RCP
○○○

Linux RTAI
○○

RTAI-Lab
○○○

EicasLab
○○○

Examples
○

Conclusions
○○

Rapid Control Prototyping with EicasLab and Linux RTAI

Roberto Bucher and Kai Kaufmann

Scuola Universitaria Professionale della Svizzera Italiana (SUPSI)
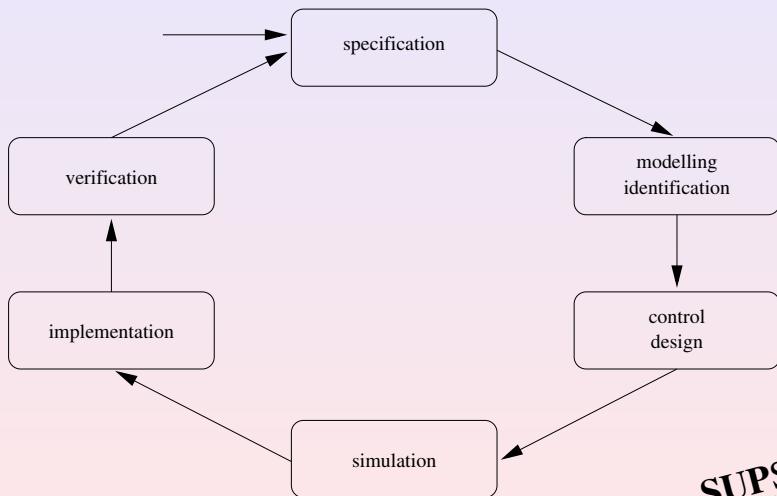
Acoduasis Workshop - Torino

SUPSI

Outline

1. Rapid Controller Prototyping

2. Linux RTAI

3. RTAI-Lab

4. EicasLab

5. Examples

6. Conclusions

SUPSI

Rapid Controller Prototyping - Design

RCP - Requirements

- 2 main components
    1. An OS with hard real-time features
    2. A CACSD environment including a code generator

SUPSI

RCP	Linux RTAI	RTAI-Lab	EicasLab	Examples	Conclusions
○●○	○○	○○○	○○○	○	○○

Introduction

RCP - Requirements

- 2 main components
  1. An OS with hard real-time features
  2. A CACSD environment including a code generator

$\rightarrow$ Linux RTAI + RTAI-Lab + EicasLab

SUPSI

RCP at the SUPSI

- Matlab/Simulink/RTW and AD SHARC DSP (1998)

SUPSI

RCP at the SUPSI

- Matlab/Simulink/RTW and AD SHARC DSP (1998)
- Matlab/Simulink/RTW and Linux RTAI in kernel space (2001)

SUPSI

RCP at the SUPSI

- Matlab/Simulink/RTW and AD SHARC DSP (1998)
- Matlab/Simulink/RTW and Linux RTAI in kernel space (2001)
- Scilab/Scicos and Linux RTAI in kernel space (2002)

SUPSI

RCP at the SUPSI

- Matlab/Simulink/RTW and AD SHARC DSP (1998)
- Matlab/Simulink/RTW and Linux RTAI in kernel space (2001)
- Scilab/Scicos and Linux RTAI in kernel space (2002)
- Scilab/Scicos and Linux RTAI in user space (2003-2005)

SUPSI

RCP at the SUPSI

- Matlab/Simulink/RTW and AD SHARC DSP (1998)
- Matlab/Simulink/RTW and Linux RTAI in kernel space (2001)
- Scilab/Scicos and Linux RTAI in kernel space (2002)
- Scilab/Scicos and Linux RTAI in user space (2003-2005)
- Targeting Matlab/Simulink/RTW for Texas FP DSP

SUPSI

RCP at the SUPSI

- Matlab/Simulink/RTW and AD SHARC DSP (1998)
- Matlab/Simulink/RTW and Linux RTAI in kernel space (2001)
- Scilab/Scicos and Linux RTAI in kernel space (2002)
- Scilab/Scicos and Linux RTAI in user space (2003-2005)
- Targeting Matlab/Simulink/RTW for Texas FP DSP
- EicasLab and Linux RTAI (2005)

SUPSI

RCP	Linux RTAI	RTAI-Lab	EicasLab	Examples	Conclusions
○○○	●○	○○○	○○○	○	○○

Basics

The Linux RTAI project

- Hard real-time extension to the Linux OS
- Based on the ADEOS nano-kernel
- Free Open Source Software (FOSS)
- Implementation of hard real-time controllers using general purpose hardware (x86, ARM, PPC)
- Same PC for the controller design, the hard real-time controller task and the soft real-time monitor task
- Hard real-time in kernel and user space
- Distributed control through the net_rpc module

SUPSI

Linux RTAI - Features

- Latency: $< 10\mu s$ depending on the HW
- Typical sampling frequencies: $1 \ldots 10 kHz$

SUPSI

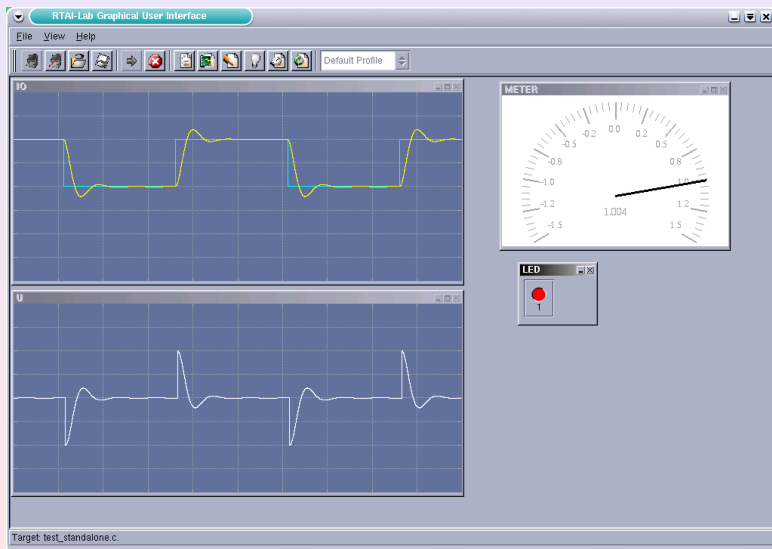| RCP	Linux RTAI	RTAI-Lab	EicasLab	Examples	Conclusions
000	00	●00	000	0	00

The framework

RTAI-Lab

- common structured framework for the integration of RTAI into CACSD environments.
- Includes support for MATLAB/Simulink and SCILAB/Scicos
- two separate systems:
  - the host with RTAI-Lab as soft realtime application
  - the target with the hard realtime task, generated by the CACSD environment
- Access to DAQ through COMEDI drivers
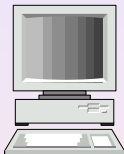- communication occurs using MBX and MSG

SUPSI

RCP
○○○

Linux RTAI
○○

RTAI-Lab
○●○

EicasLab
○○○

Examples
○

Conclusions
○○

The framework

The GUI application - xrtailab

RCP	Linux RTAI	RTAI-Lab	EicasLab	Examples	Conclusions
ooo	oo	ooo●	ooo	o	oo

The framework

The GUI application - xrtailab

PC with HRT task and xrtailab

SUPSI

RCP	Linux RTAI	RTAI-Lab	EicasLab	Examples	Conclusions
ooo	oo	ooo●	ooo	o	oo

The framework

The GUI application - xrtailab

PC with HRT task   PC with xrtailab

Network

UDP/IP
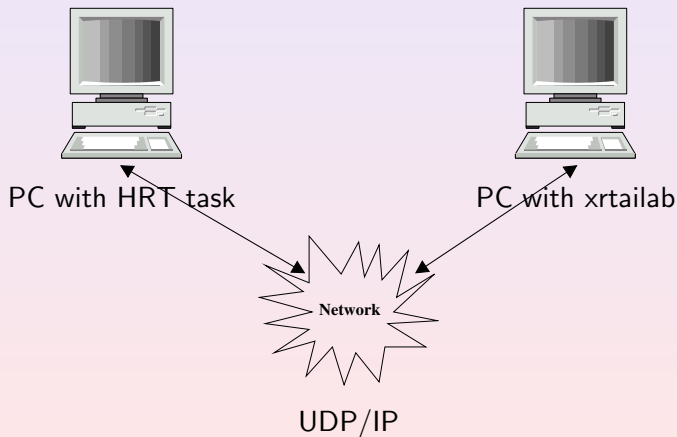
SUPSI

The student project

- The project started on September 12
- 1 Processor, 1 Thread, 1 sampling time
- Integration in the RTAI-Lab environment
- Future work: multiple sampling times, multiple processors, multiple threads
- Collect experience in order to produce code for other targets

SUPSI

Code generation
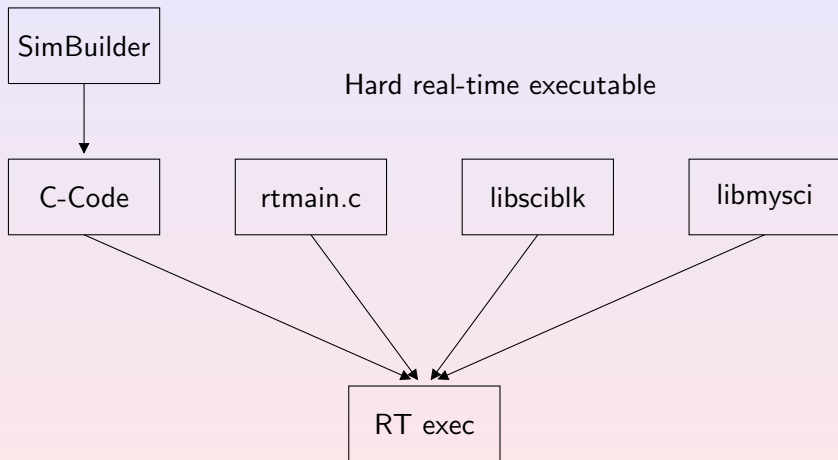
SimBuilder

SUPSI

Code generation

| SimBuilder |

↓

| C-Code |

ACG

SUPSI

Code generation

SimBuilder

Compile e link

C-Code   rtmain.c   libsciblk   libmysci
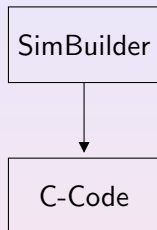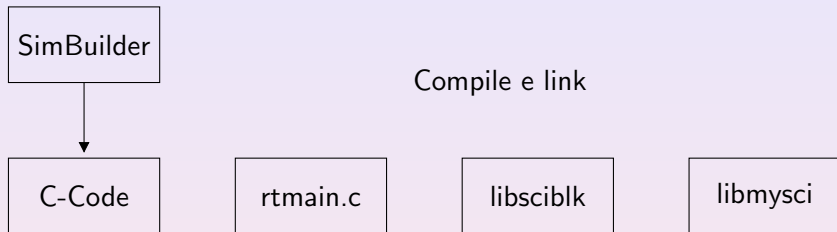
SUPSI

Code generation

The main file *rtmain.c*

3 threads:

- Main thread *rtMain*
- Hard RT thread *rt_BaseRate*
- Communication thread with the GUI client application *rt_HostInterface*
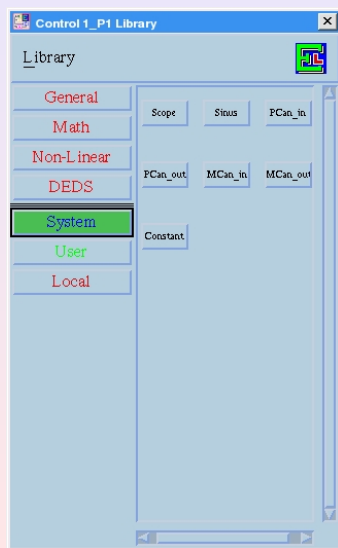
SUPSI

RCP	Linux RTAI	RTAI-Lab	**EicasLab**	Examples	Conclusions
000	00	000	0●0	0	00

The "main" file
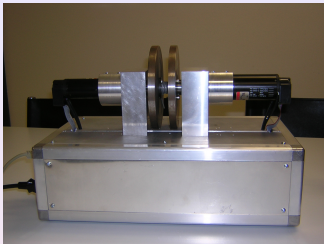
The *rt_BaseRate* thread
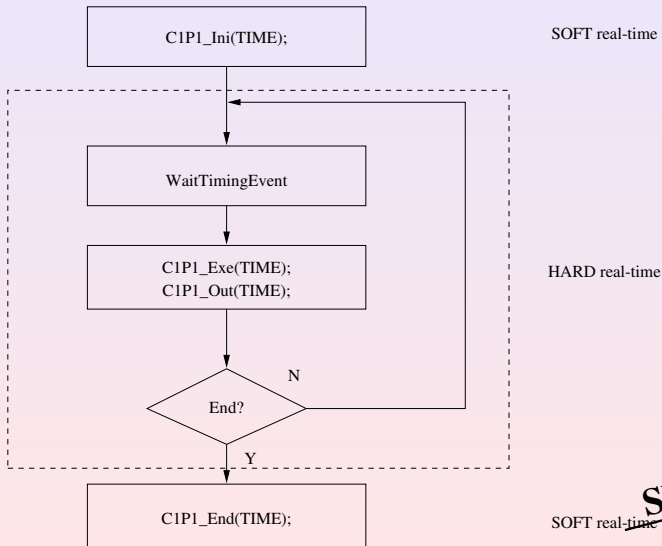
Eicas Library

Disks and spring
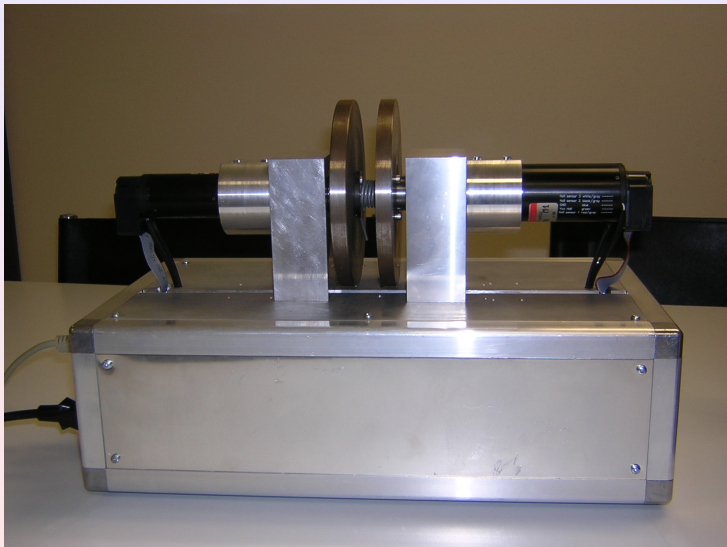
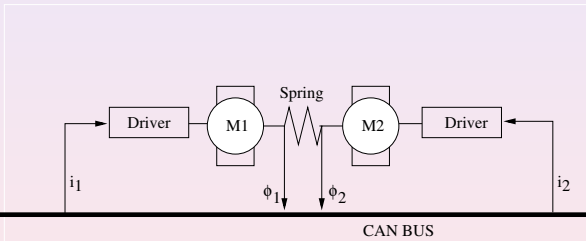# Disks and spring

Disks and spring
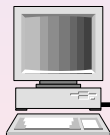
PC with HRT
controller

SUPSI

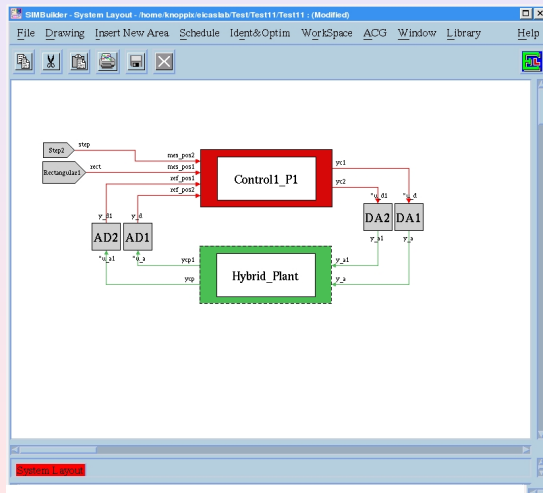RCP
○○○

Linux RTAI
○○

RTAI-Lab
○○○

EicasLab
○○○

Examples
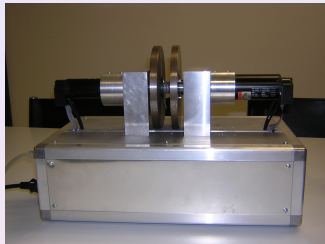●

Conclusions
○○

Disks and spring

Disks and spring

Conclusions

- Work in progress
- At present: 1 Controller, 1 Processor
- Future work: Multiprocessors, Multicontrollers
- Other targets!

SUPSI

Questions?

Thank You

- www.rtai.org
- www.dti.supsi.ch/~bucher
- www.dti.supsi.ch

SUPSI