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

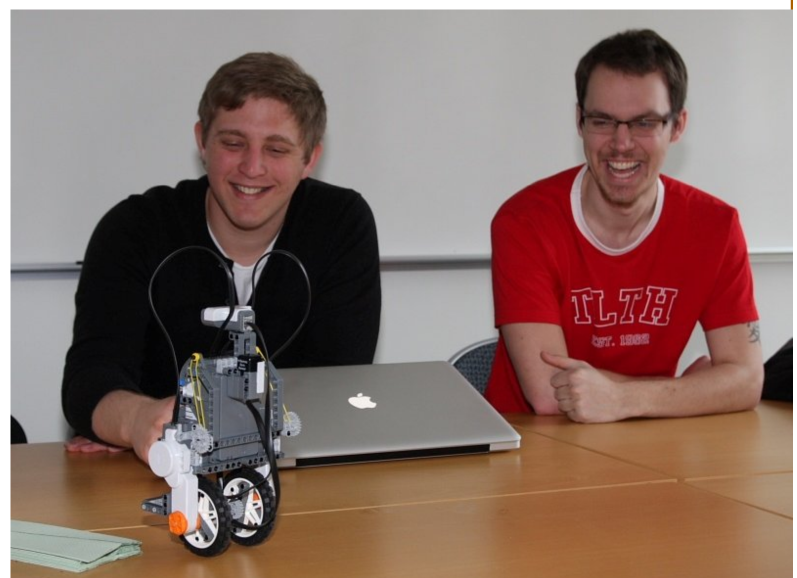
Johan Åkesson
Department of Automatic Control
Faculty of Engineering
Lund University

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



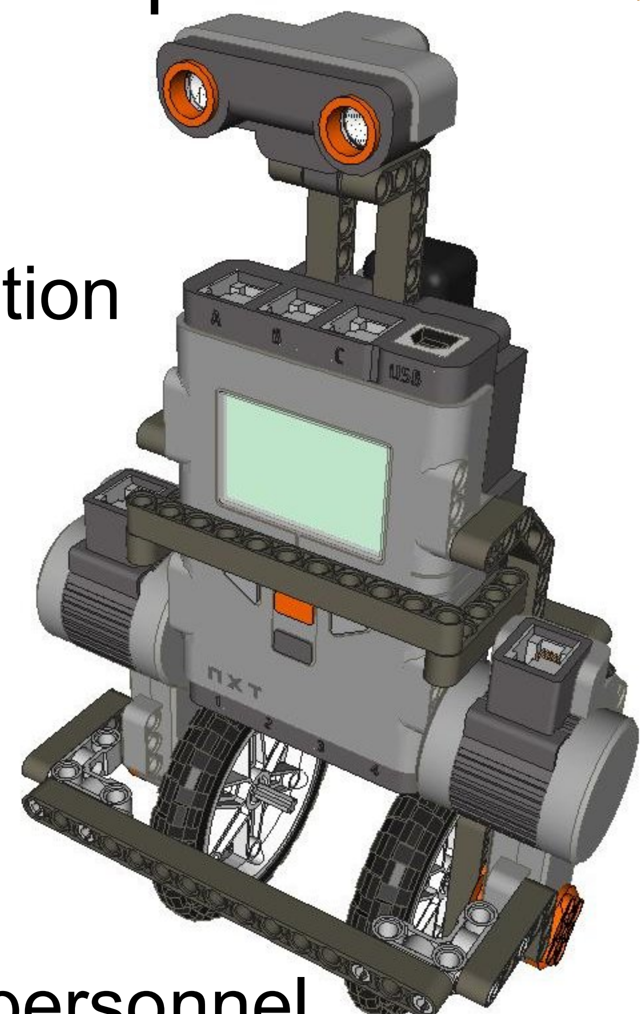
- Two groups working with Lego/Dymola
  - Five students in each group

Lego with Modelica/Dymola

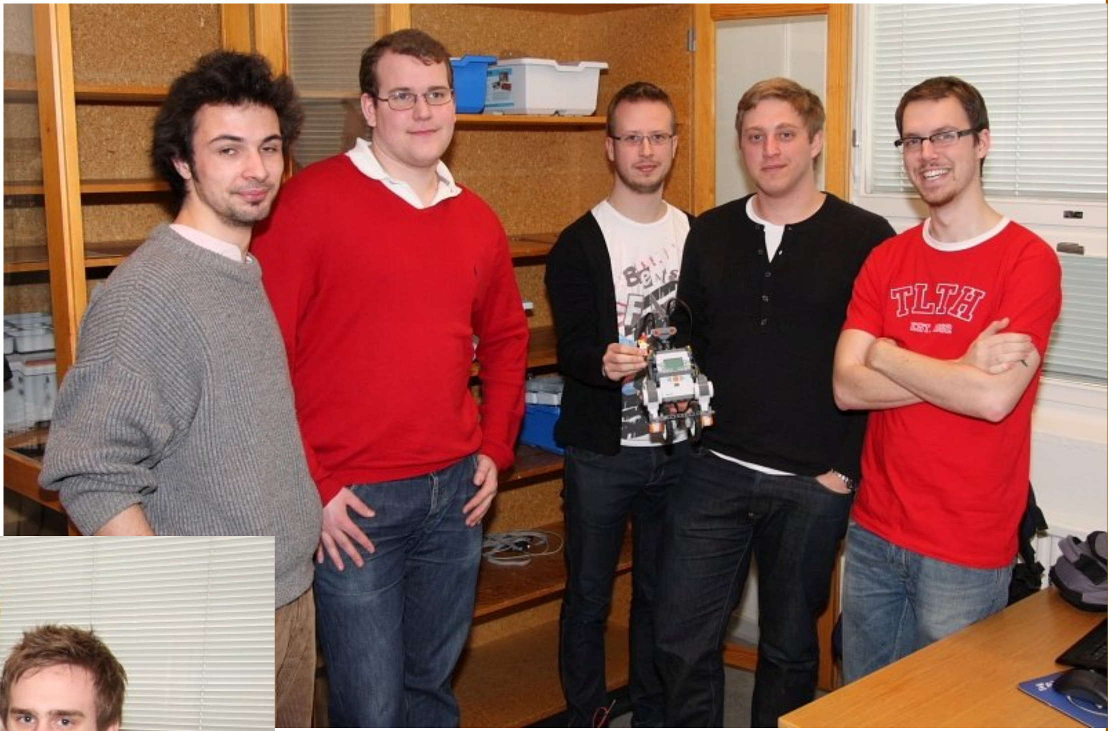
- Build the NXTway two-wheel robot
- Physical modeling with Modelica
  - Multi-body dynamics
- Model calibration
  - Experiments
  - Dymola Calibration module
- Control design
  - Derive simple model
  - Develop control scheme

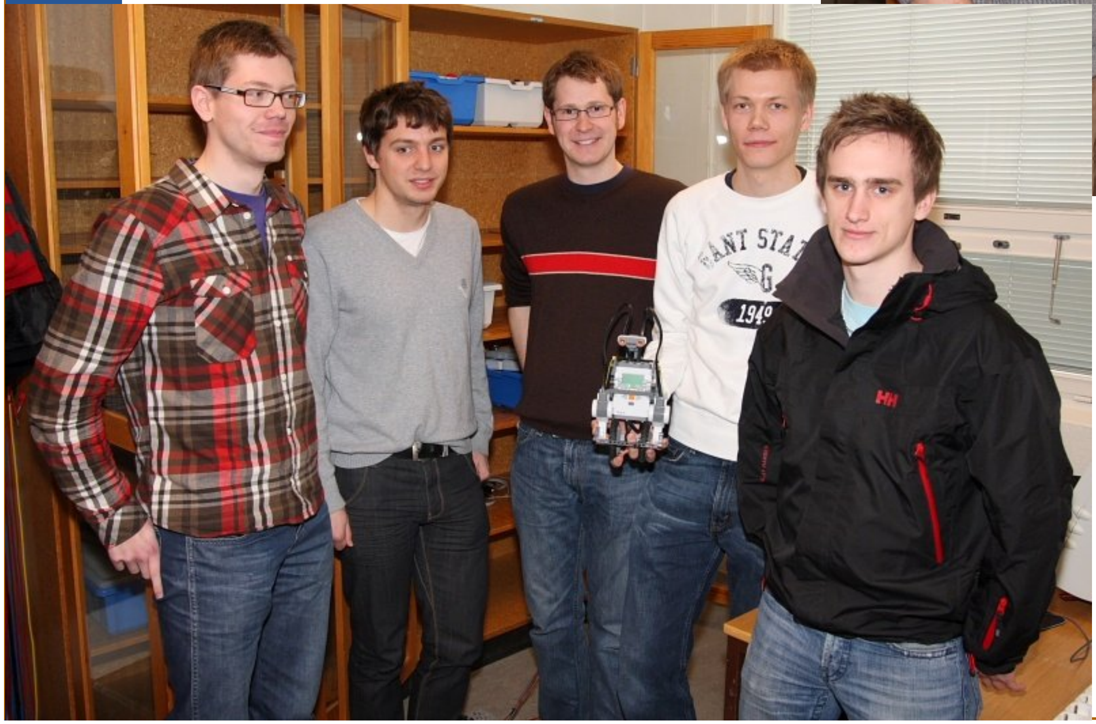
Lego with Modelica/Dymola

- 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 17th – group announcement
  - Friday March 20th – 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

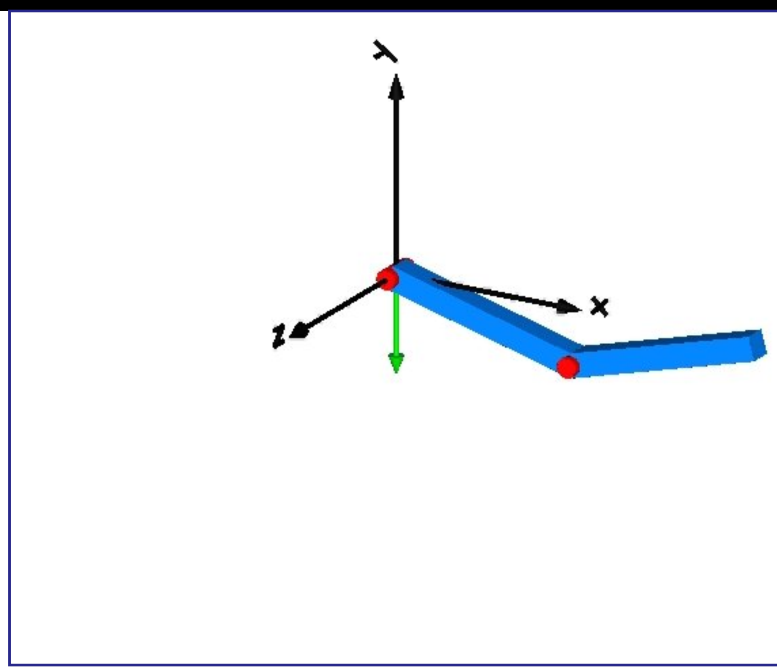
1. Introduction to Modelica (AC)
2. Multi-body modeling (Dynasim)
   - Wheel models (by Martin Otter)
3. Code generation with Dymola (Dynasim)
   - Modelica_Embedded

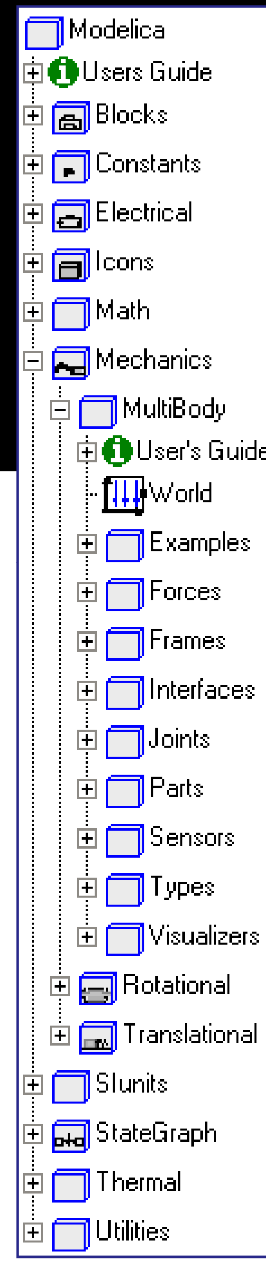
Modelica/Dymola Modeling

Hilding Elmqvist

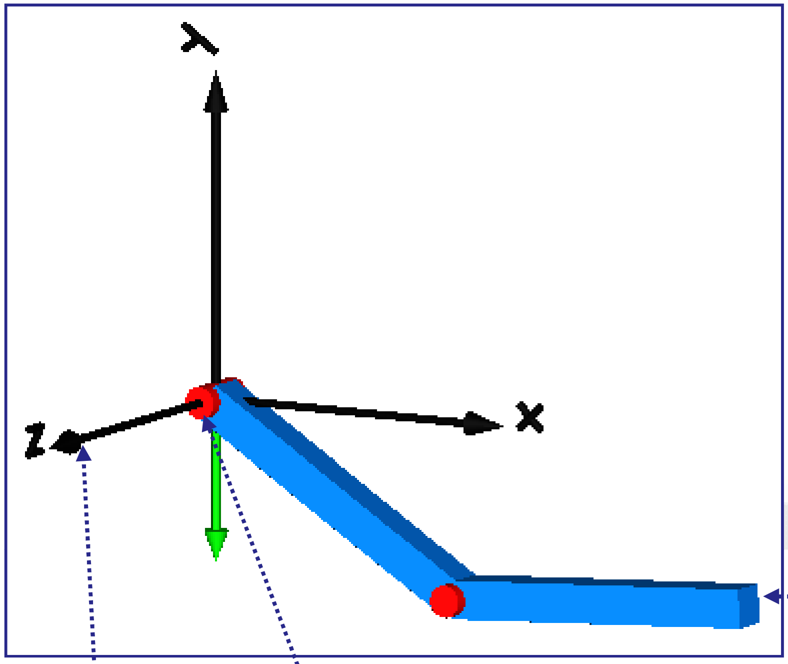
MultiBody modeling

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





Example – Double pendulum



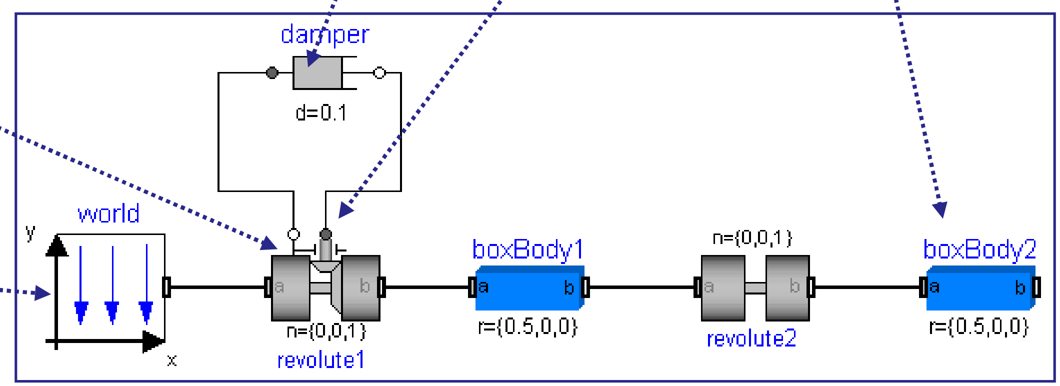
1D rotational damper

Bodies

1D rotational flange for drive

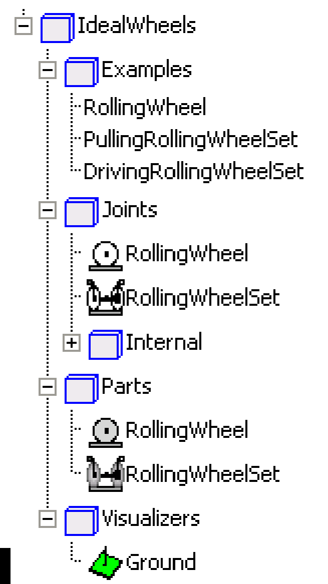
Rotational joint

World system (= Inertial system)



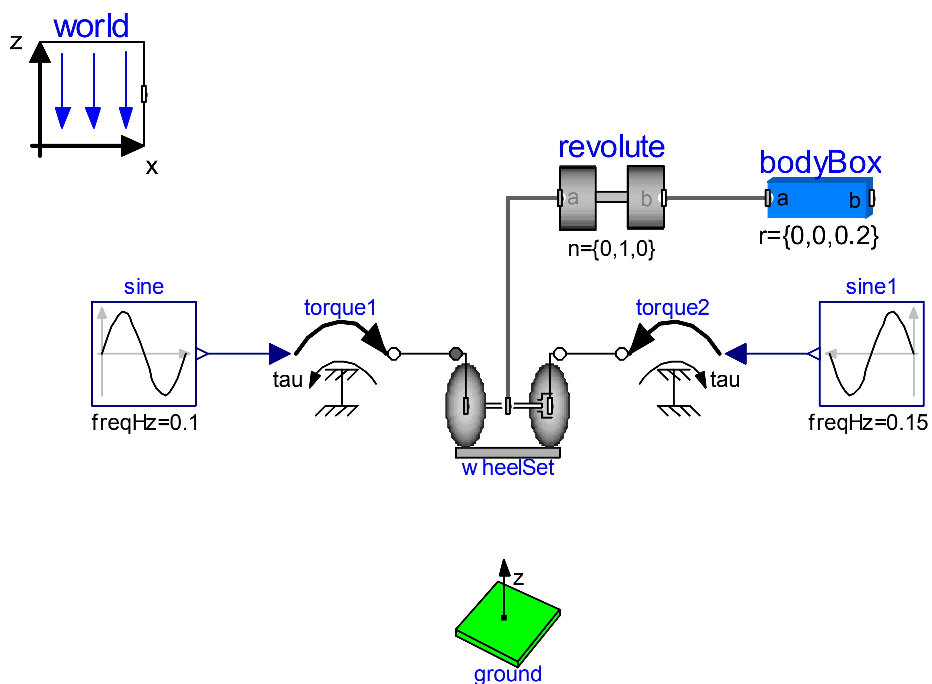
Wheel Model

- 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

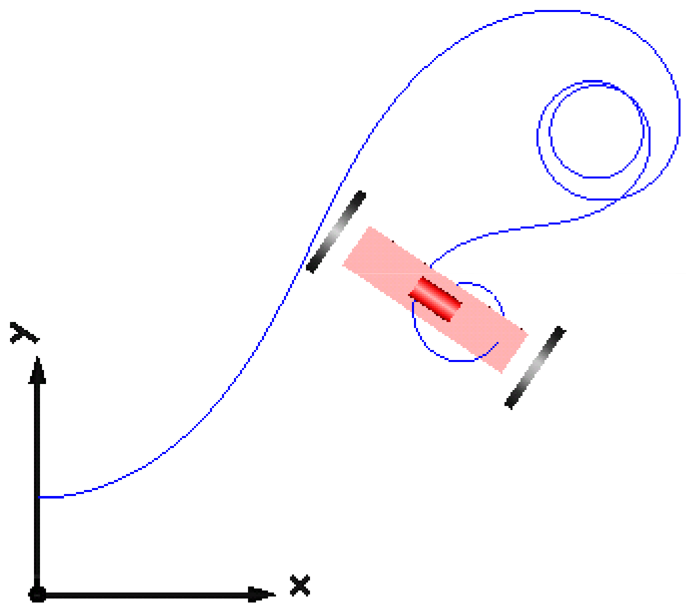


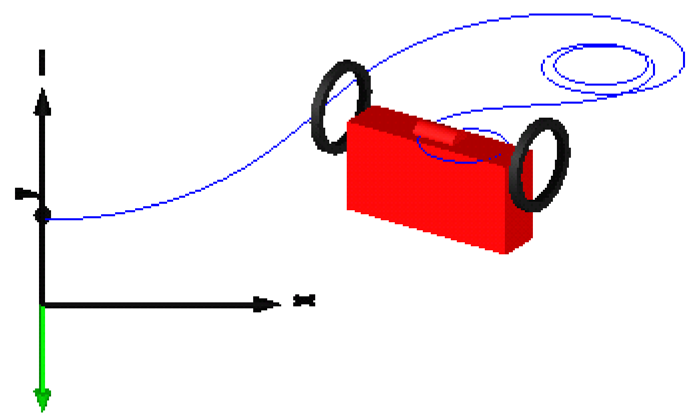
Body with two wheels

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

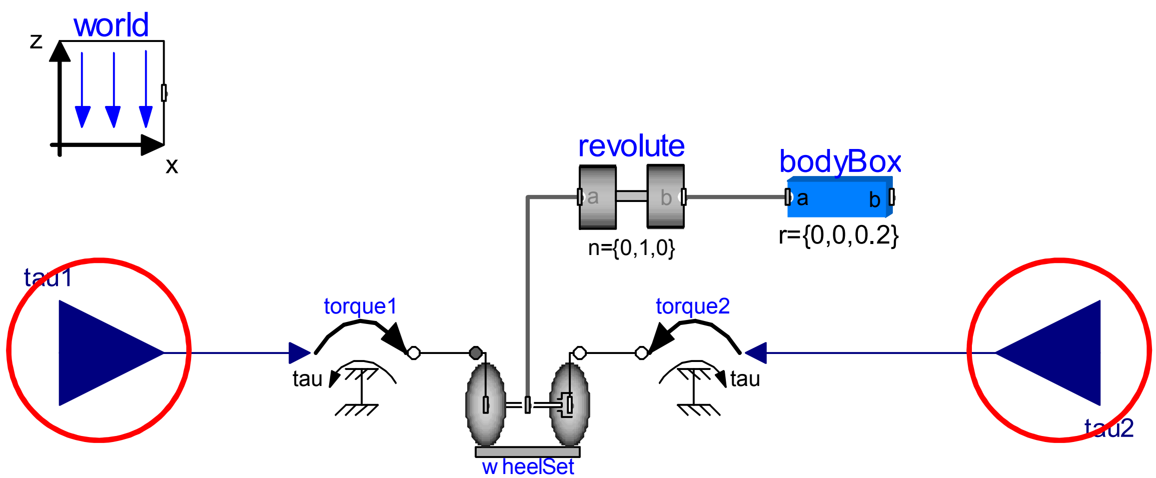


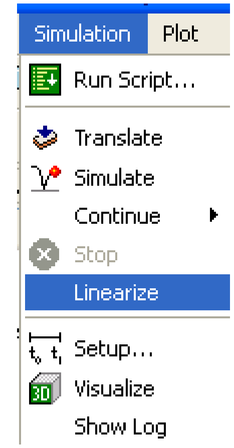
Resulting animations

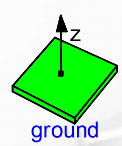




Linearization



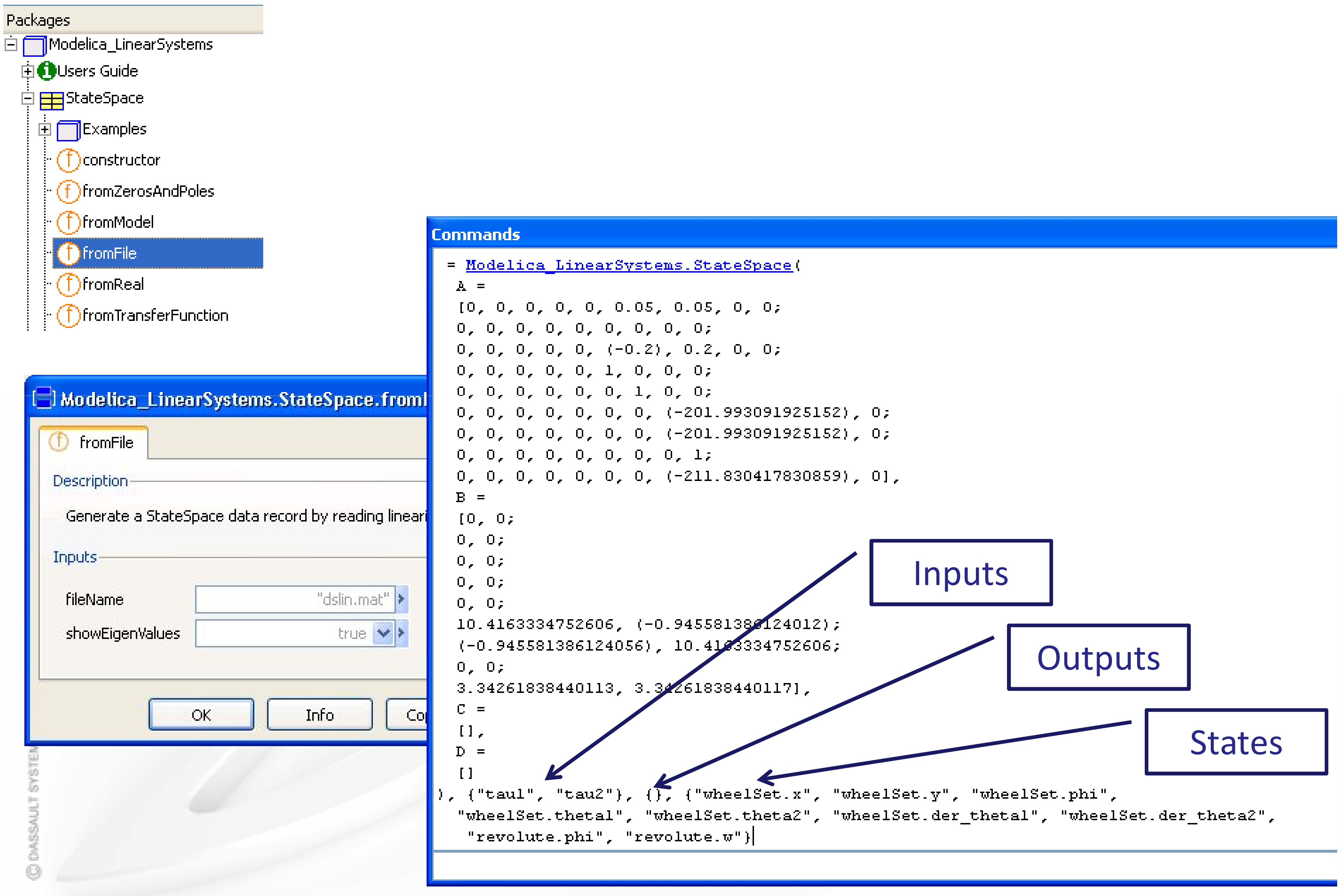




x_0

Variables	Values	Unit
x	0	m
der(x)		m/s
y	0.2	m
der(y)		m/s
phi	0	rad
der(phi)		rad/s
theta1	0	rad
der(theta1)		rad/s
theta2	0	rad
der(theta2)		rad/s
der_theta1	0	rad/s
der(der_theta1)		rad/s/s
der_theta2	0	rad/s
der(der_theta2)		rad/s/s

Linear model

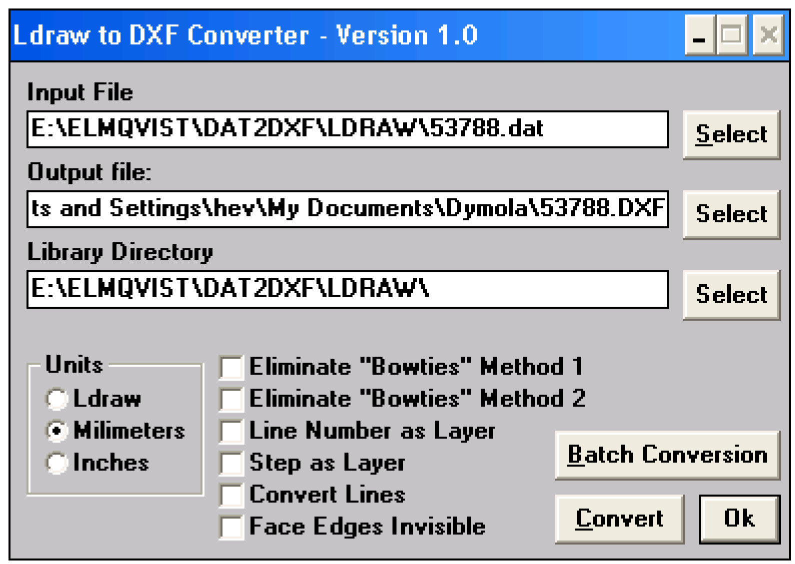


Animation with LEGO shapes

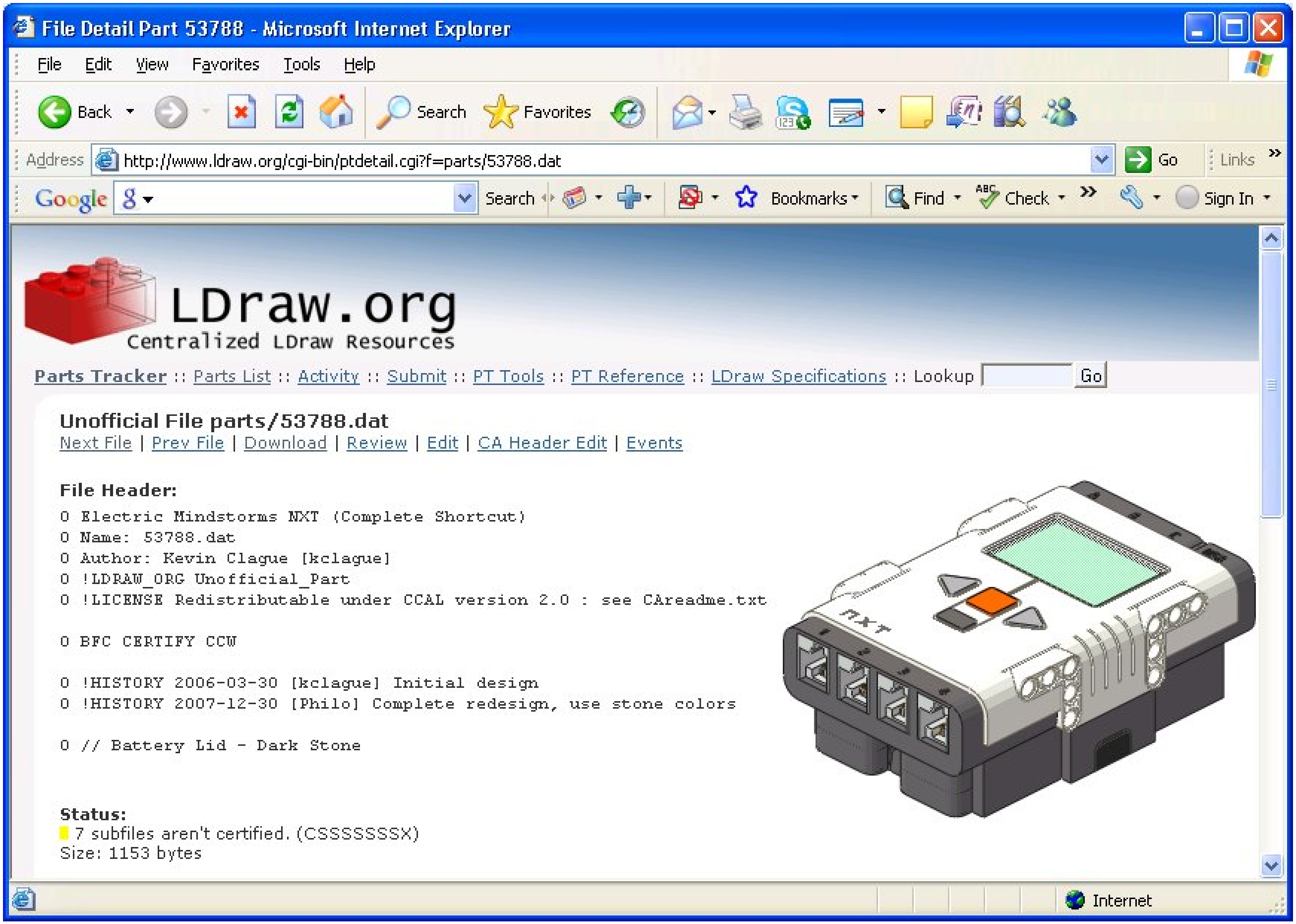
- **Dymola supports vizualization of shapes described by DXF, STL and VRML (subset) files.**
- **LEGO building blocks are described in .DAT format:**
  - http://www.ldraw.org
- **DAT2DXF converter available:**
  - http://www.ldraw.org/Downloads-req-viewdownload-cid-6.html
- **LEGO parts are available 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

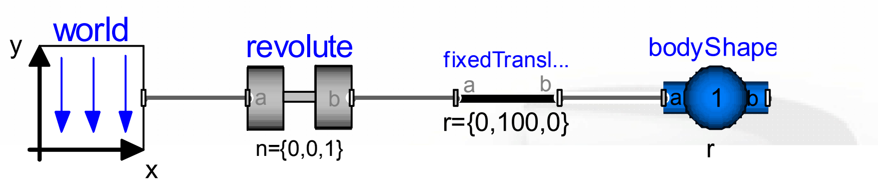
- **Download parts directories from:**
  - http://www.ldraw.org/Downloads-req-viewdownload-cid-1.html
- **Mindstorms parts are among the unofficial parts**
- **See dat2dxf\readme.txt for instructions on how to organize files**

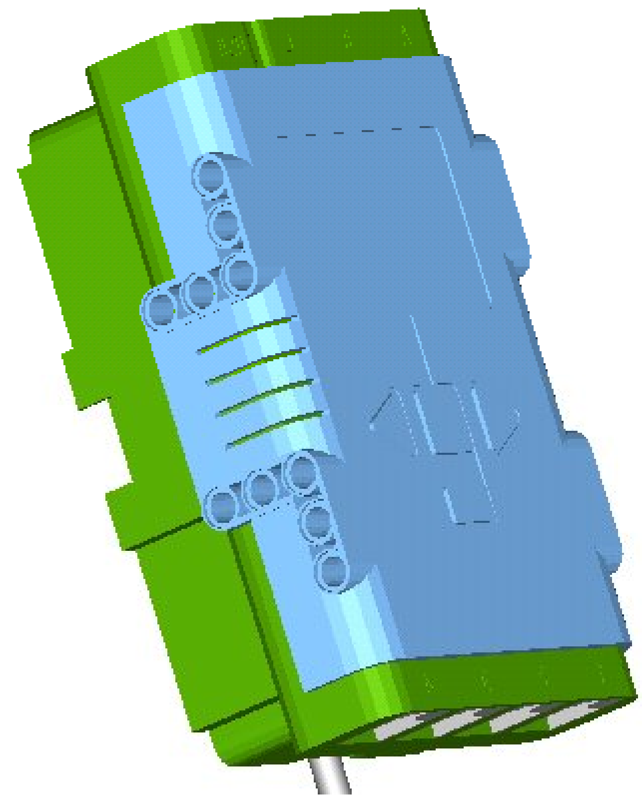


LEGO blocks resource



Dymola animation of NXT (shape 53788)

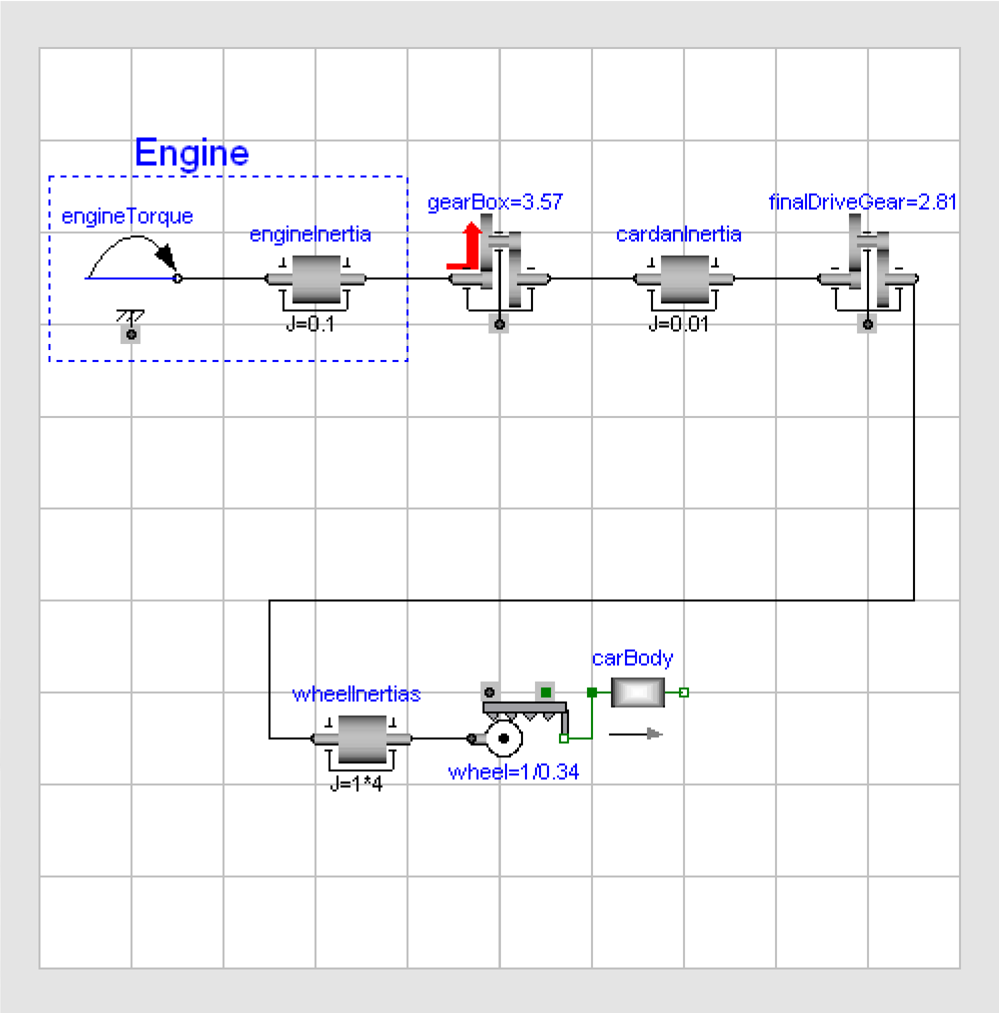




Tutorial
Modelica and Dymola for System Design
Model Building and Calibration

H. Elmqvist
Dynasim AB, Lund

Calibrate engine map parameter and friction



Modelica_EmbeddedSystems and Code generation for Lego Mindstorms NXT

Ulf Nordström

DASSAULT SYSTEMES

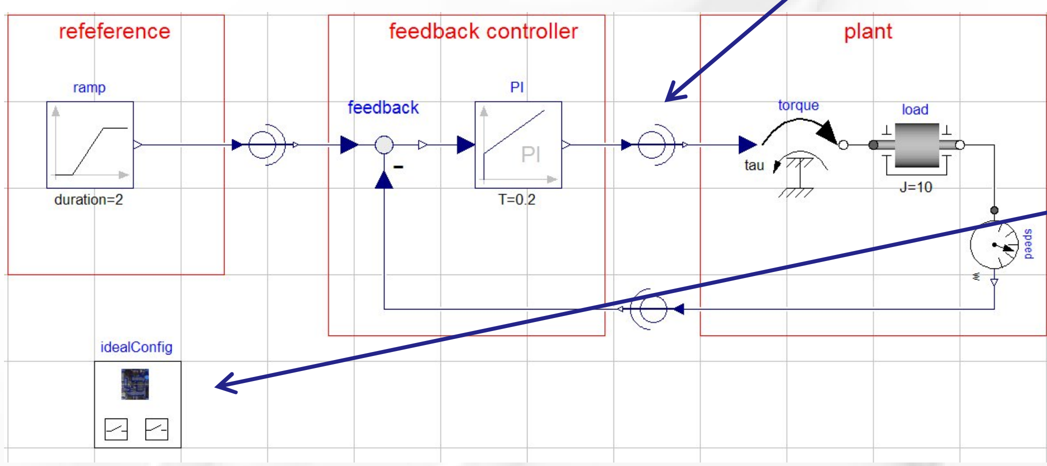
Contents

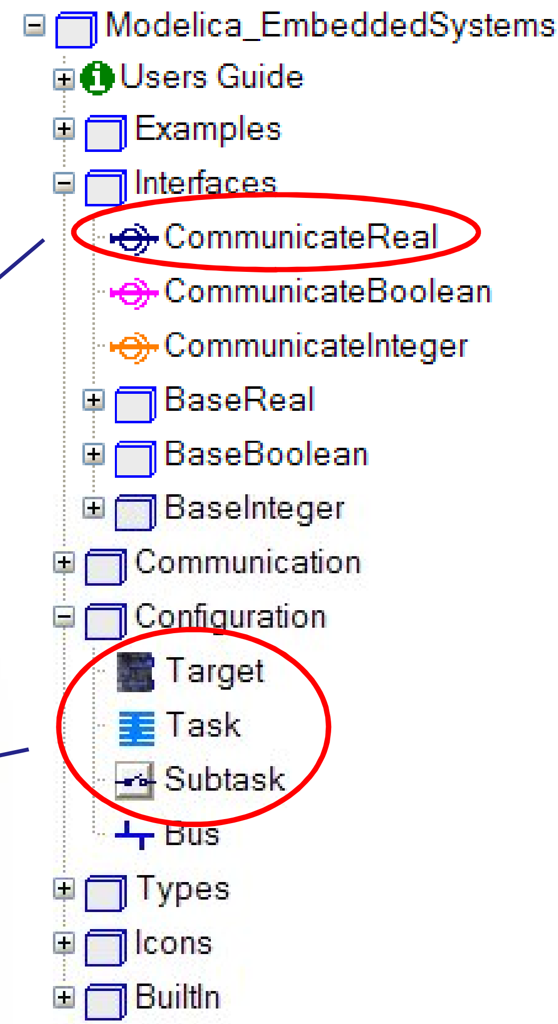
- Overview
- Modelica_EmbeddedSystems
- Fixed point Code Generation
- Lego Mindstorms

Modelica_EmbeddedSystems

Key Components

- **Communication**
- **Configuration**
  - **Target**
  - **Task**
  - **Subtask**





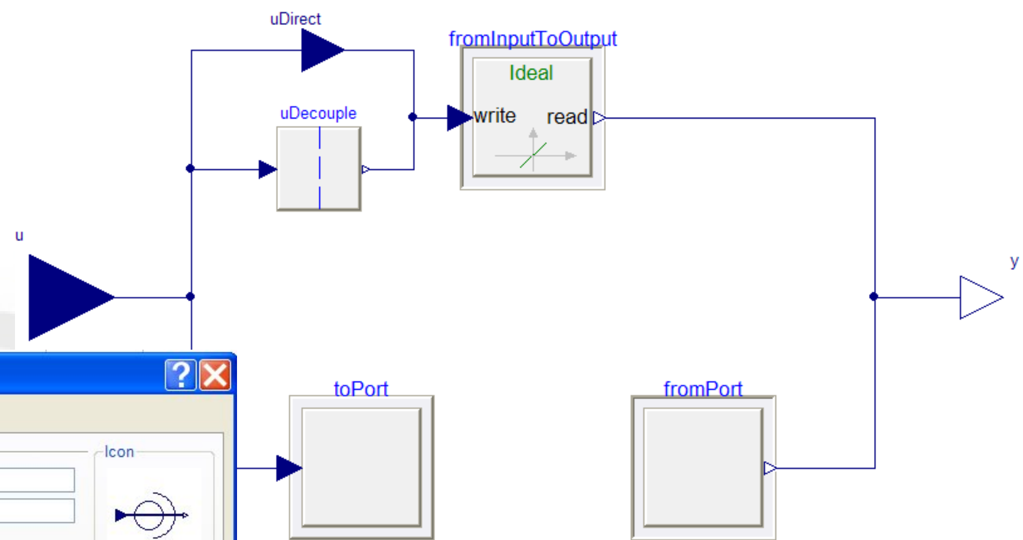
Modelica_EmbeddedSystems

Key Components

Communication

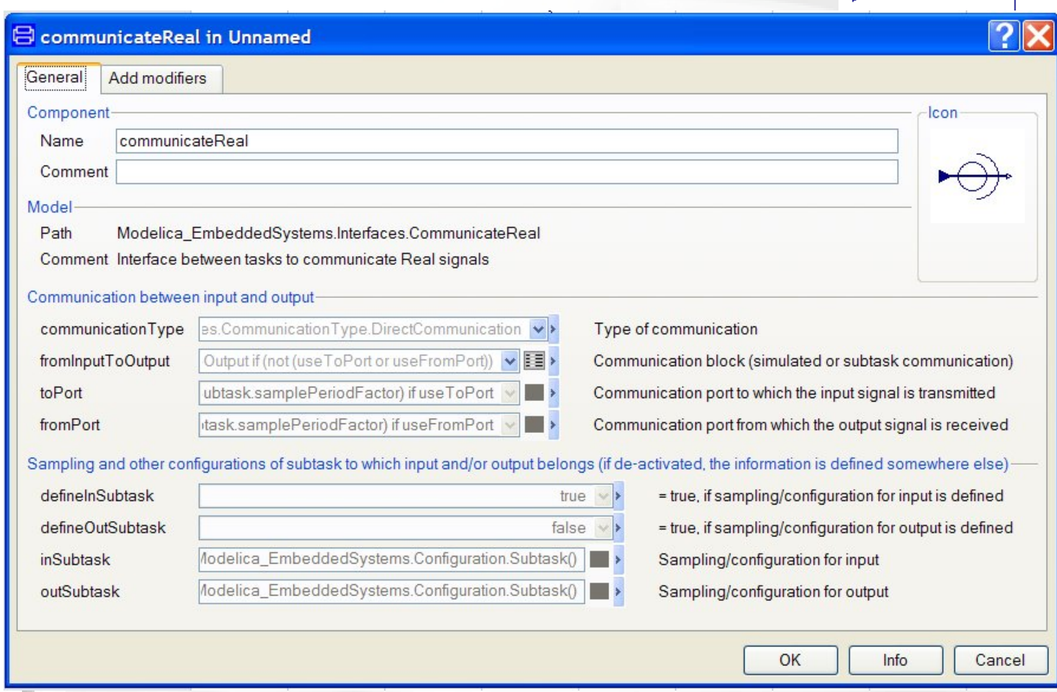
Input and output reside in the same base-rate task

decoupleReal specifies boundary between subtasks



Communication using ports

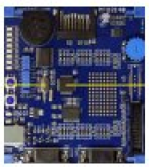
Inter-task communication or using external I/O



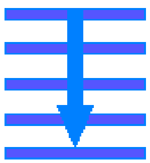
Modelica_EmbeddedSystems

Key Components

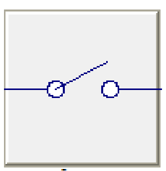
Configuration



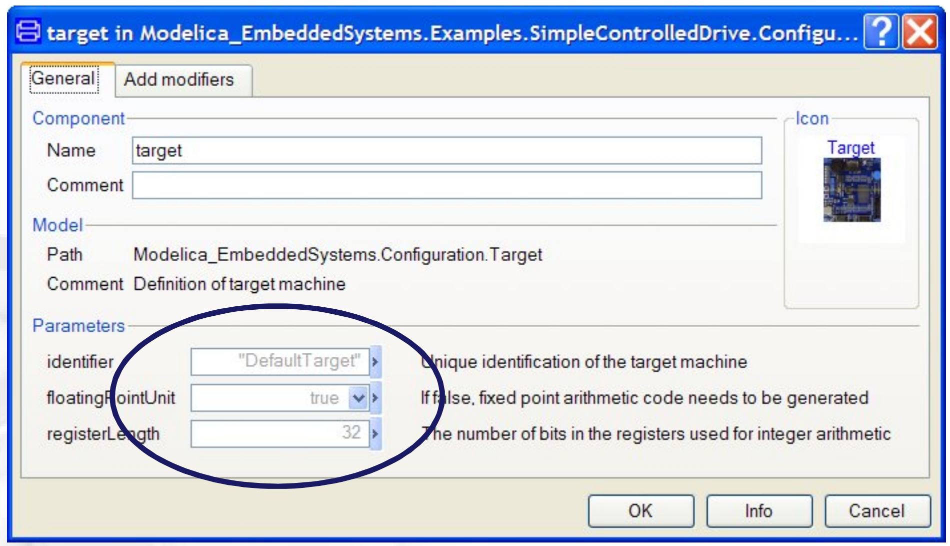
Target



Task



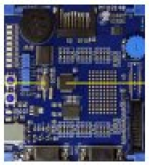
Subtask



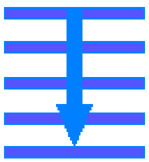
Modelica_EmbeddedSystems

Key Components

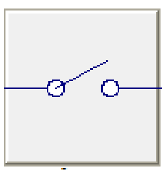
Configuration



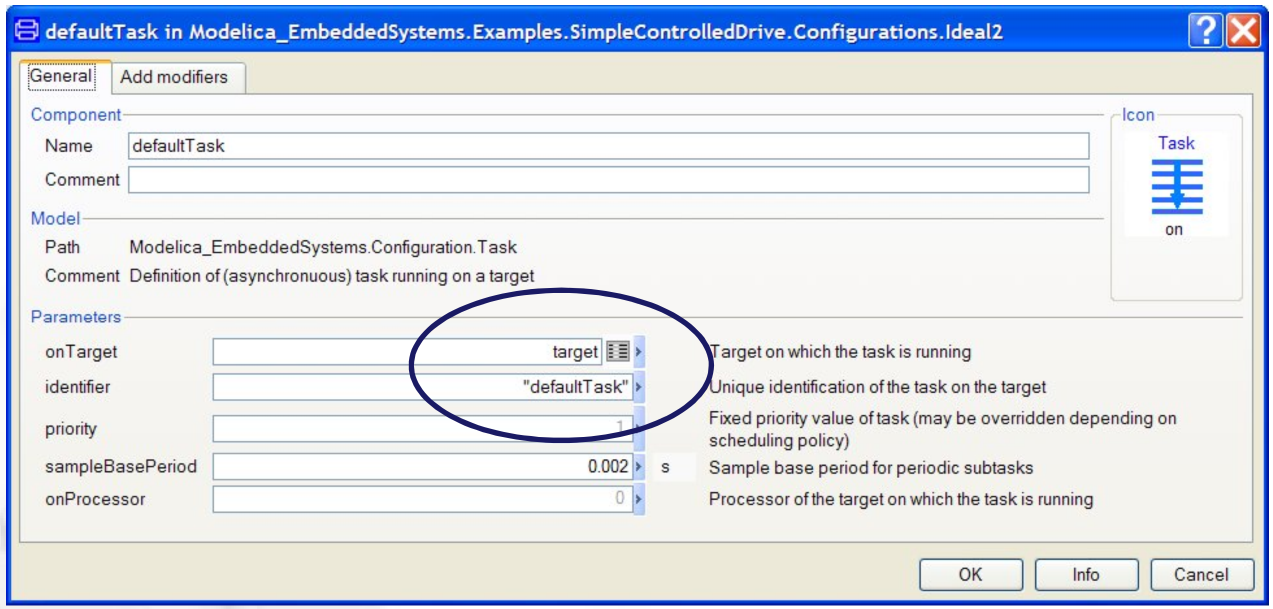
Target



Task



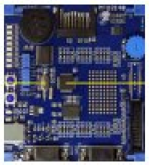
Subtask



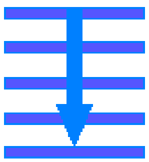
Modelica_EmbeddedSystems

Key Components

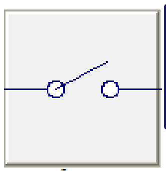
Configuration



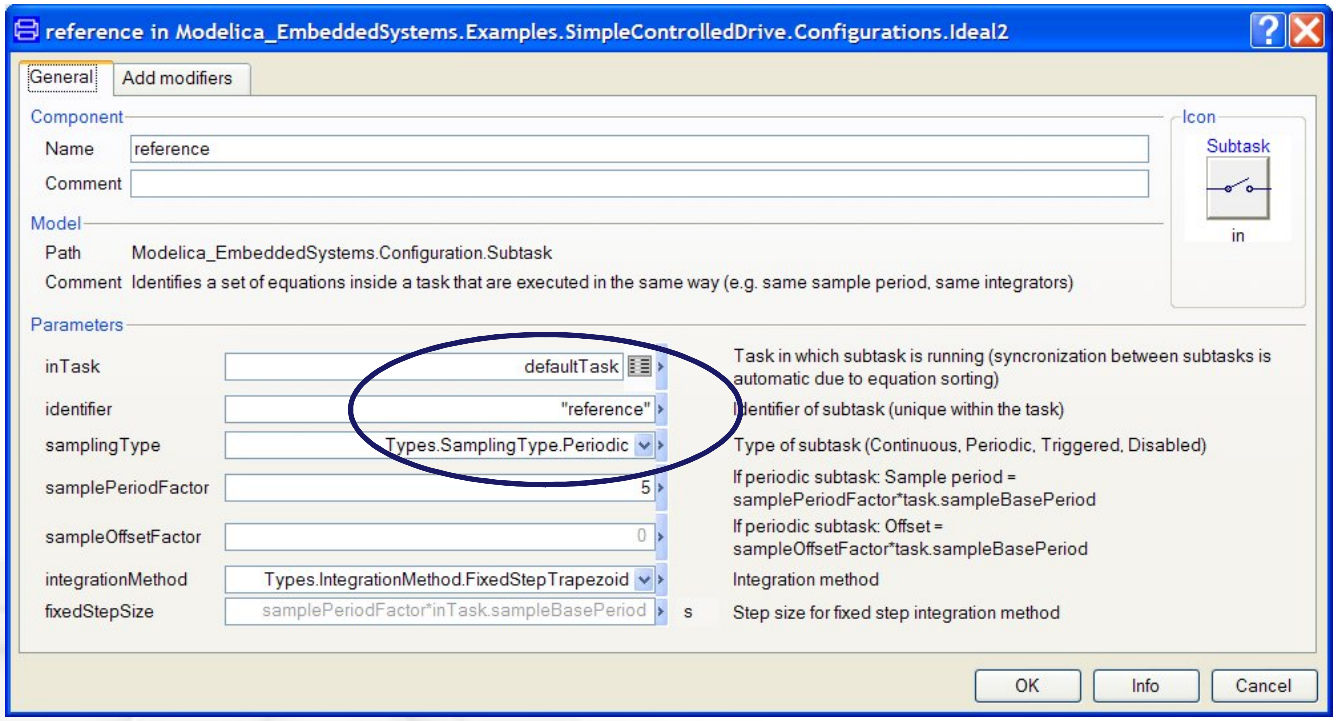
Target



Task



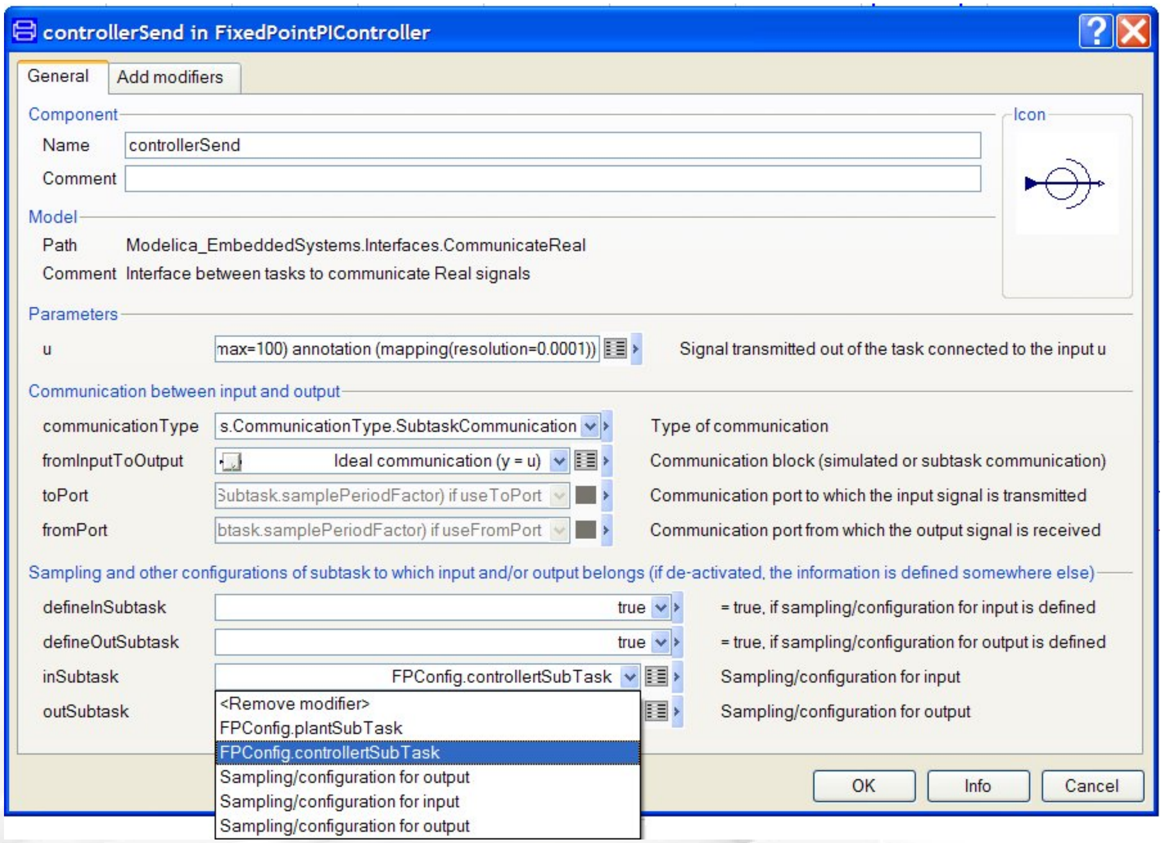
Subtask



Modelica_EmbeddedSystems

Key Components

Use pull-down menue to select target/task/subtask"



Modelica_EmbeddedSystems

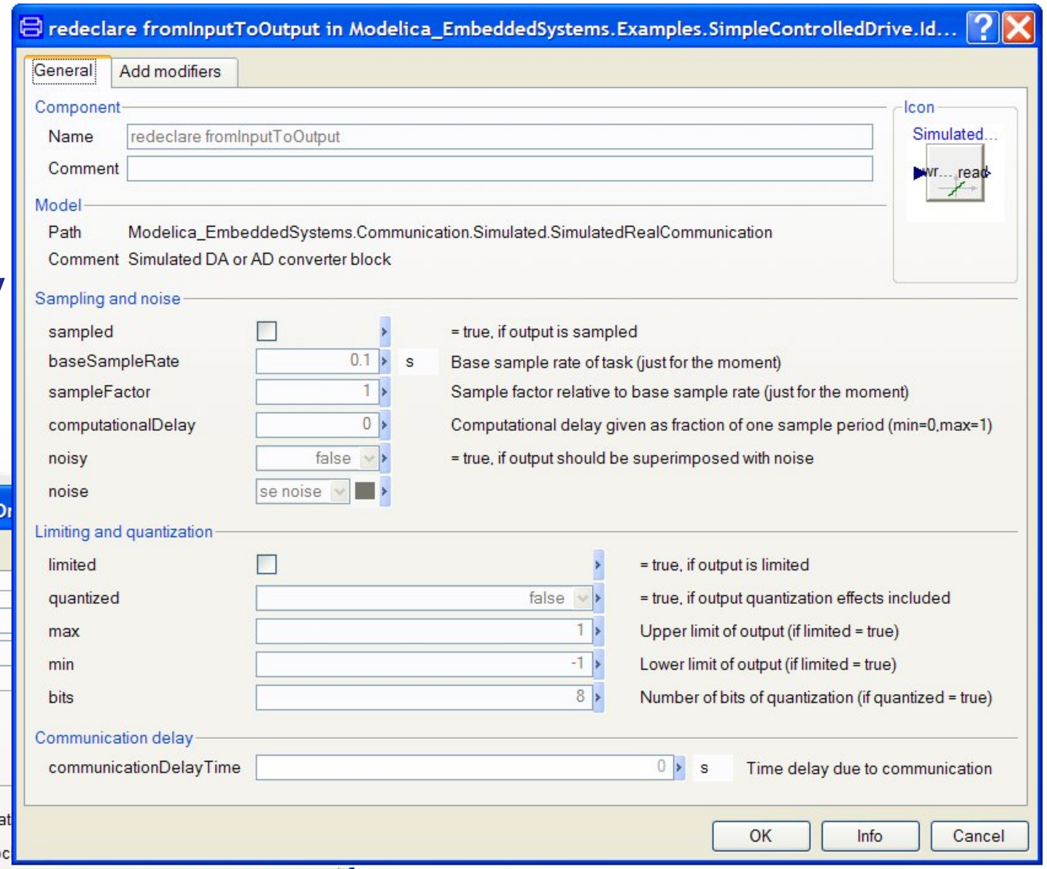
Scenarios

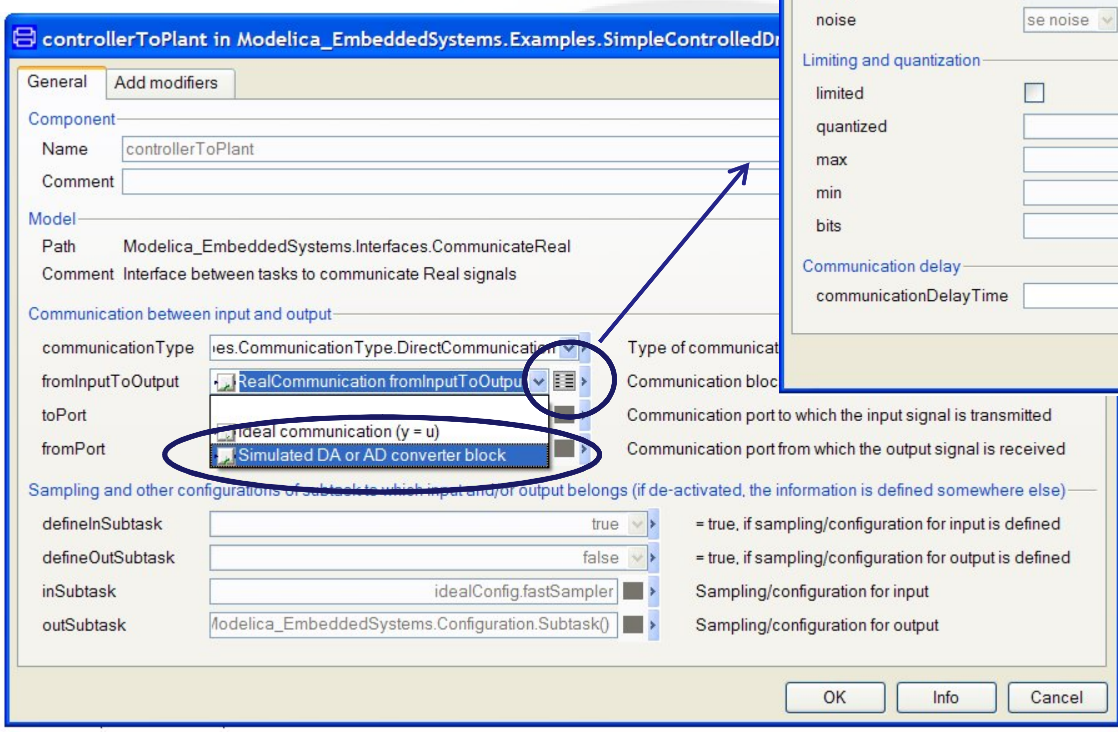
- **Model In the Loop**
  - Quantization
  - Communication delay
- **Software In the Loop**
  - Task decomposition
  - Effects of fixed point arithmetics
- **Embedded**

Modelica_EmbeddedSystems

Scenarios

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



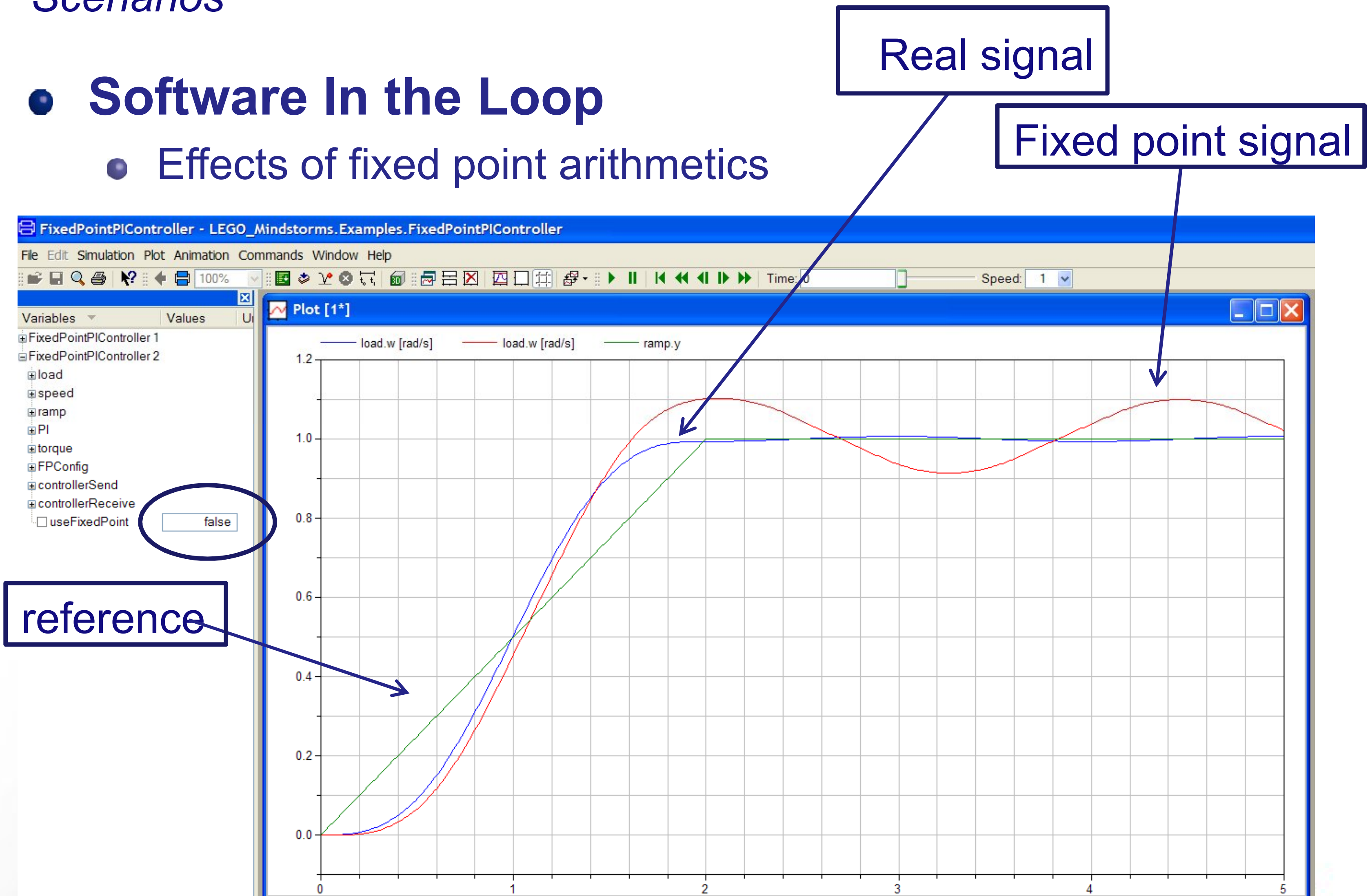


DASSAULT SYSTEMES

Modelica_EmbeddedSystems

Scenarios

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



DASSAULT SYSTEMES

Modelica_EmbeddedSystems

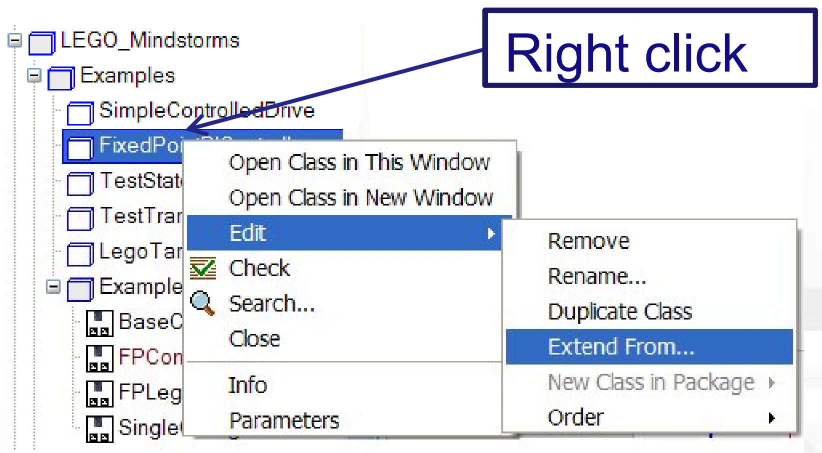
Scenarios

- **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



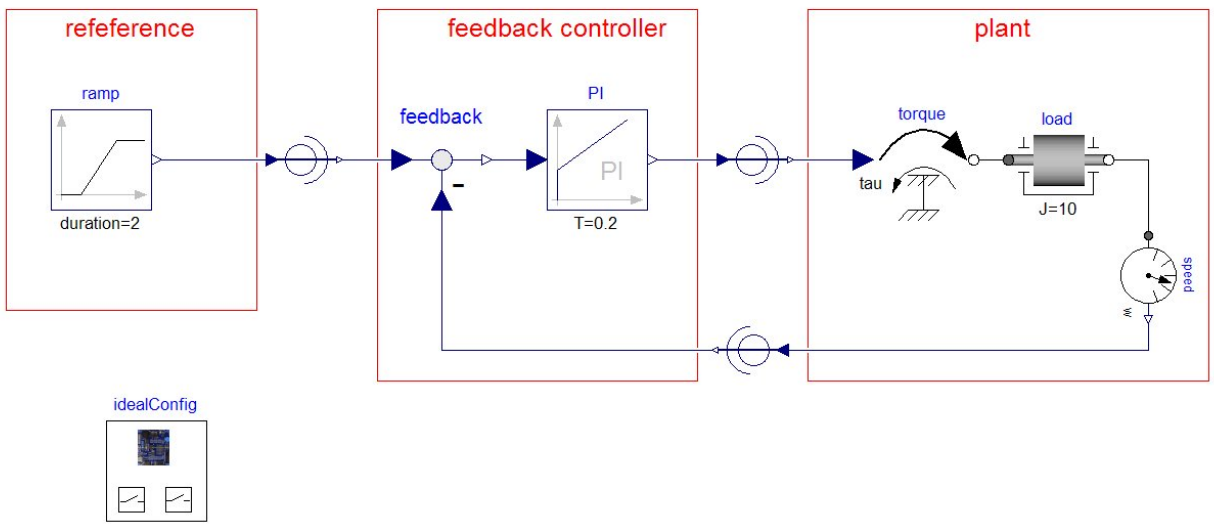
Modelica_EmbeddedSystems

Example – Software In the Loop

Set Hidden.DecomposeInTasks = true

Task decomposition

- Reference subtask
- Feedback subtask
- Plant subtask

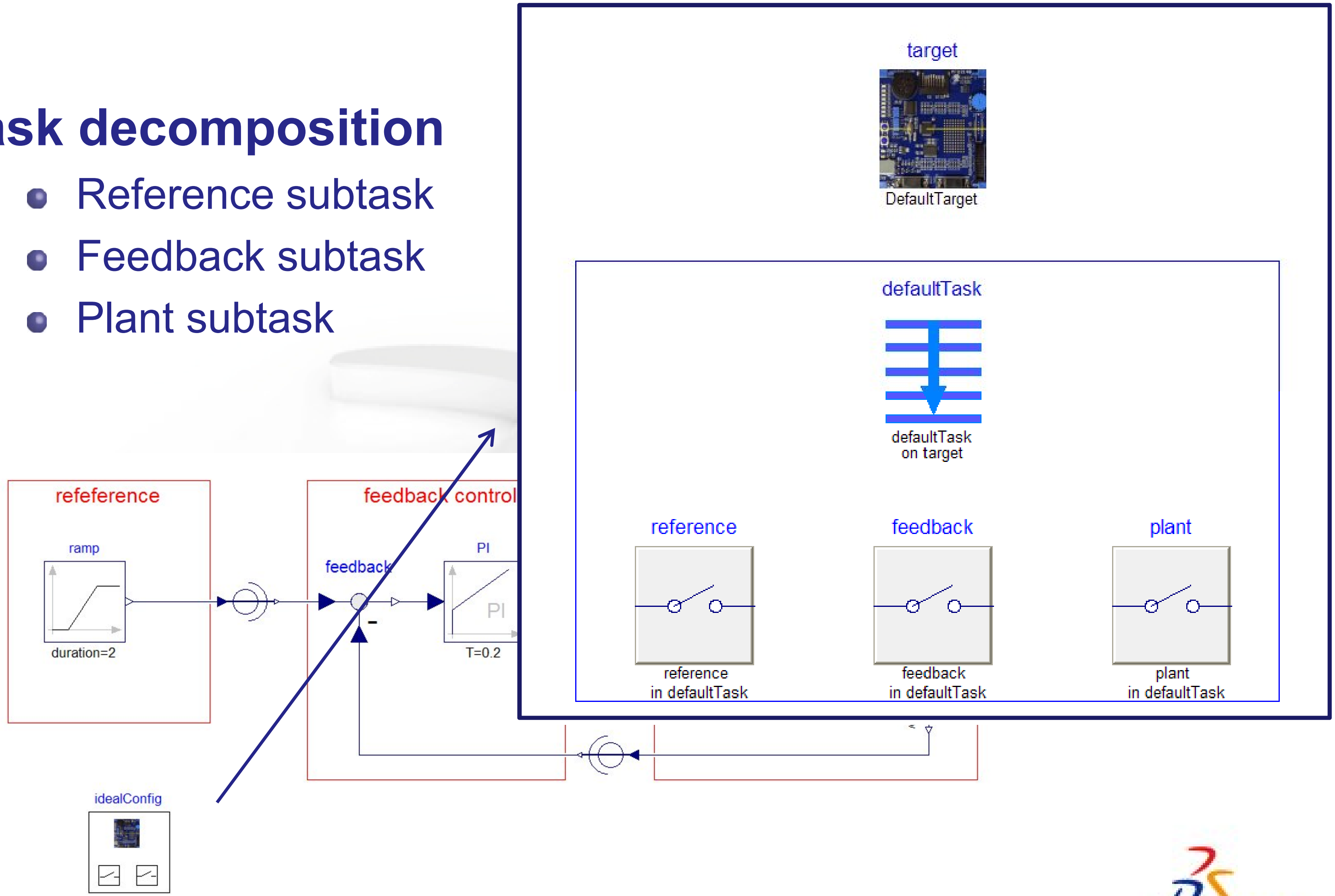


Modelica_EmbeddedSystems

Example – Software In the Loop

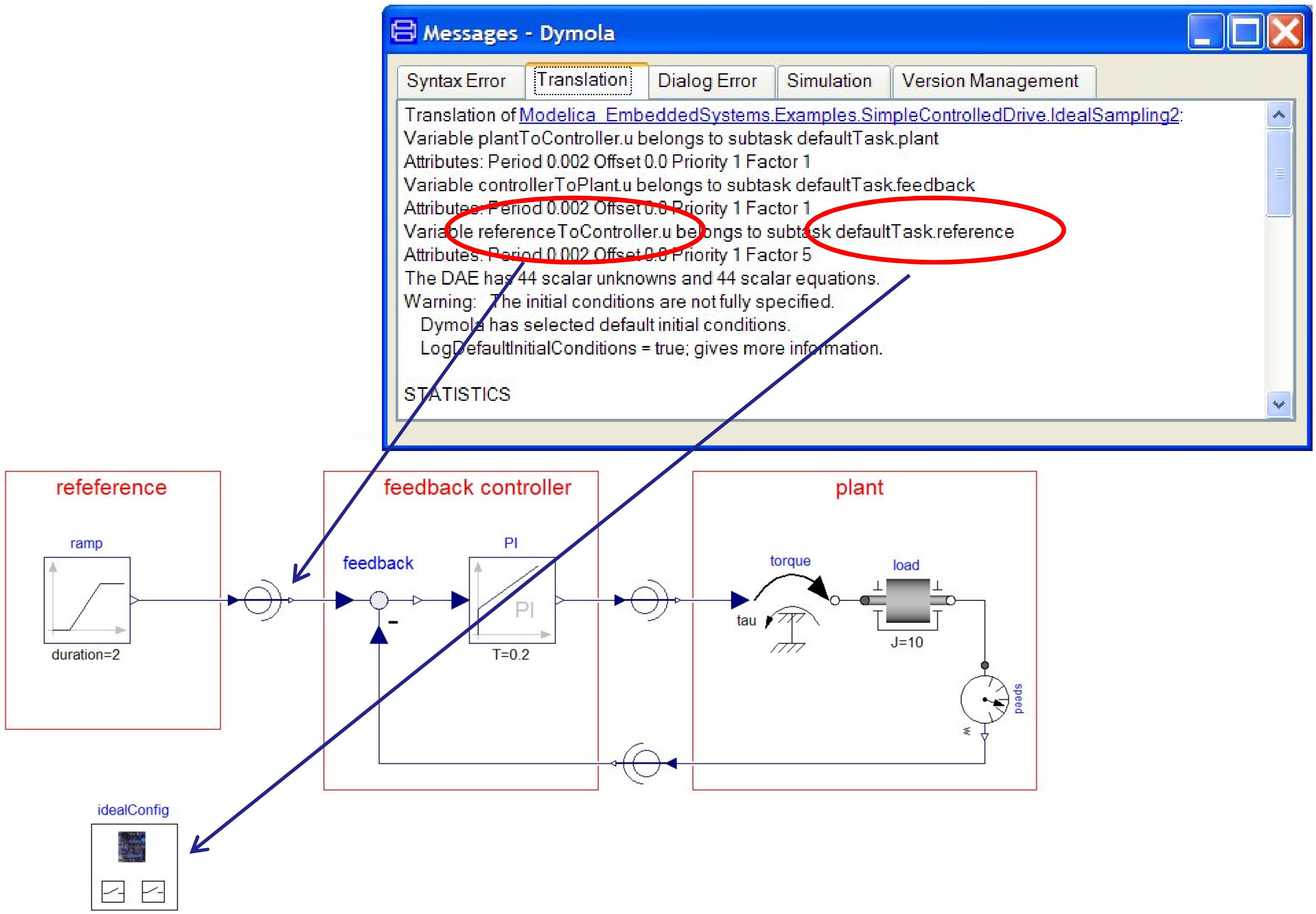
Task decomposition

- Reference subtask
- Feedback subtask
- Plant subtask



Modelica_EmbeddedSystems

Example – Software In the Loop



FixedPoint Code Generation

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

DASSAULT SYSTEMES

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]

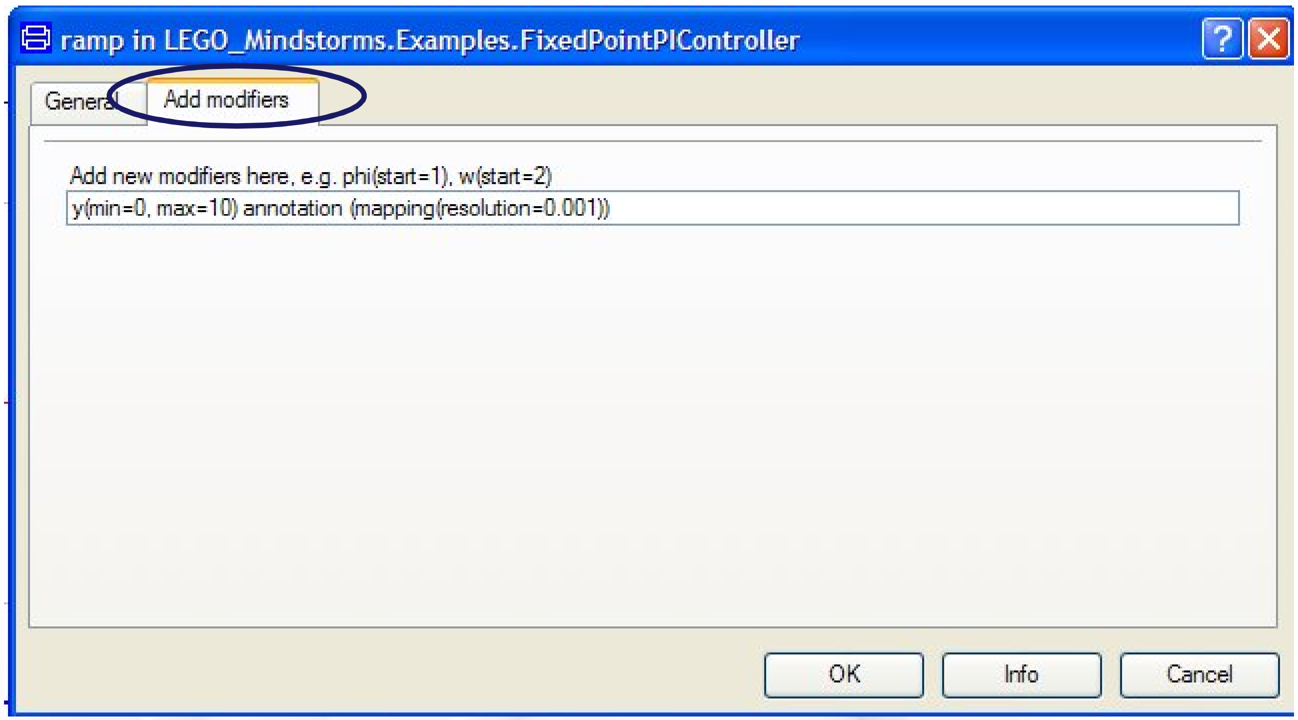
Generated code

```
/* 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

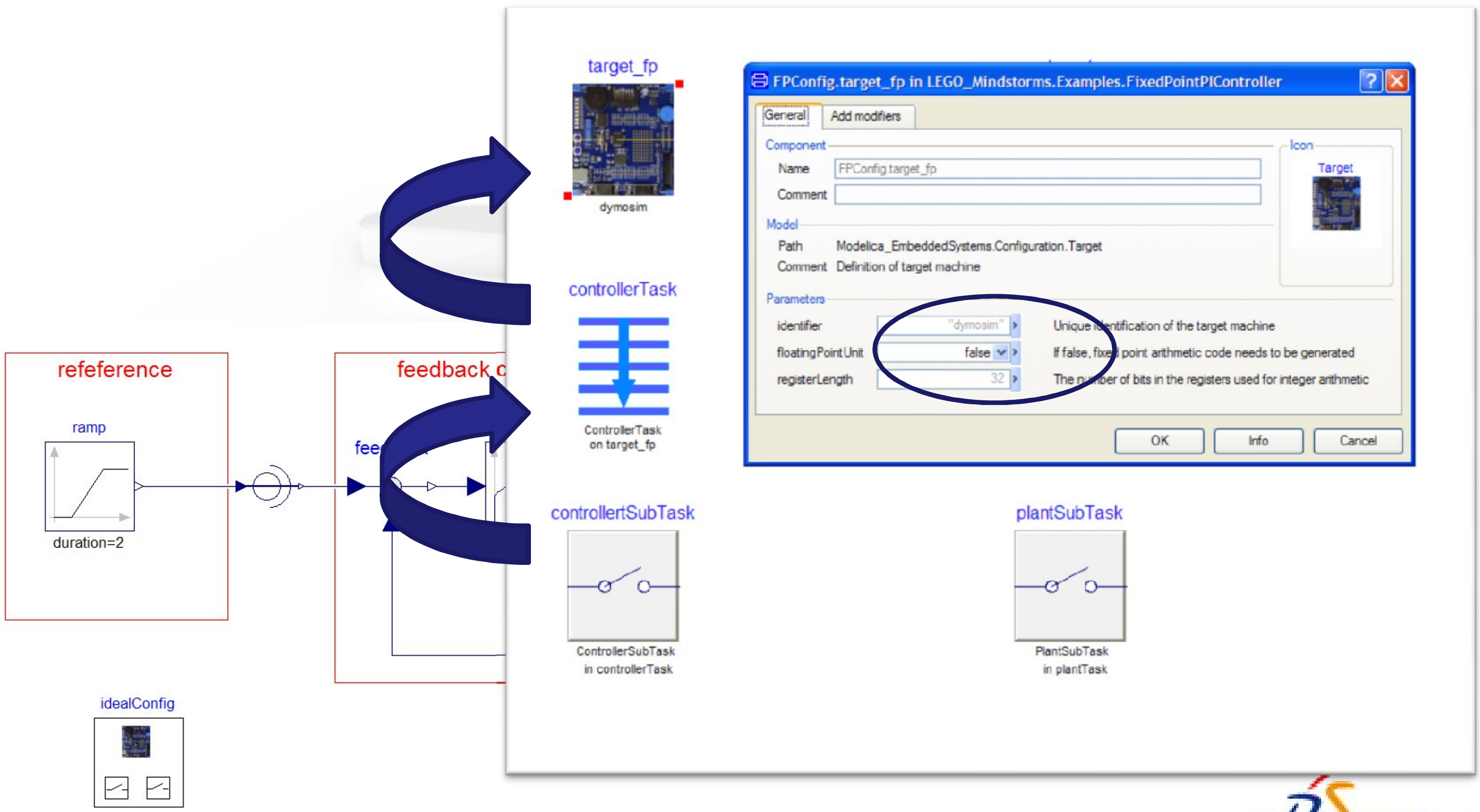
Use parameter dialog to input annotations as modifiers



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

declarations.c

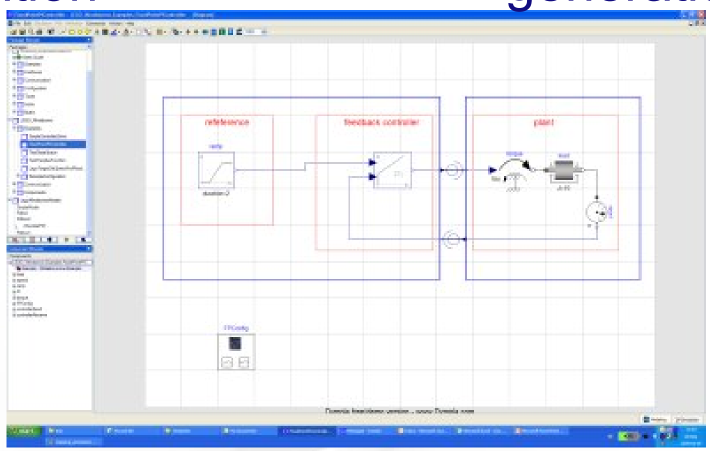
equations.c

Code generation

+

dymola_wrapper.c

build



Code generation

dsmodel.c

Store and plot

Send internal variables over bluetooth

dymosim

Lego Mindstorms

Framework – dymola_wrapper.c

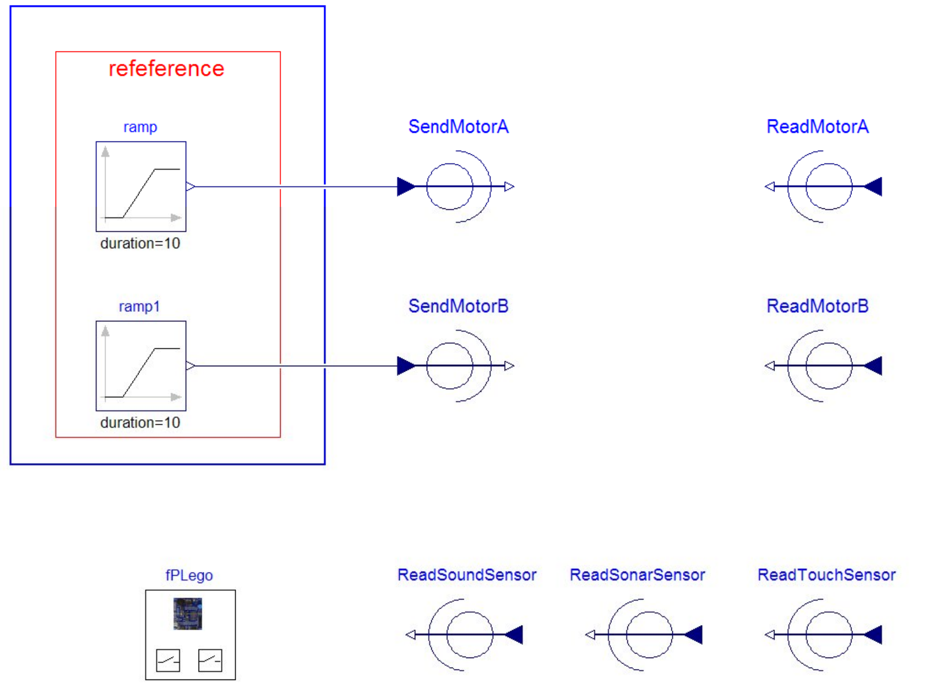
```
#include "kernel.h"
#include "kernel_id.h"
#include "ecrobot_interface.h"
#include "target_port.h"
/* OSEK declarations */
DeclareCounter(SysTimerCnt);
DeclareTask(Task1);
/* Include fixedpoint variable declarations */
#include "declarations.c"
...
/* Task1 executed every 50msec */
TASK(Task1)
{
    /* 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

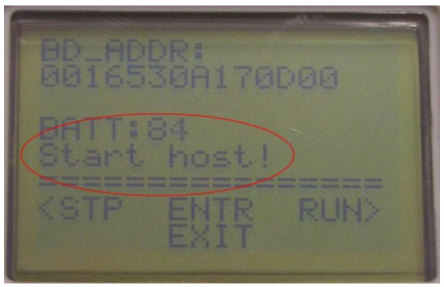
- CD to ”**..\nxtOSEK\samples\dymola**”
- Compile with ”**make all**”
- Start Lego and download executable with ”**./ramboot.sh**”

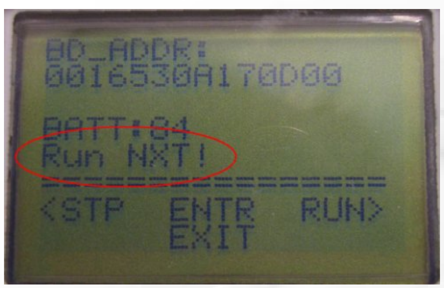
…

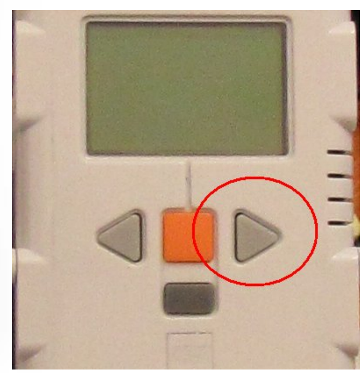
Lego Mindstorms

Setup/Configuration

Starting Lego and dymosim for Bluetooth communication and program execution



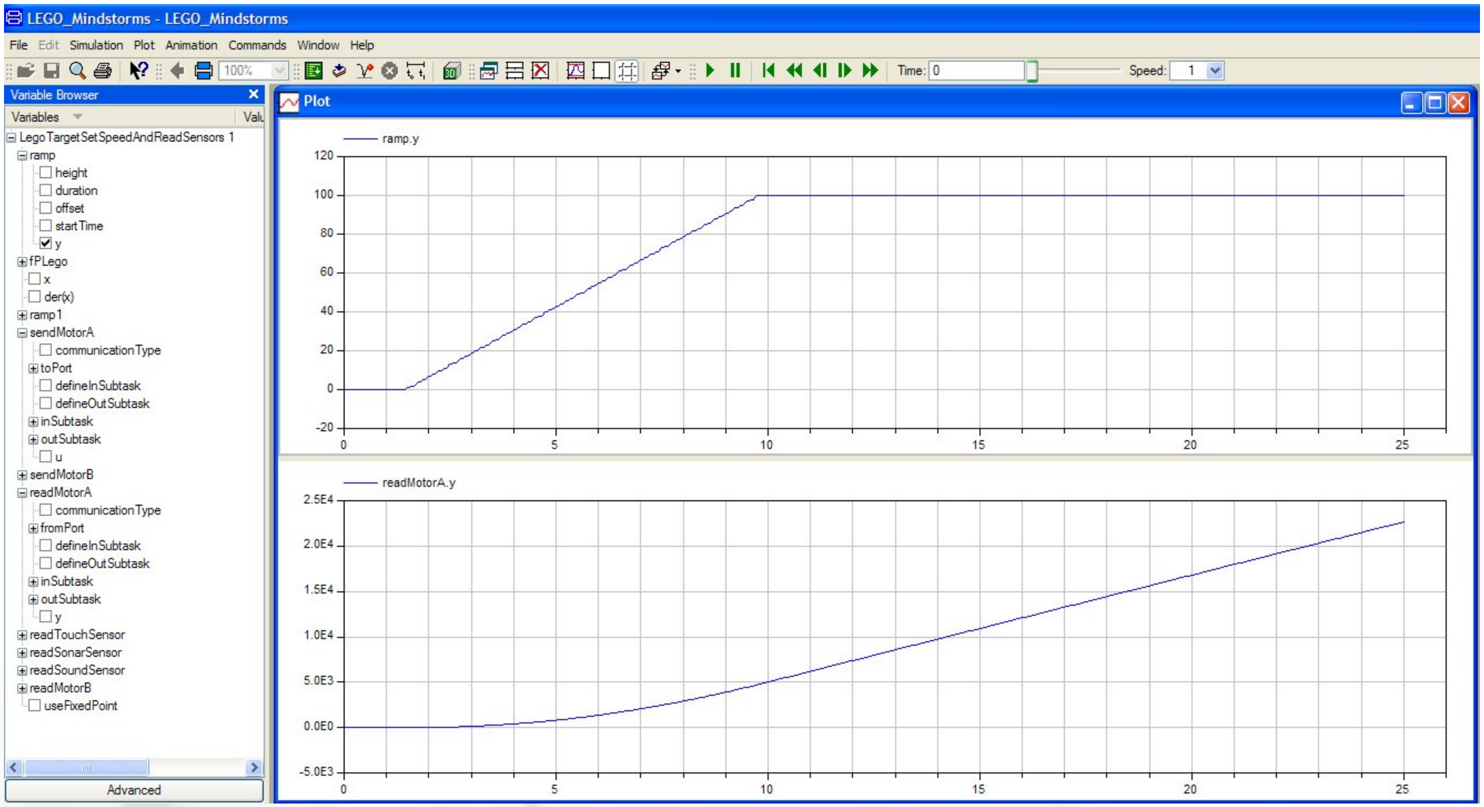




Lego Mindstorms

Setup/Configuration

- **Results are automatically stored in Dymola**



DASSAULT SYSTEMES

Thank you for you attention
and
Good Luck with your projects