SafetyChip
A Time Monitoring and Policing Device

SigAda 2005

2005-11-16

Gustaf Naeser, Lars Asplund and Johan Furunäs

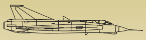
Mälardalen University

MÄLARDALENS HÖGSKOLA

MRTC MÄLARDALEN REAL-TIME RESEARCH CENTRE

Outline

- Formal Methods
- **Hardware Monitoring**



Application : Robocup – Soccer playing robots

Hardware Monitoring and Policing

- How can the operation of a system during run time be
  - Observed
  - Modified
- Aids development
- The largest part of a system's life is after deployment

Kinds of monitoring

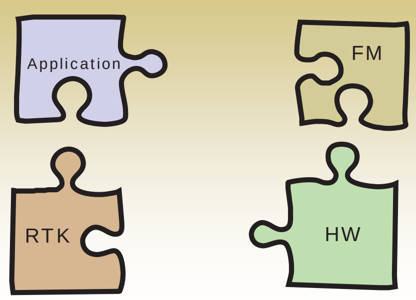
Intrusive

- Changes the behaviour of the monitored system
  - Code is added to generate observable events
  - Monitor can be off target
  - Instructions are kept in the final system to keep the behaviour

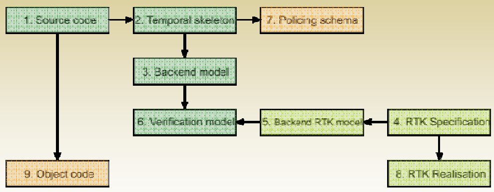
Non-intrusive

- Does not change the systems behaviour
  - Harder to implement since it requires access to tap points "inside" the system

The SafetyChip framework parts

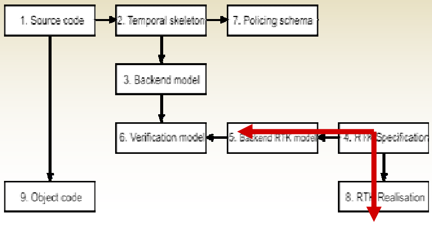


The Framework



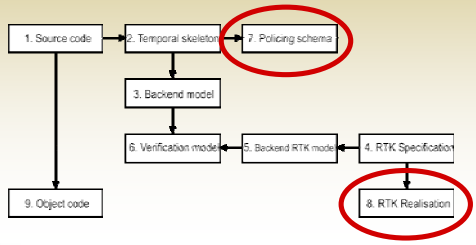
System on Chip

A Ravenscar RTK
- Component based
- Allows timing analysis

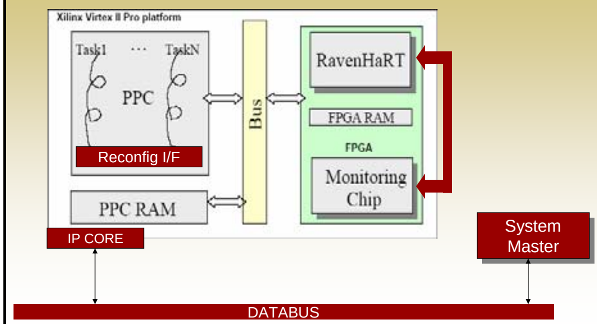


Monitoring and Policing

Reuses the model from the verification



Non-Intrusive Fault-Tolerance

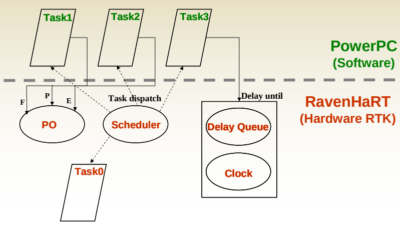


Developments in hardware

- Field Programmable Gate Arrays (FPGA) is a programmable hardware combining properties from both hardware and software
  - Faster than SW, slower than HW (ASIC)
  - Easier to change than HW, harder than SW

RavenHaRT

- Hardware implemented RTK
  - Based on the Ravenscar profile of Ada95
  - Implemented using the Xilinx Virtex II Pro
- Deterministic run-time behavior
- Non-Deadlocking Inter-task communication



RTK

- Ravenscar profile supported
- Supports a more functionality at low cost
  - *e.g.*, dynamic priorities
  - Multiple processors
- Component design to allow
  - Easy change
  - HW/SW locality in final system
- Implemented using generic templates
  - Allows the kernel to be tailored to each application

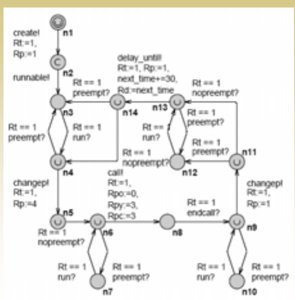
RTK components

- Ready Queue
- Delay Queue
- Protected Objects Queue
- Interrupt Queue (uses PO)
- Hardware
  - Processors with null tasks
  - clock
- Application
  - Tasks and protected objects

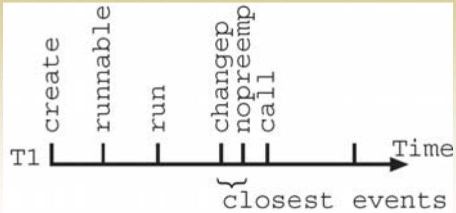
The Ada Code for task T1

```
task T1;
task body T1 is
   T_1 :  Task_ID := 1;
   Next_Time :  Time := Start_Time
                        + To_Time_Span(3.0);
begin
   loop
      PO.P(T_1);
      Next_Time:= Next_Time + To_Time_Span(3.0);
      delay until Next_Time;
   end loop;
end T1;
```

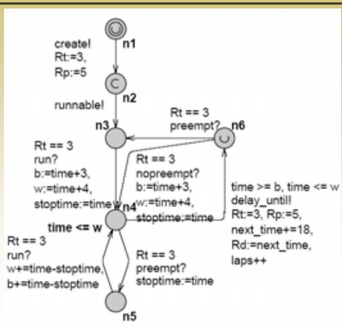

UPPAAL of Task T1



Timing of Task T1



Task T3



Behaviour for Task T3

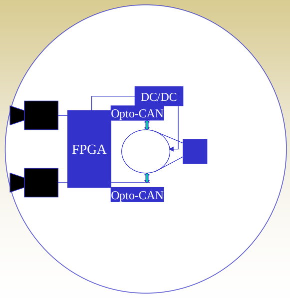
Address	Branch	Call / Node	BCET	WCET
00	00	node (n_1)	0	0
01	02	runnable	0	0
02	03	period	18	18
03	00	node (n_3)	0	0
04	05	run	0	0
05	00	node (n_4)	3	4
06	07	delay_until	3	4
07	00	node (n_6)	0	0
08	03	preempt	0	0
09	10	nopreempt	0	0
10	05	period	18	18

RoboCup

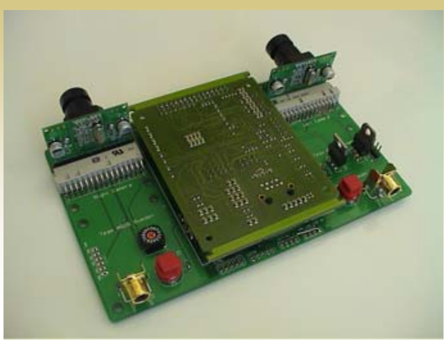
- Continued

Team Aros

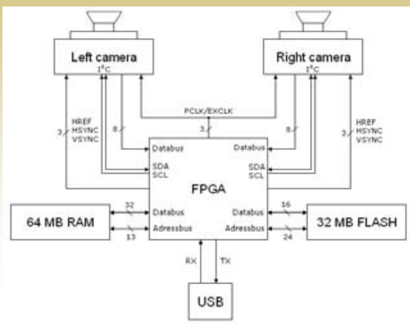
Aros disk – Camera



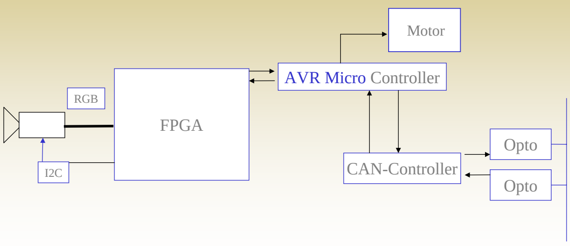
FPGA and Cameras



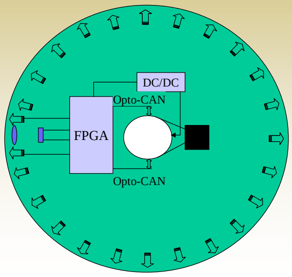
Block Diagram of the Vision System



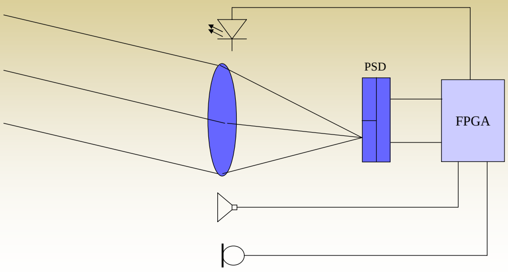
Camera Block Diagram



IR-Ultra



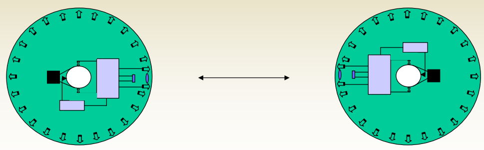
IR / Ultra



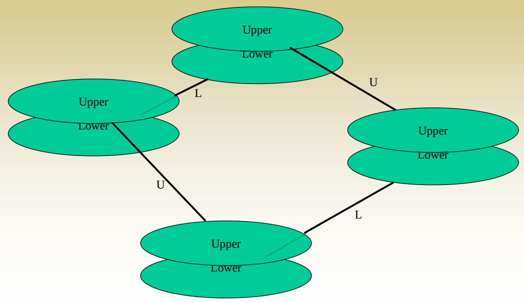
No Synchronization

Searching

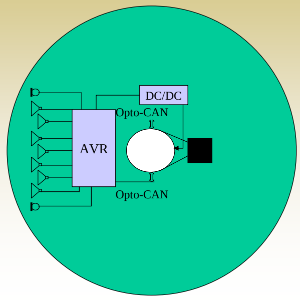
Synchronized - contact



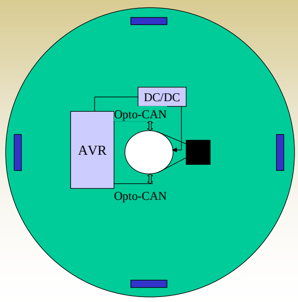
IR/Ultra between 4 robots



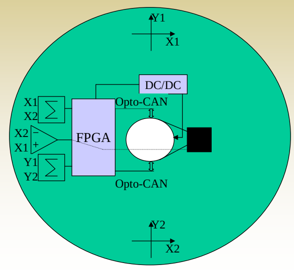
Ultra sonic 'radar'



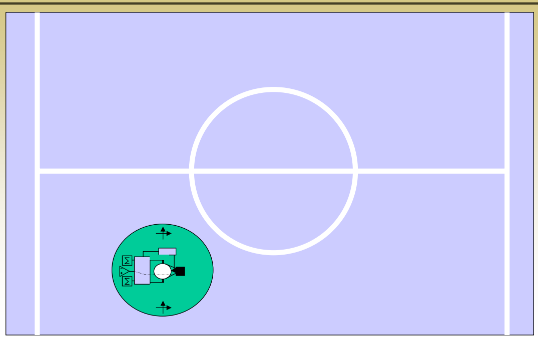
Ir – Collision Avoidance



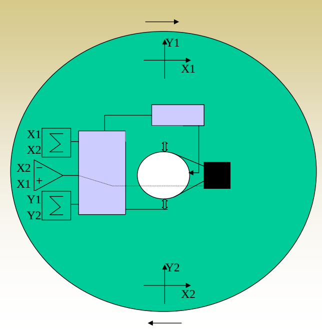
Gyro



Line up at start



Rotation

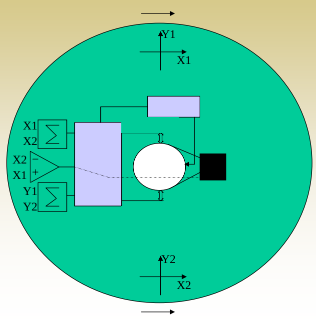


X1 = Omega * R
X2 = -Omega * R
Y1 = 0
Y2 = 0

X1-X2 = 2 * Omega * R
X1+X2 = 0
Y1+Y2 = 0

X1-X2 styr motorn
(kompensation)

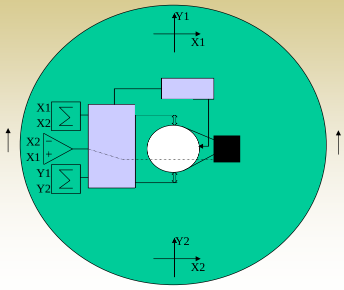
X-translation



X1 = a (v eller s)
X2 = a
Y1 = 0
Y2 = 0

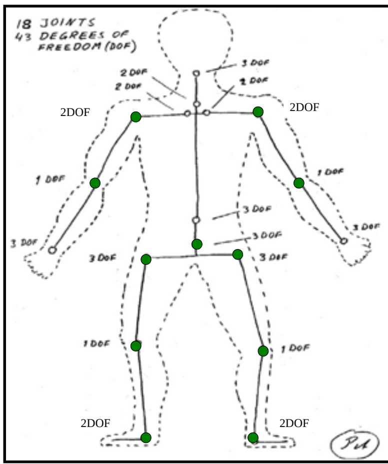
X1-X2 = 0
X1+X2 = 2 * a
Y1+Y2 = 0

Y-translation



X1 = 0
X2 = 0
Y1 = a
Y2 = a

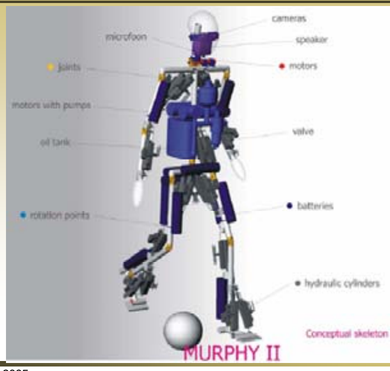
X1-X2 = 0
X1+X2 = 0
Y1+Y2 = 2 * a



Murphy

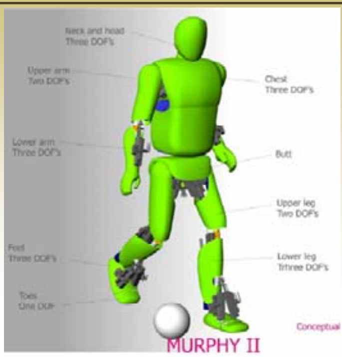
- Arm 2*(2+1) = 6
- Back = 3
- Leg 2*(3+1) = 8
- Feet 2*2 = 4
- Total = 21

Next Version of Murphy



SigAda 2005 57

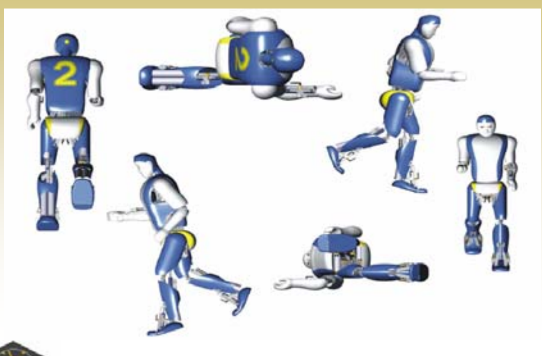
Murphy with Cover



SigAda 2005 58

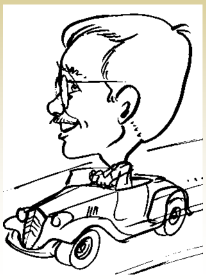
New Name = Anders??

SigAda 2005 59



SigAda 2005 60

The End



SigAda 2005 61