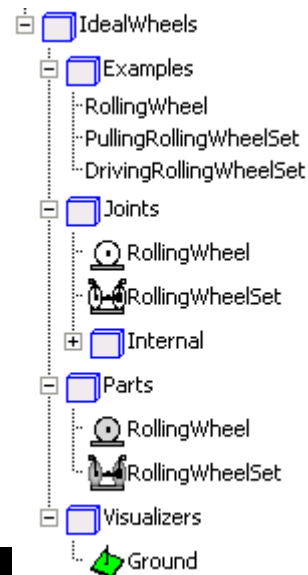
Rotational joint

World system  
(= Inertial system)



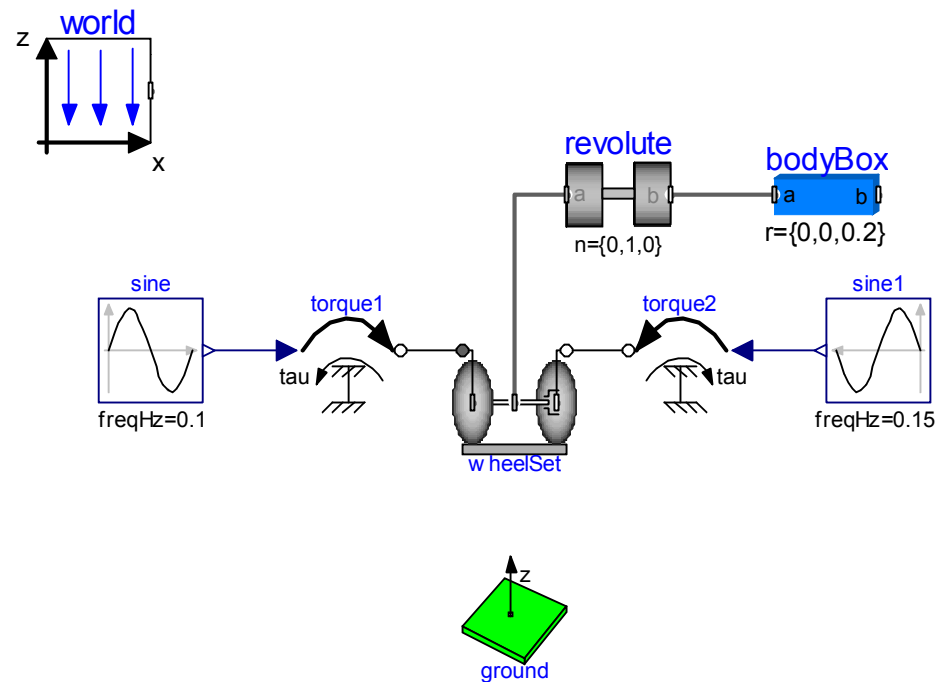
# Wheel Model

- The MultiBody library does not contain any wheel models.
- A separate library IdealWheels contains a wheel set where each wheel can be driven separately.
- This is thus suitable for modeling of LEGO Mindstorms robots

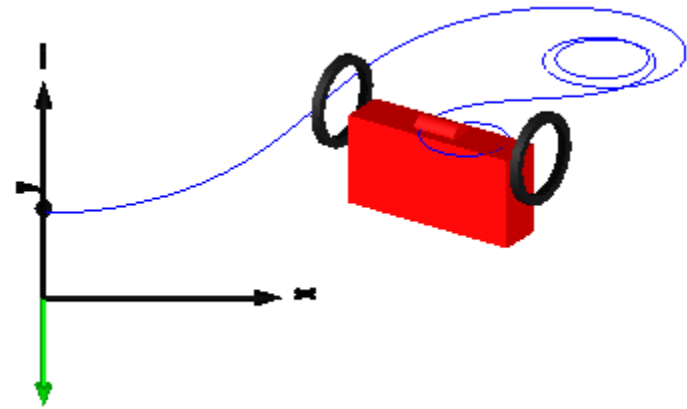
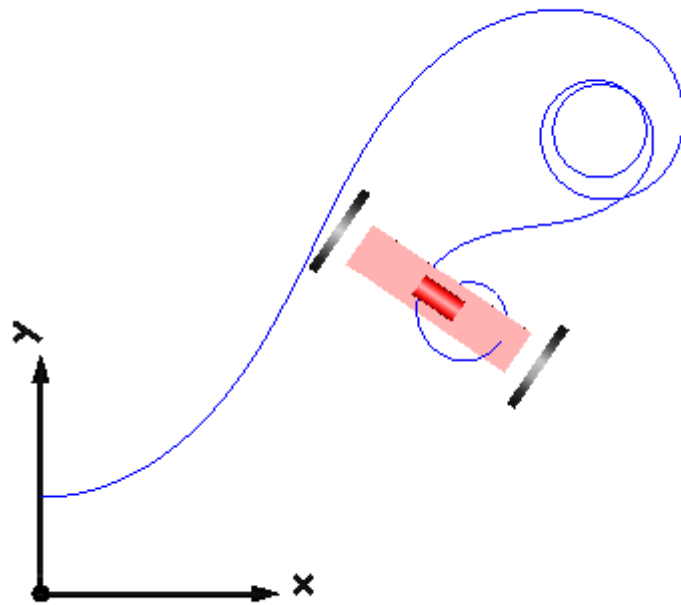


# Body with two wheels

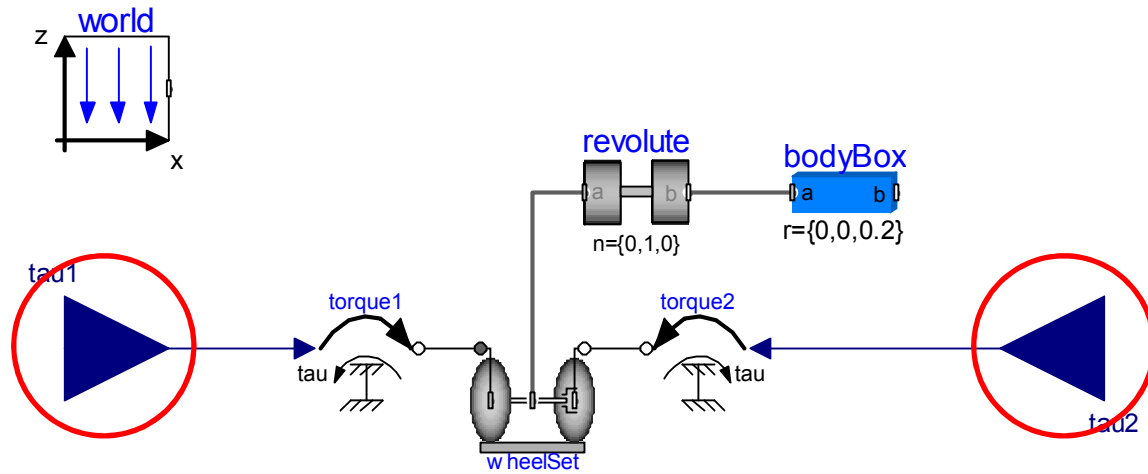
- The separate model WheelSetAndBody contains a body with two wheels.
- The wheels are driven by two sinusoidal torques with different frequencies.



# Resulting animations



# Linearization



Simulation Plot

- Run Script...
- Translate
- Simulate
- Continue ▶
- Stop
- Linearize**
- Setup...
- Visualize
- Show Log

$X_0$

Variables	Values	Unit
<input type="checkbox"/> x	0	m
<input type="checkbox"/> der(x)		m/s
<input type="checkbox"/> y	0.2	m
<input type="checkbox"/> der(y)		m/s
<input type="checkbox"/> phi	0	rad
<input type="checkbox"/> der(phi)		rad/s
<input type="checkbox"/> theta1	0	rad
<input type="checkbox"/> der(theta1)		rad/s
<input type="checkbox"/> theta2	0	rad
<input type="checkbox"/> der(theta2)		rad/s
<input type="checkbox"/> der_theta1	0	rad/s
<input type="checkbox"/> der(der_theta1)		rad/s/s
<input type="checkbox"/> der_theta2	0	rad/s
<input type="checkbox"/> der(der_theta2)		rad/s/s



# Linear model

Packages

- Modelica\_LinearSystems
  - Users Guide
  - StateSpace
    - Examples
      - constructor
      - fromZerosAndPoles
      - fromModel
      - fromFile
      - fromReal
      - fromTransferFunction

Modelica\_LinearSystems.StateSpace.fromFile

fromFile

Description

Generate a StateSpace data record by reading linear

Inputs

fileName: "dslin.mat"

showEigenValues: true

OK Info Co

Commands

```

= Modelica_LinearSystems.StateSpace(
  A =
  [0, 0, 0, 0, 0, 0, 0.05, 0.05, 0, 0;
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0;
  0, 0, 0, 0, 0, 0, (-0.2), 0.2, 0, 0;
  0, 0, 0, 0, 0, 0, 1, 0, 0, 0;
  0, 0, 0, 0, 0, 0, 0, 1, 0, 0;
  0, 0, 0, 0, 0, 0, 0, 0, (-201.993091925152), 0;
  0, 0, 0, 0, 0, 0, 0, 0, (-201.993091925152), 0;
  0, 0, 0, 0, 0, 0, 0, 0, 0, 1;
  0, 0, 0, 0, 0, 0, 0, 0, (-211.830417830859), 0],
  B =
  [0, 0;
  0, 0;
  0, 0;
  0, 0;
  0, 0;
  10.4163334752606, (-0.945581386124012);
  (-0.945581386124056), 10.4163334752606;
  0, 0;
  3.34261838440113, 3.34261838440117],
  C =
  [],
  D =
  [],
  {"taul", "tau2"}, {}, {"wheelSet.x", "wheelSet.y", "wheelSet.phi",
  "wheelSet.theta1", "wheelSet.theta2", "wheelSet.der_theta1", "wheelSet.der_theta2",
  "revolute.phi", "revolute.w"}]
    
```

Inputs

Outputs

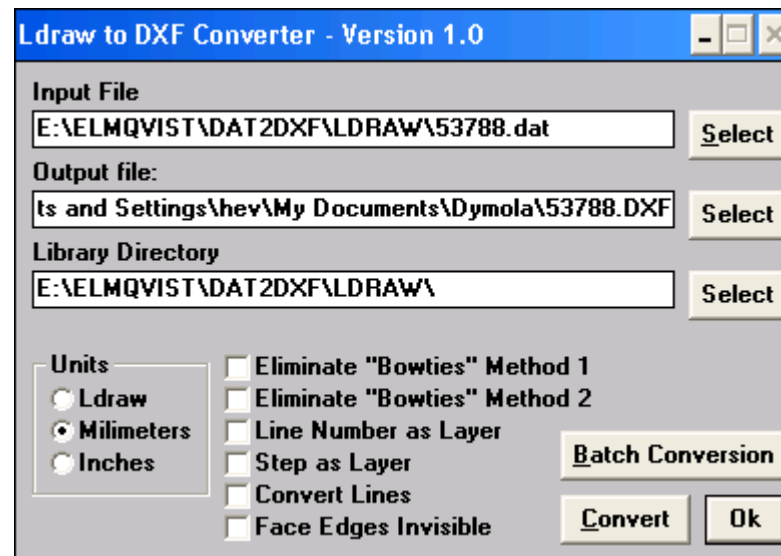
States

## Animation with LEGO shapes

- Dymola supports visualization of shapes described by DXF, STL and VRML (subset) files.
- LEGO building blocks are described in .DAT format:
  - <http://www.ldraw.org>
- **DAT2DXF converter available:**
  - <http://www.ldraw.org/Downloads-req-viewdownload-cid-6.html>
- **LEGO parts are available at:**
  - <http://www.ldraw.org/Downloads-req-viewdownload-cid-1.html>
- **Electric Mindstorms NXT shape:**
  - <http://www.ldraw.org/cgi-bin/ptdetail.cgi?f=parts/53788.dat>

# Ldraw to DXF Converter

- **Download parts directories from:**
  - <http://www.ldraw.org/Downloads-req-viewdownload-cid-1.html>
- **Mindstorms parts are among the unofficial parts**
- **See dat2dxf\readme.txt for instructions on how to organize files**



# LEGO blocks resource

File Detail Part 53788 - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address <http://www.ldraw.org/cgi-bin/ptdetail.cgi?f=parts/53788.dat>

Google Search

LDraw.org  
Centralized LDraw Resources

Parts Tracker :: [Parts List](#) :: [Activity](#) :: [Submit](#) :: [PT Tools](#) :: [PT Reference](#) :: [LDraw Specifications](#) :: [Lookup](#)  [Go](#)

**Unofficial File parts/53788.dat**  
[Next File](#) | [Prev File](#) | [Download](#) | [Review](#) | [Edit](#) | [CA Header Edit](#) | [Events](#)

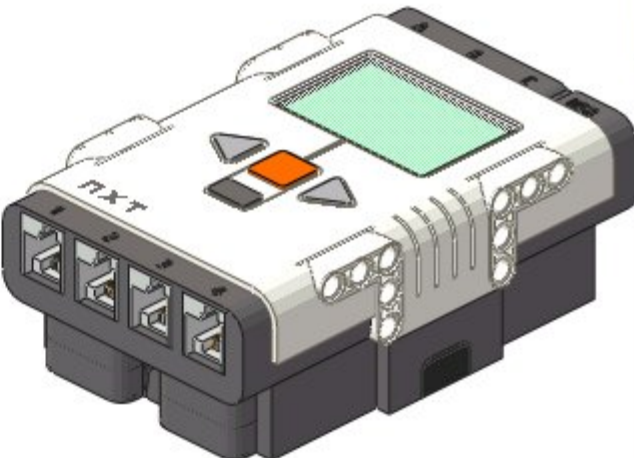
**File Header:**

```
0 Electric Mindstorms NXT (Complete Shortcut)
0 Name: 53788.dat
0 Author: Kevin Clague [kclague]
0 !LDRW_ORG Unofficial_Part
0 !LICENSE Redistributable under CCAL version 2.0 : see CAreadme.txt

0 BFC CERTIFY CCW

0 !HISTORY 2006-03-30 [kclague] Initial design
0 !HISTORY 2007-12-30 [Philo] Complete redesign, use stone colors

0 // Battery Lid - Dark Stone
```

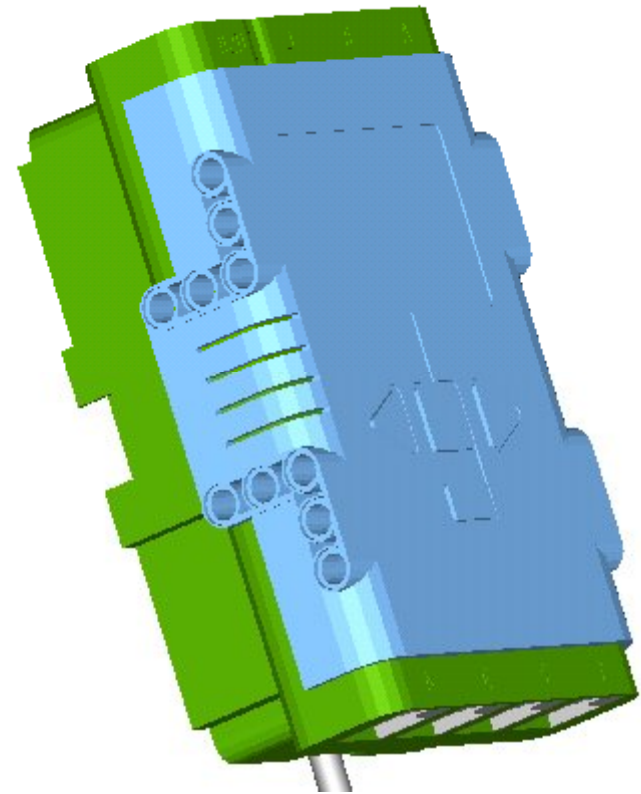
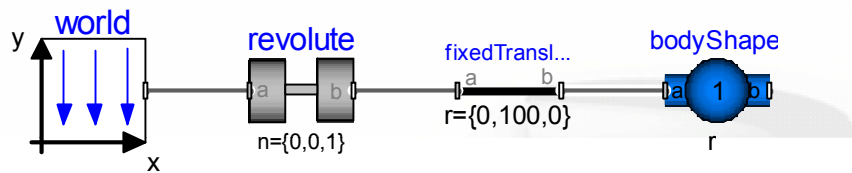


**Status:**  
■ 7 subfiles aren't certified. (CSSSSSSX)  
Size: 1153 bytes

Internet

# Dymola animation of NXT (shape 53788)

---



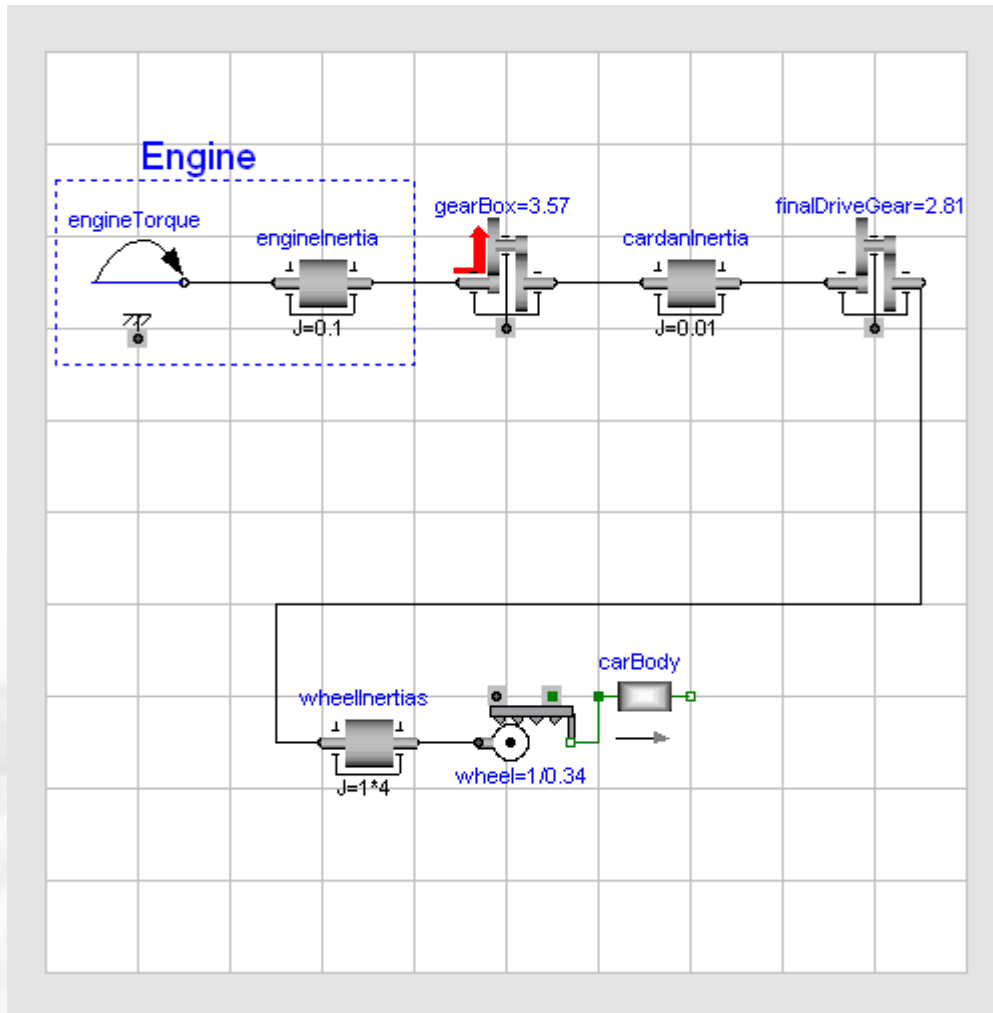
---

# Tutorial

## Modelica and Dymola for System Design Model Building and Calibration

H. Elmqvist  
Dynasim AB, Lund

# Calibrate engine map parameter and friction



# Modelica\_EmbeddedSystems and Code generation for Lego Mindstorms NXT

Ulf Nordström



# Contents

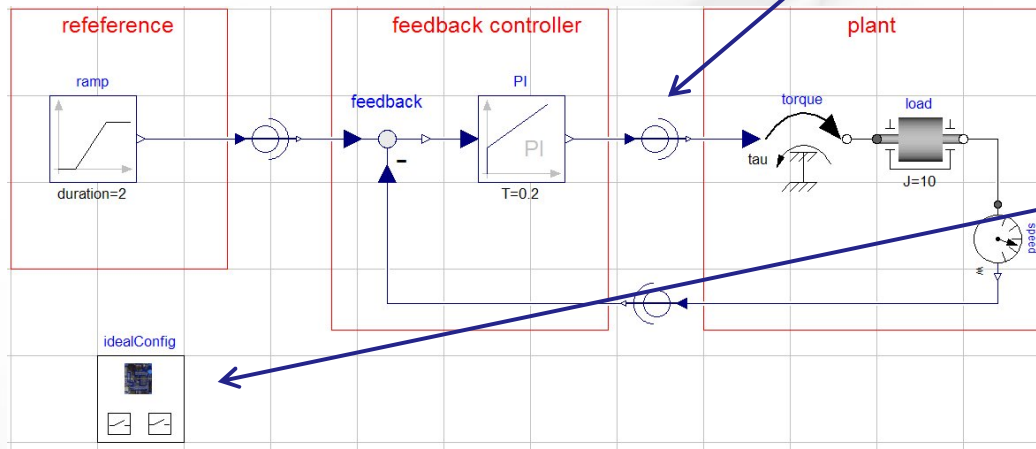
---

- Overview
- Modelica\_EmbeddedSystems
- Fixed point Code Generation
- Lego Mindstorms

# Modelica\_EmbeddedSystems

## Key Components

- Communication
- Configuration
  - Target
  - Task
  - Subtask



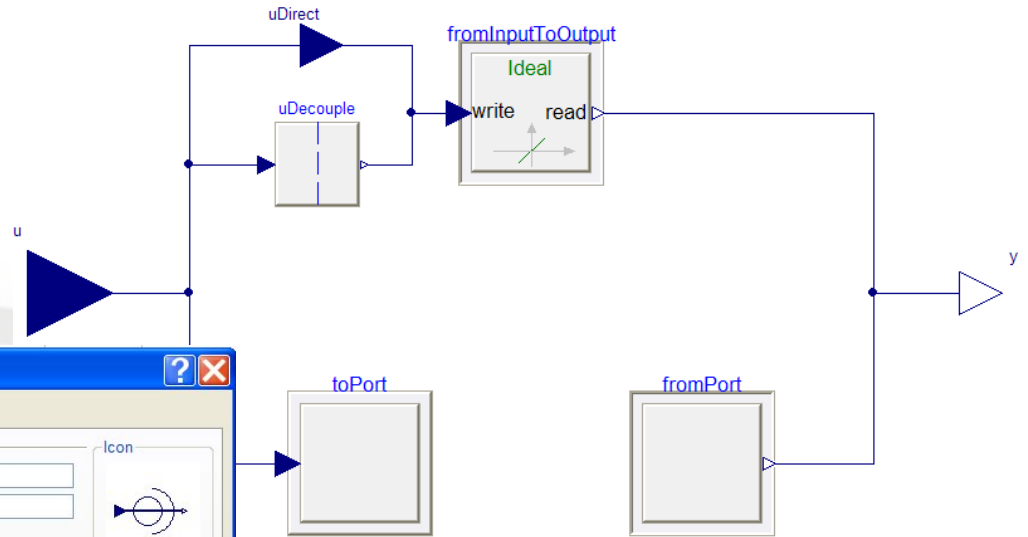
- Modelica\_EmbeddedSystems
  - Users Guide
  - Examples
  - Interfaces
    - CommunicateReal**
    - CommunicateBoolean
    - CommunicateInteger
  - BaseReal
  - BaseBoolean
  - BaseInteger
  - Communication
  - Configuration
    - Target**
    - Task**
    - Subtask**
    - Bus
  - Types
  - Icons
  - BuiltIn

# Modelica\_EmbeddedSystems

## Key Components

### Communication

Input and output reside in the same base-rate task  
decoupleReal specifies boundary between subtasks



**communicateReal in Unnamed**

General Add modifiers

Component Name:   
Comment:

Model Path: Modelica\_EmbeddedSystems.Interfaces.CommunicateReal  
Comment: Interface between tasks to communicate Real signals

Icon:

Communication between input and output

communicationType	<input type="text" value="ss.CommunicationType.DirectCommunication"/>	Type of communication
fromInputToOutput	<input type="text" value="Output if (not (useToPort or useFromPort))"/>	Communication block (simulated or subtask communication)
toPort	<input type="text" value="ubtask.samplePeriodFactor) if useToPort"/>	Communication port to which the input signal is transmitted
fromPort	<input type="text" value="ltask.samplePeriodFactor) if useFromPort"/>	Communication port from which the output signal is received

Sampling and other configurations of subtask to which input and/or output belongs (if de-activated, the information is defined somewhere else)

defineInSubtask	<input type="text" value="true"/>	= true, if sampling/configuration for input is defined
defineOutSubtask	<input type="text" value="false"/>	= true, if sampling/configuration for output is defined
inSubtask	<input type="text" value="Modelica_EmbeddedSystems.Configuration.Subtask()"/>	Sampling/configuration for input
outSubtask	<input type="text" value="Modelica_EmbeddedSystems.Configuration.Subtask()"/>	Sampling/configuration for output

OK Info Cancel

Communication using ports  
Inter-task communication or using external I/O

# Modelica\_EmbeddedSystems

## Key Components

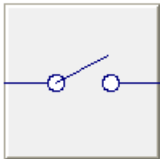
## Configuration



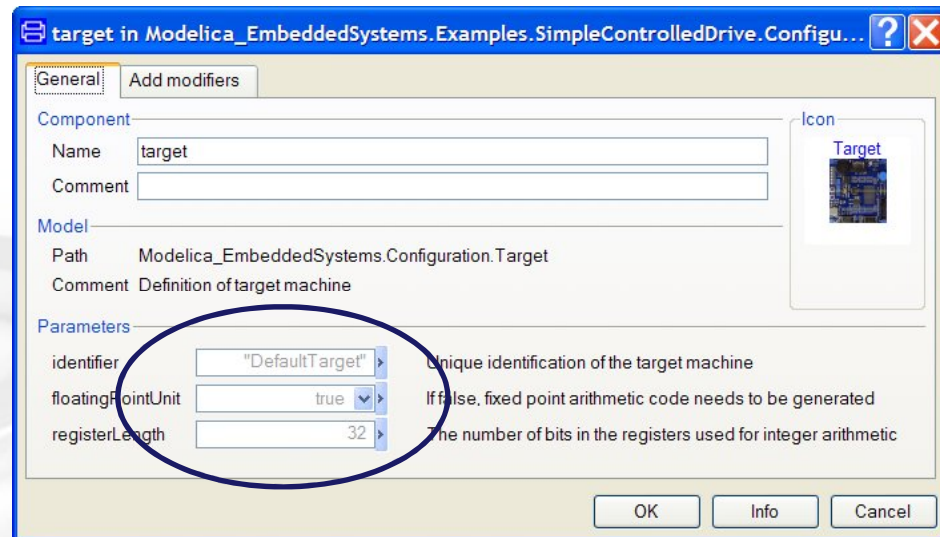
**Target**



**Task**



**Subtask**



# Modelica\_EmbeddedSystems

## Key Components

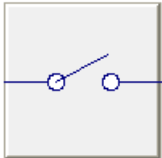
## Configuration



Target



Task



Subtask

defaultTask in Modelica\_EmbeddedSystems.Examples.SimpleControlledDrive.Configurations.Ideal2

General | Add modifiers

Component

Name: defaultTask

Comment:

Model

Path: Modelica\_EmbeddedSystems.Configuration.Task

Comment: Definition of (asynchronous) task running on a target

Parameters

onTarget	target	Target on which the task is running
identifier	"defaultTask"	Unique identification of the task on the target
priority	1	Fixed priority value of task (may be overridden depending on scheduling policy)
sampleBasePeriod	0.002 s	Sample base period for periodic subtasks
onProcessor	0	Processor of the target on which the task is running

Icon: Task on

OK Info Cancel

# Modelica\_EmbeddedSystems

## Key Components

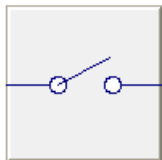
## Configuration



Target



Task



Subtask

reference in Modelica\_EmbeddedSystems.Examples.SimpleControlledDrive.Configurations.Ideal2

General Add modifiers

Component

Name reference

Comment

Model

Path Modelica\_EmbeddedSystems.Configuration.Subtask

Comment Identifies a set of equations inside a task that are executed in the same way (e.g. same sample period, same integrators)

Icon

Subtask

in

Parameters

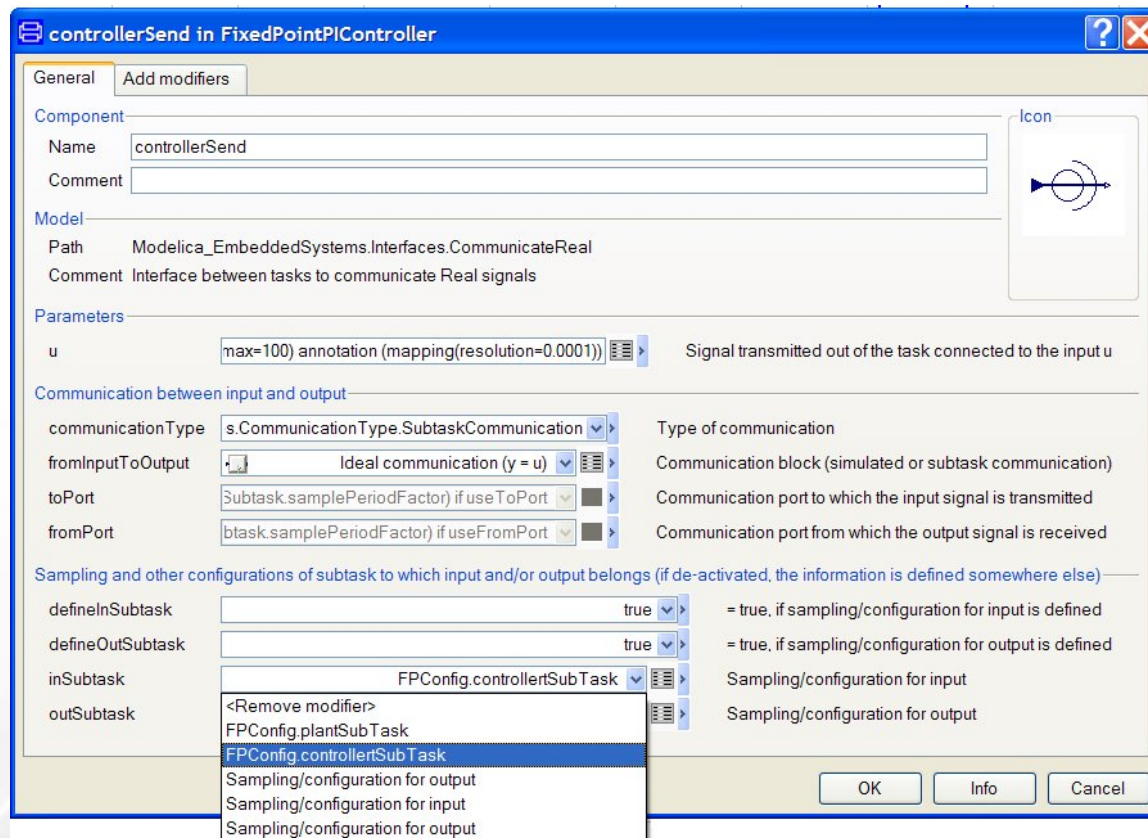
inTask	defaultTask	Task in which subtask is running (synchronization between subtasks is automatic due to equation sorting)
identifier	"reference"	Identifier of subtask (unique within the task)
samplingType	Types.SamplingType.Periodic	Type of subtask (Continuous, Periodic, Triggered, Disabled)
samplePeriodFactor	5	If periodic subtask: Sample period = samplePeriodFactor*task.sampleBasePeriod
sampleOffsetFactor	0	If periodic subtask: Offset = sampleOffsetFactor*task.sampleBasePeriod
integrationMethod	Types.IntegrationMethod.FixedStepTrapezoid	Integration method
fixedStepSize	samplePeriodFactor*inTask.sampleBasePeriod	Step size for fixed step integration method

OK Info Cancel

# Modelica\_EmbeddedSystems

## Key Components

Use pull-down menu to select target/task/subtask”



# Modelica\_EmbeddedSystems

---

## *Scenarios*

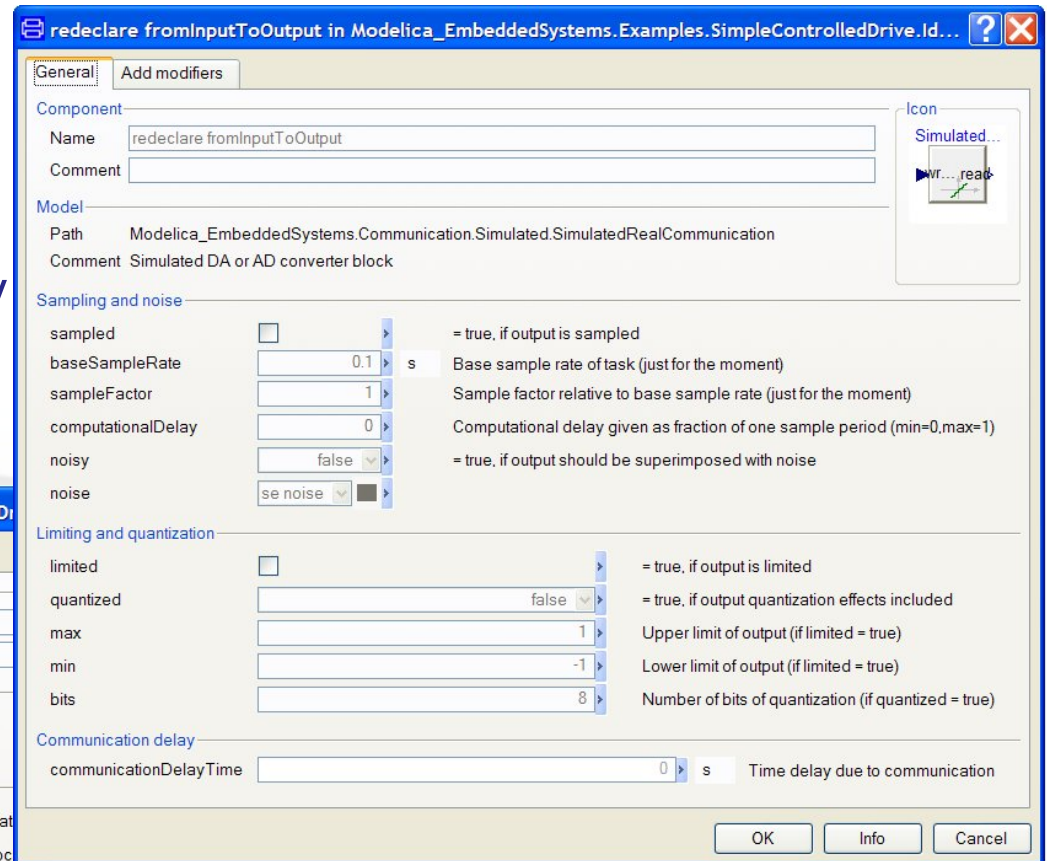
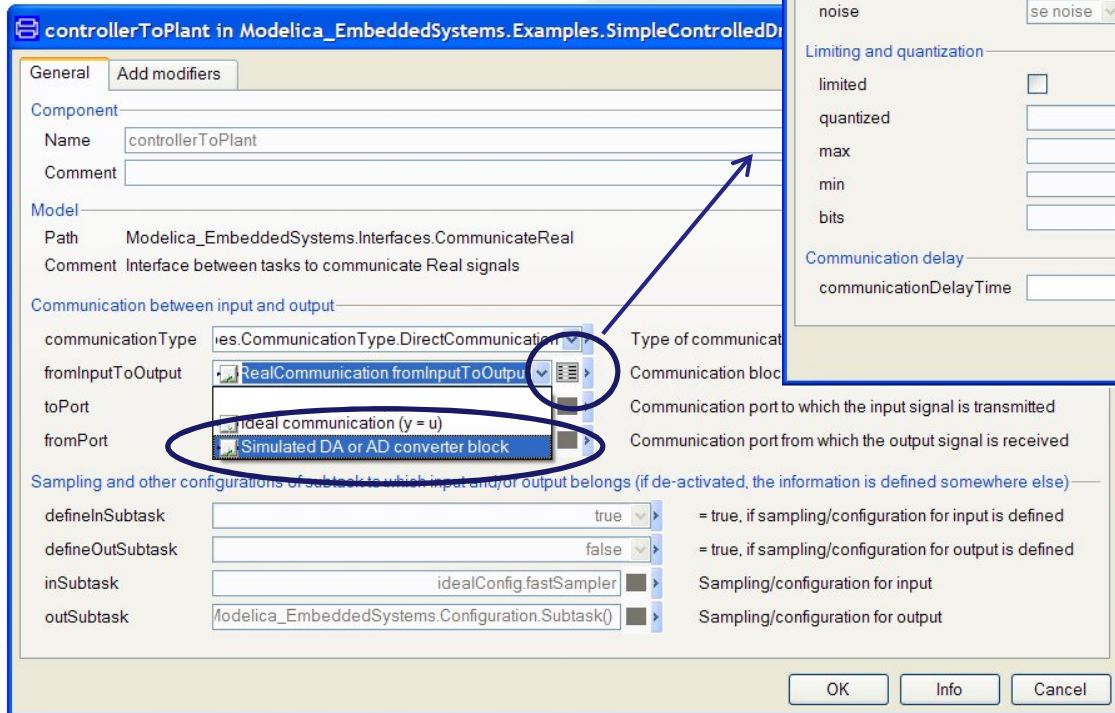
- **Model In the Loop**
  - Quantization
  - Communication delay
- **Software In the Loop**
  - Task decomposition
  - Effects of fixed point arithmetics
- **Embedded**



# Modelica\_EmbeddedSystems

## Scenarios

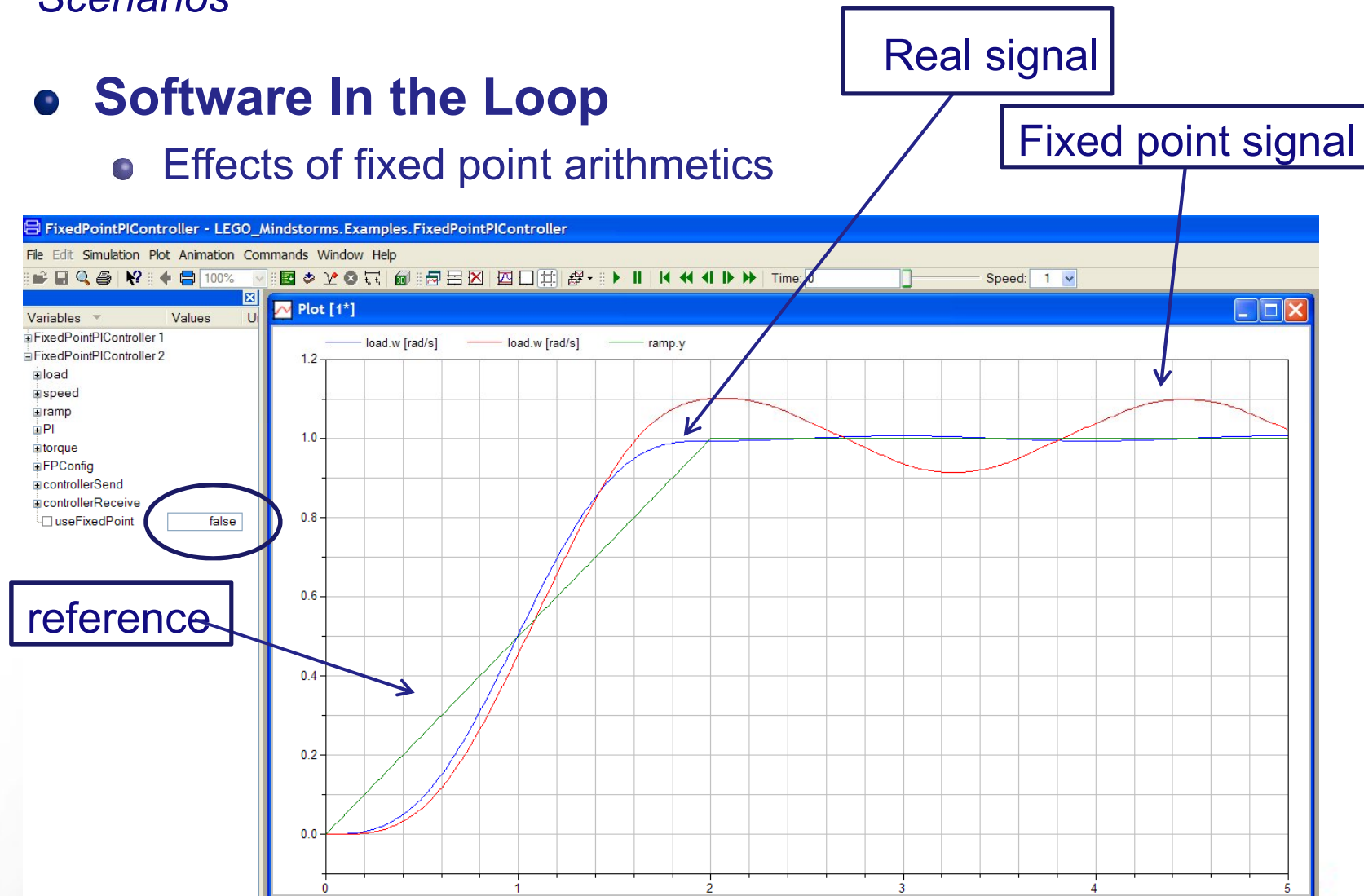
- **Model In the Loop**
  - Quantization
  - Communication delay



# Modelica\_EmbeddedSystems

## Scenarios

- **Software In the Loop**
  - Effects of fixed point arithmetics



# Modelica\_EmbeddedSystems

---

## Scenarios

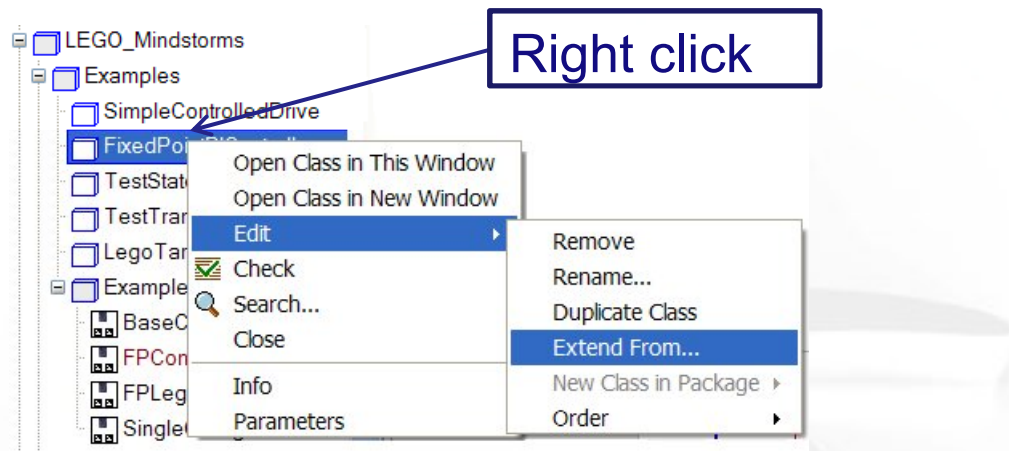
- **Embedded**

- Change target identifier to "lego"
- Will be discussed in the Lego Mindstorms section

# Modelica\_EmbeddedSystems

## Scenarios

- Use "extend from" to create new configurations/scenarios from the same base model
  - Use modifiers to change attributes



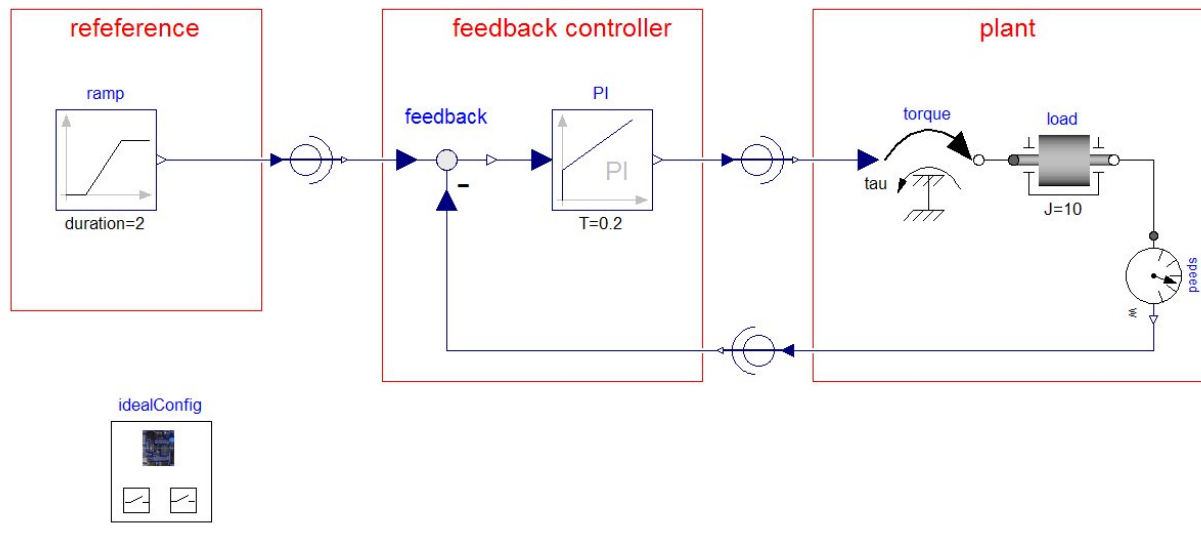
# Modelica\_EmbeddedSystems

Example – Software In the Loop

Set `Hidden.DecomposeInTasks = true`

## Task decomposition

- Reference subtask
- Feedback subtask
- Plant subtask

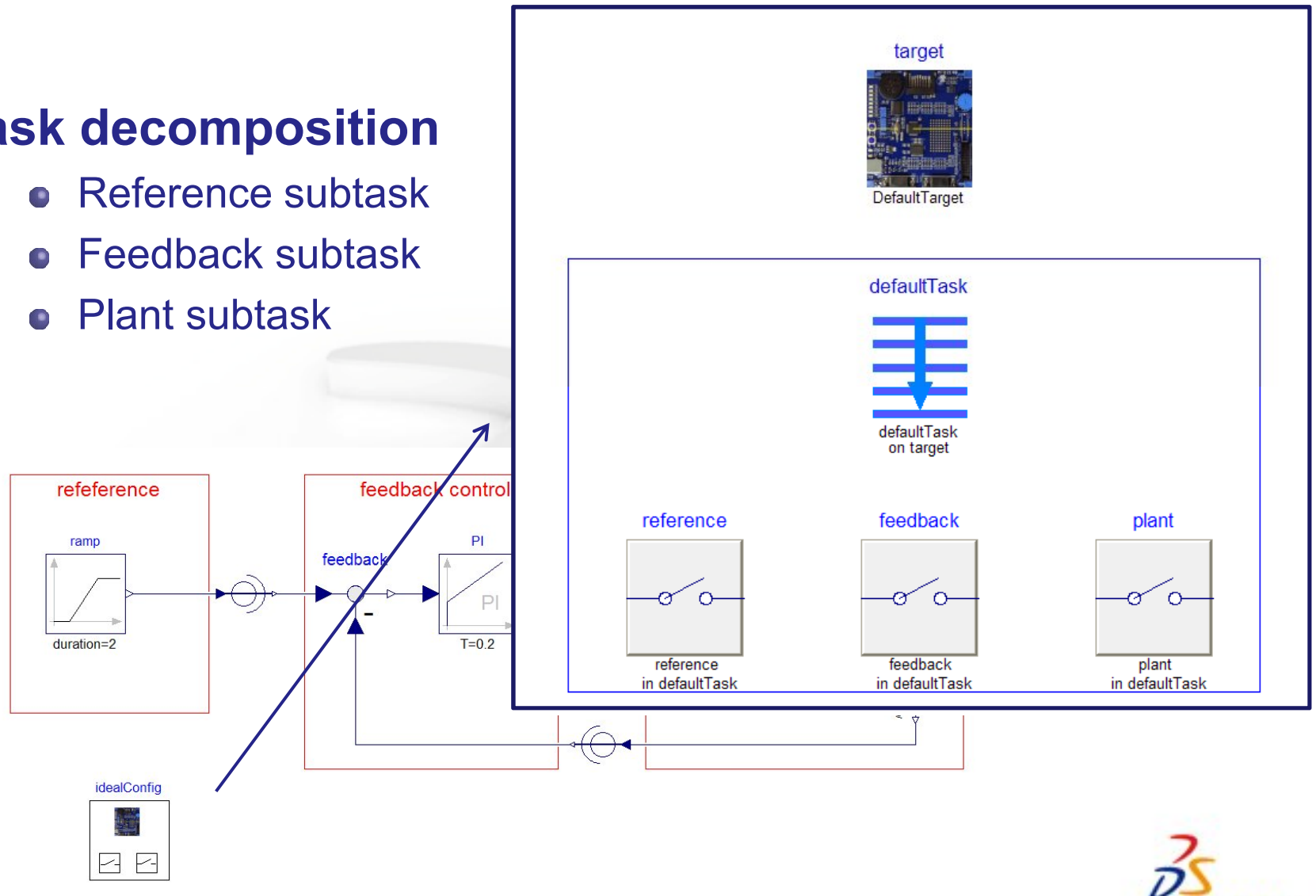


# Modelica\_EmbeddedSystems

Example – Software In the Loop

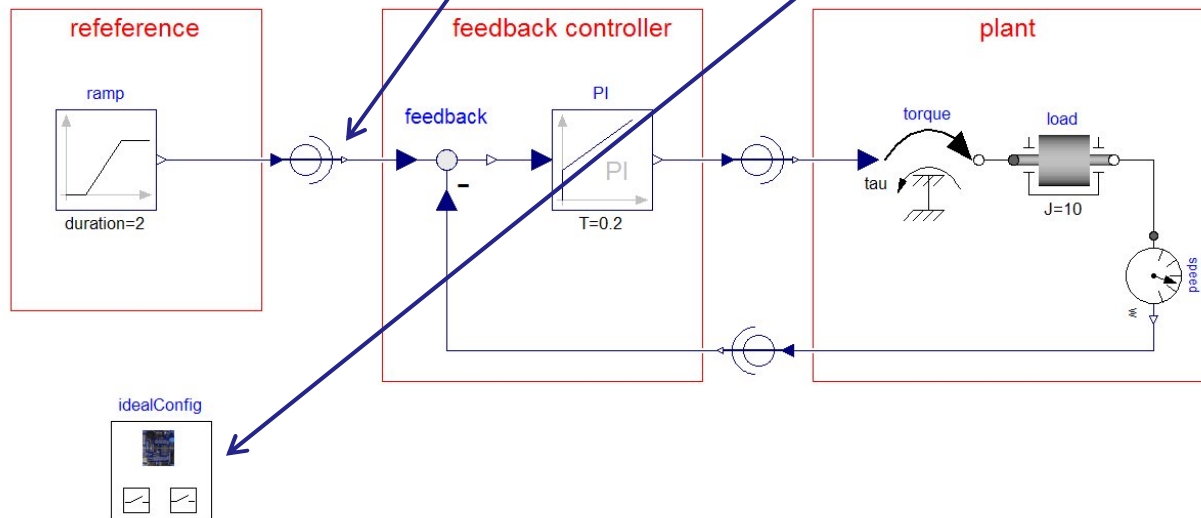
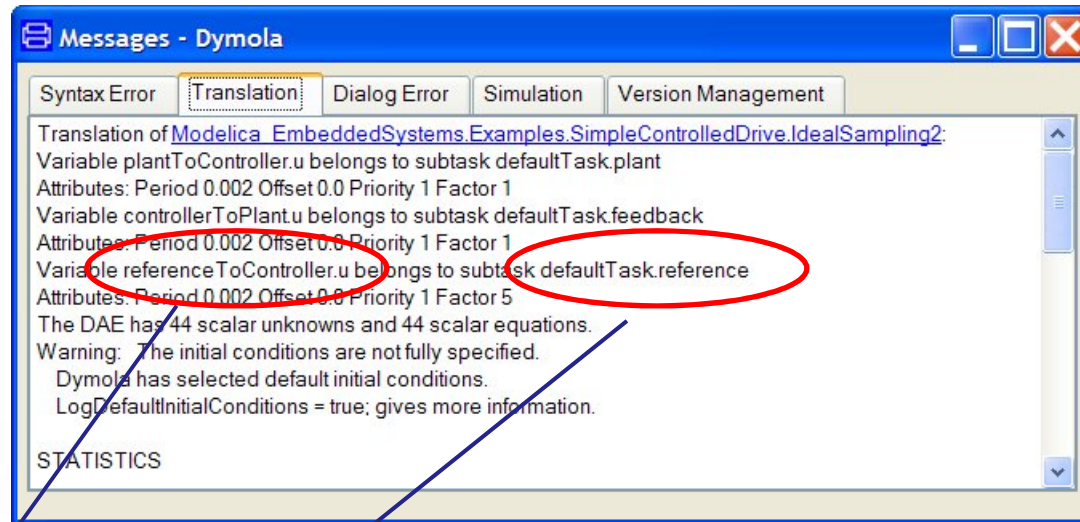
## Task decomposition

- Reference subtask
- Feedback subtask
- Plant subtask



# Modelica\_EmbeddedSystems

## Example – Software In the Loop



# FixedPoint Code Generation

---

- **Setup/Configuration**
  - Mapping annotation
- **Restrictions**



# FixedPoint Code Generation

---

*Setup/Configuration – mapping annotation*

## FixedPoint variables must be annotated

- min
- max
- resolution

```
Modelica.Blocks.Sources.Ramp ramp(  
  height(  
    min=0,  
    max=100) = 100 annotation (mapping(resolution=0.001)),  
  duration(  
    min=0,  
    max=50) = 10 annotation (mapping(resolution=0.001)),  
  y(min=0, max=100) annotation (mapping(resolution=0.01)),|
```

# FixedPoint Code Generation

## Setup/Configuration – mapping annotation

### Declaration

```
/* output Modelica.Blocks.Interfaces.RealOutput ramp.y(min = 0.0, max = 100.0)
   annotation(mapping(resolution = 0.01));*/
int ramp_yFP;    /* Q[7, 0] */

/* parameter Modelica.SIunits.Time ramp.duration(min = 0.0, max = 50.0) = 10
   annotation(mapping(resolution = 0.001));*/
int ramp_durationFP = 320; /* Q[6, 5] */

/* parameter Real ramp.height(min = 0.0, max = 100.0) = 100
   annotation(mapping(resolution = 0.001));*/
int ramp_heightFP = 1600; /* Q[7, 4] */
```

Q[nQ, nQ] = [integer bits, fractional bits]

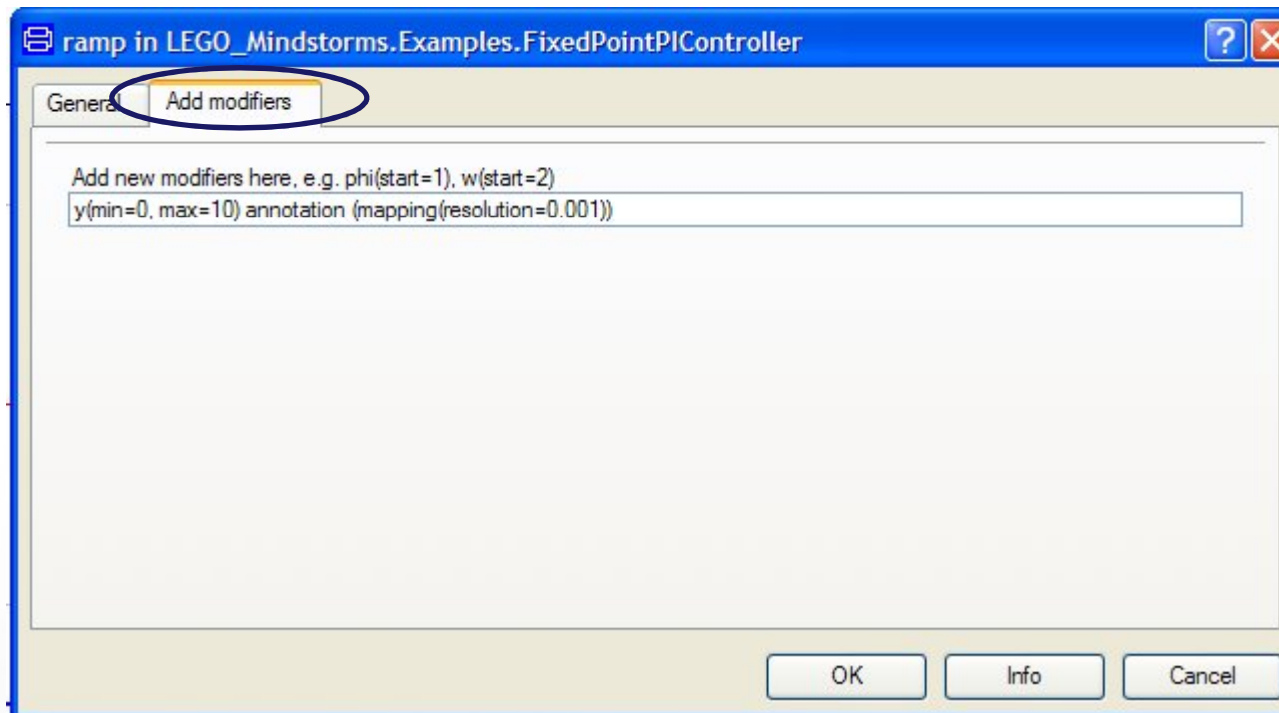
### Generated code

```
/* ramp.y = ramp.offset+(if time < ramp.startTime then 0 else (if time <
   ramp.startTime+ramp.duration then (time-ramp.startTime)*ramp.height/
   ramp.duration else ramp.height)); */
ramp_yFP = (((ramp_offsetFP << 9) + (((timeFP0_0 < (ramp_startTimeFP << 5)) ? (0
   << 9) : (((timeFP0_0 < ((ramp_startTimeFP + ramp_durationFP) << 5)) ? (((
   timeFP0_0 - (ramp_startTimeFP << 5)) * ramp_heightFP) / ramp_durationFP) : (
   ramp_heightFP << 5)))))) >> 9;
```

# FixedPoint Code Generation

*Setup/Configuration – mapping annotation*

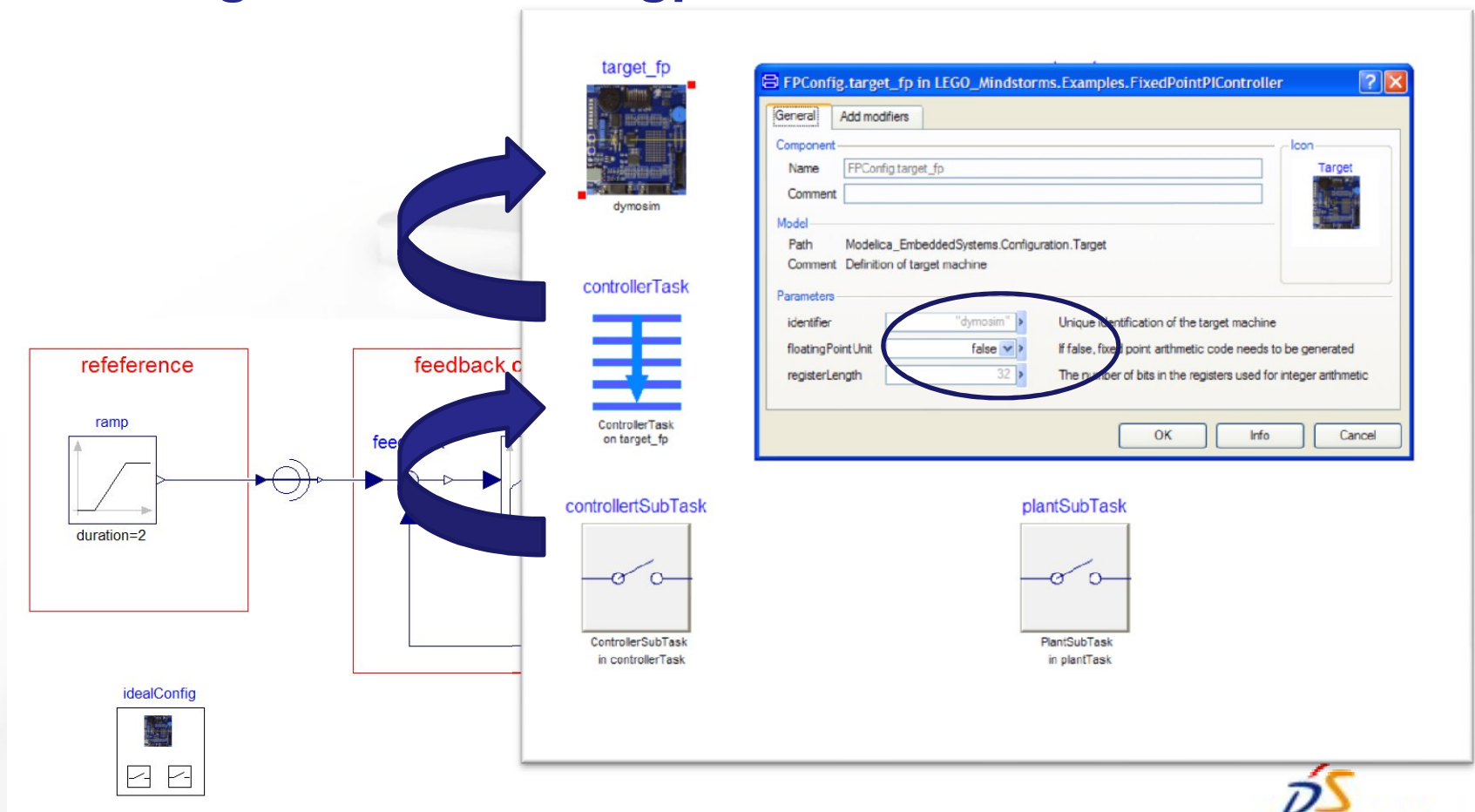
**Use parameter dialog to input annotations as modifiers**



# FixedPoint Code Generation

Setup/Configuration – enable FixedPoint

FixedPoint will be enabled when the subtask belongs to a target with "floatingpointUnit = false"



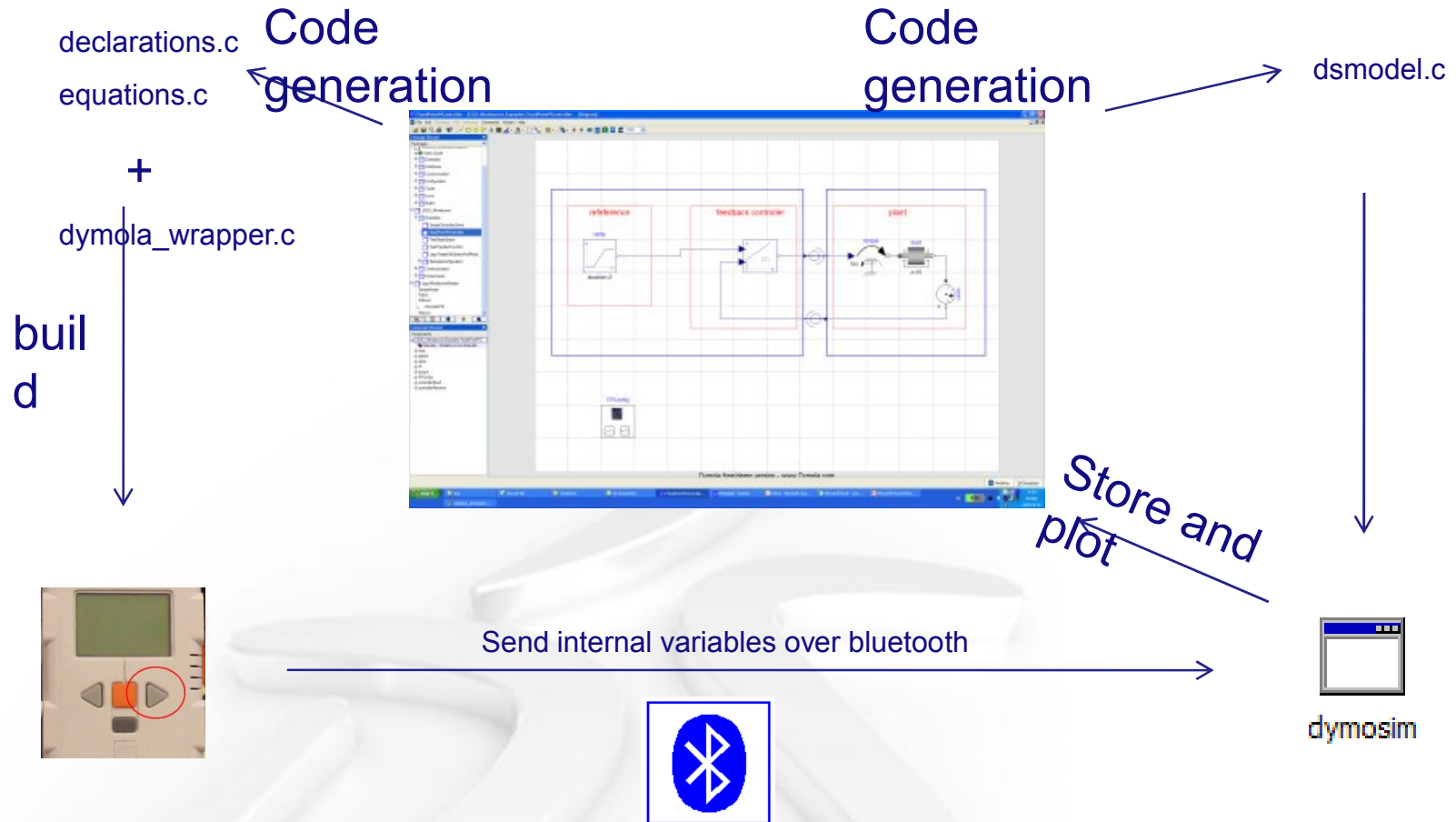
# Lego Mindstorms

---

- **Framework**
  - dsmodel and dymosim
  - dymola\_wrapper in nxtOSEK
- **Setup/Configuration**
- **Limitations**
- **Example**

# Lego Mindstorms

## Framework



# Lego Mindstorms

Framework – *dymola\_wrapper.c*

```
#include "kernel.h"
#include "kernel_id.h"
#include "ecrobot_interface.h"
#include "target_port.h"
/* OSEK declarations */
DeclareCounter(SysTimerCnt);
DeclareTask(Task1);
/* include fixedpoint variable declarations */
#include "declarations.c"
...
/* Task1 executed every 50msec */
TASK(Task1)
{
    /* map system time to fixedpoint time */
    timeFP0_0 = (int)1024*systick_get_ms()/1000;

    /* include fixedpoint equations */
    include "equations.c"

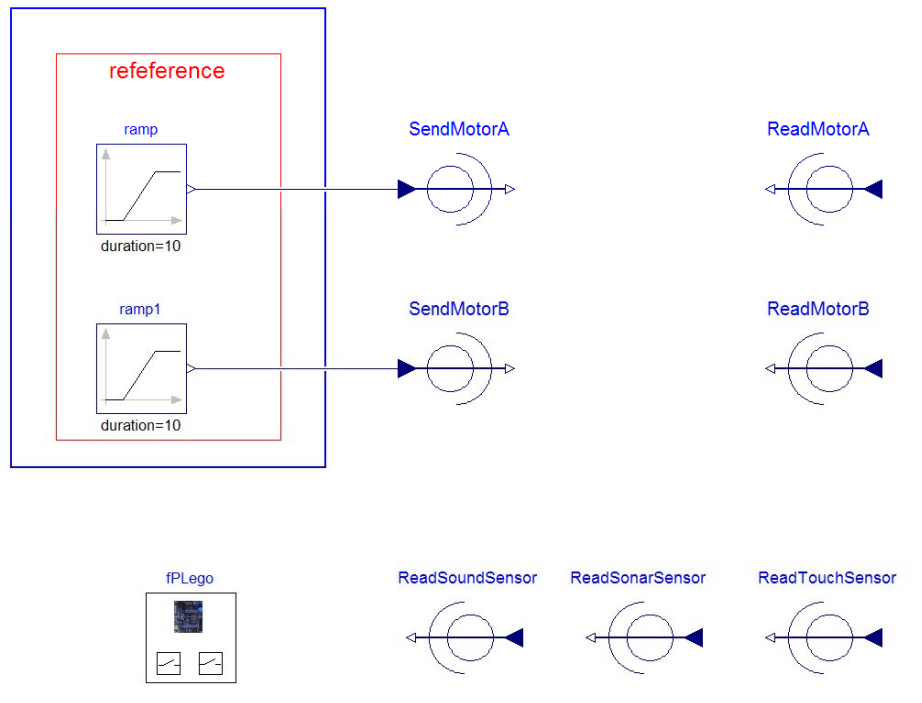
    TerminateTask();
}
```

Generated by  
Dymola

# Lego Mindstorms

## Setup/Configuration

### Configuration for Lego target and bluetooth communication





# Lego Mindstorms

---

## *Setup/Configuration*

### Translate and compile Dymola model

- CD to `"..\nxtOSEK\samples\dymola"`
- Press translate button in Dymola

### Compile dymola\_wrapper using Cygwin

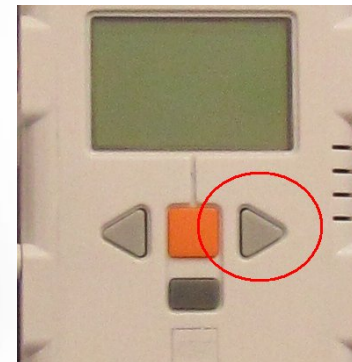
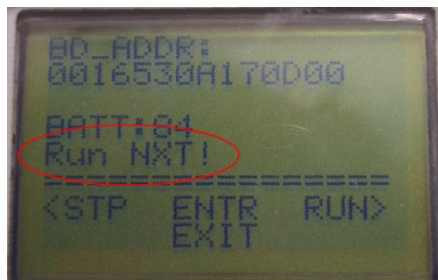
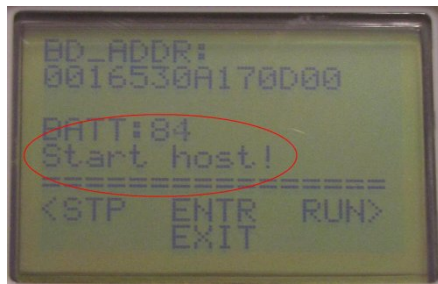
- CD to `"..\nxtOSEK\samples\dymola"`
- Compile with `"make all"`
- Start Lego and download executable with `"./ramboot.sh"`

...

# Lego Mindstorms

## Setup/Configuration

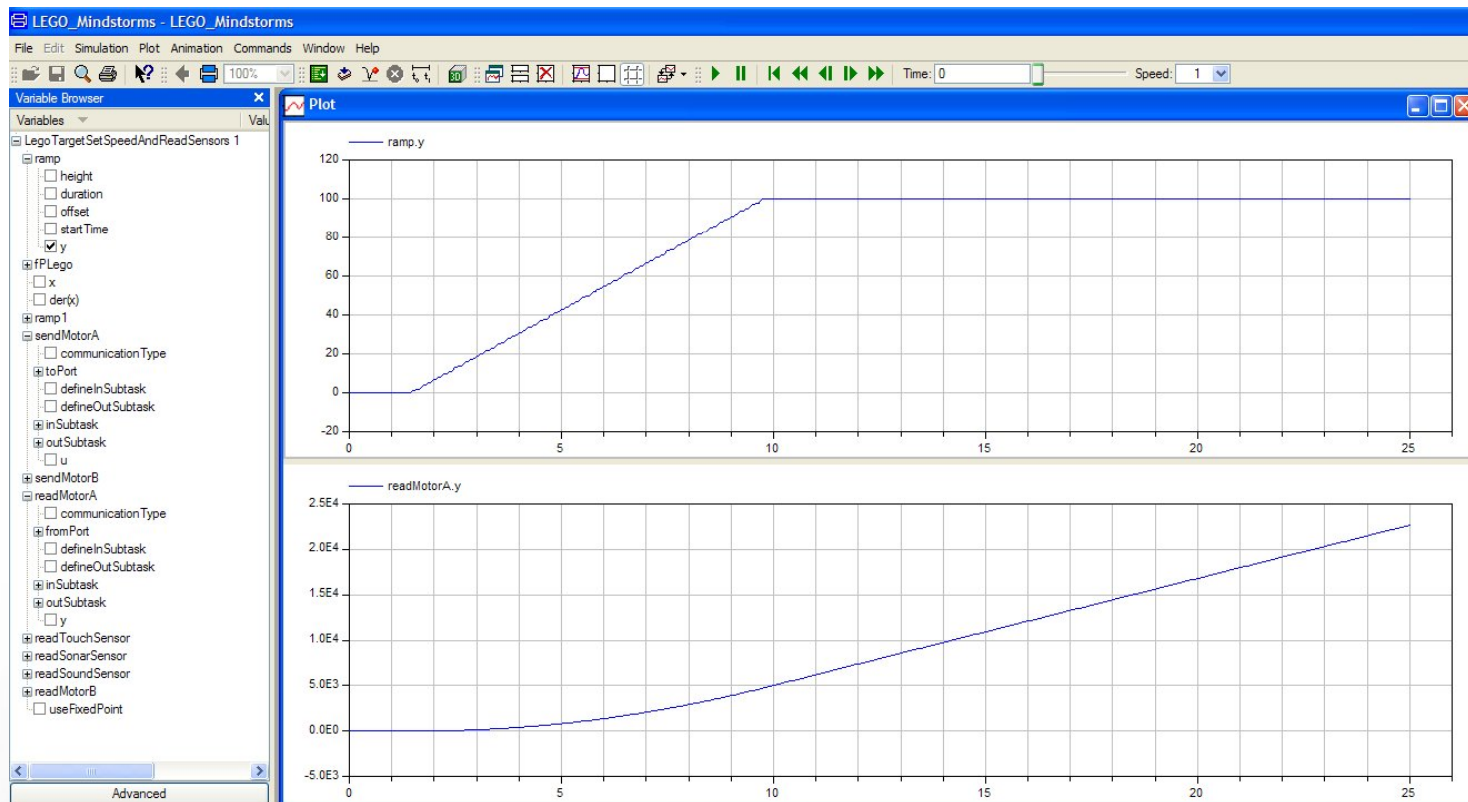
### Starting Lego and dymosim for Bluetooth communication and program execution



# Lego Mindstorms

## Setup/Configuration

- Results are automatically stored in Dymola





---

**Thank you for you attention  
and  
Good Luck with your projects**