

Digital Ship

NORSHIPPING
9 June 2005

Managing the risk of ship control, monitoring and alarm systems

Computers and Networks in Ship Control

Jens Dalsgaard Nielsen, Henrik Schiøler

Aalborg University Denmark

<http://www.control.aau.dk>
<http://www.control.aau.dk/~{jdn,henrik}>

Jens Dalsgaard Nielsen

- Employed at Aalborg University – AAU (1984 - ...)
 - Institute of Electronic Systems
 - Department of Control Engineering - www.control.aau.dk
- Areas of interest
 - Control Systems
 - Field buses / distributed systems
 - Real time systems
 - fault resilience
 - Automation
- Ship automation
 - ATOMOS (Advanced Technology for Optimizing Manpower Onboard Ships)
 - DBC, PBCES, MITS, ...

JDN today

- Autonomous systems
 - satellites (students at AAU, ESA)
 - agricultural autonomous vehicles
 - ship boilers
- Infrastructure Convergence
 - CTIF - Center for Tele Infrastructure
 - QoS in ...
 - Integrated Services
 - Real time properties
 - Fail resilience
- Embedded systems
 - linux, rtai, ...
 - rtp, ...
- Network planning
 - Just now building on a 40 nodes routing test bench (linux)

Computers and Networks in Ship Control

- *To appear as a homogeneous robust control system*
- ...

Computers and Networks in Ship Control

- *To appear as a homogeneous robust control systems*
- *Intelligent machines/subsystems*
- ...

Computers and Networks in Ship Control

- *To appear as a homogeneous robust control systems*
- *Intelligent machines/subsystems*
- *To be able to operate in degraded mode(s)*
 - *network failure*
 - *node failure*
 - *I/O failure*
 - *machinery failure*
- ...

Computers and Networks in Ship Control

- *To appear as a homogeneous robust control systems*
- *Intelligent machines/subsystems*
- *To be able to operate in degraded mode(s)*
 - *network failure*
 - *node failure*
 - *I/O failure*
 - *machinery failure*
- *No experts around*
- *To expect several days of "degraded" operation*

Computers and Networks in Ship Control

should be as natural as ...



A footnote: The legal stuff

- Rules "rules"
- ...



A footnote: The legal stuff

- Rules "rules"
- which rules ?
- ...



A footnote: The legal stuff

- Rules "rules"
- which rules ?
- 100% compatible ?
- ...



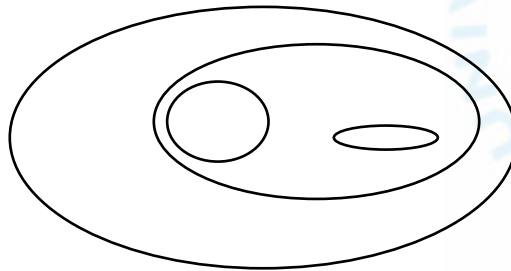
A footnote: The legal stuff

- Rules "rules"
- which rules ?
- 100% compatible ?
- loopholes ? -not my business today ;-)



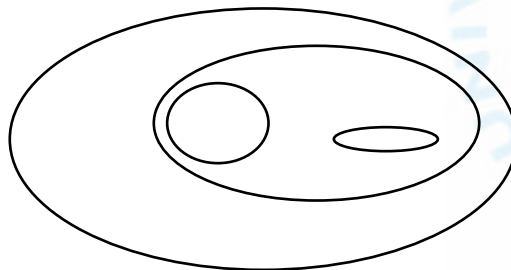
Computers and Safety

- *High (est) degree of safety/robustness is a local phenomena*
- ...



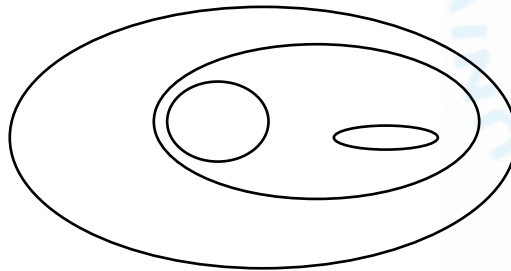
Computers and Safety

- *High (est) degree of safety/robustness is a local phenomena*
- *High (est) safety can be on system level due to use of redundancy or other techniques*
- ...



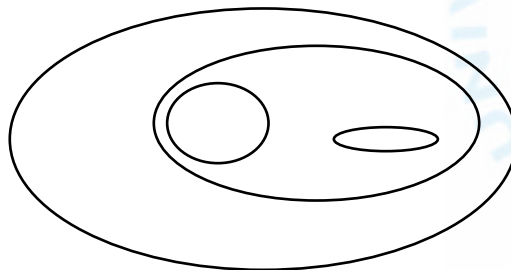
Computers and Safety

- High (est) degree of safety/robustness is a local phenomena
- High (est) safety can be on system level due to use of redundancy or other techniques
- Still atomic item is a computer/node
- ...



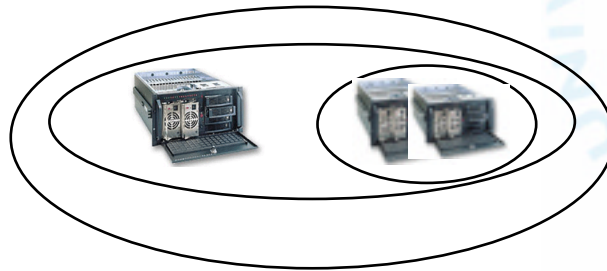
Computers and Safety

- High (est) degree of safety/robustness is a local phenomena
- High (est) safety can be on system level due to use of redundancy or other techniques
- Still atomic item is a computer/node
- basic argument: *if anything fails the single controllers still runs*
- ...



Computers and Safety

- High (est) degree of safety/robustness is a local phenomena
- High (est) safety can be on system level due to use of redundancy or other techniques
- Still atomic item is a computer/node
- basic argument: if anything fails the single controller still runs
- System can have higher quality than the single components or the opposite !!! depending on your design...



Safety and quality considerations

HW

- All standard electrical stuff, EMC, IP65,
- ...



A comment up front

We are aware of:

The functional standard for describing and “verifying” your system

- IEC 61 508 functional safety
 - SIL 1..4 (Safety Integrity Level 1..4)
 - SIL -1:1 bad error every 11.4 year
 - SIL -4:1 bad error every < 1.142 year or better ...
- 7 chapters including training, SW , ...

...



A comment up front

We are aware of:

The functional standard for describing and “verifying” your system

- IEC 61 508 functional safety
 - SIL 1..4 (Safety Integrity Level 1..4)
 - SIL -1:1 bad error every 11.4 year
 - SIL -4:1 bad error every < 1.142 year or better ...
- 7 chapters including training, SW , ...

We think it is nearly necessary

...

A comment up front

We are aware of:

The functional standard for describing and “verifying” your system

- IEC 61508 functional safety
 - SIL 1..4 (Safety Integrity Level 1..4)
 - SIL -1:1 bad error every 11.4 year
 - SIL -4:1 bad error every < 1.142 year or better ...
- 7 chapters including training, SW, ...
- SIL 3 can be made on SIL 2 components
- and the other way around – it is up to you – and your money

We think it is nearly necessary

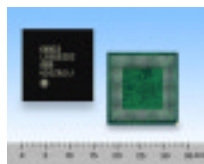
Does not say anything about how – it is a functional standard

Fieldbus standards, Operating systems, ... is still needed

!

Safety and quality considerations

HW – a model



Safety and quality considerations

HW - a model



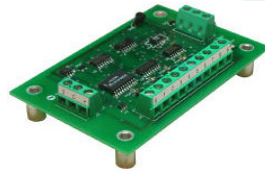
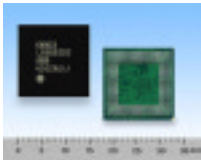
Safety and quality considerations

HW - a model



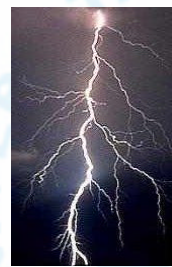
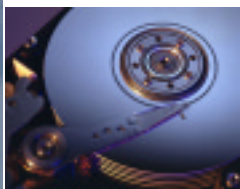
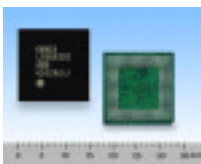
Safety and quality considerations

HW - a model



Safety and quality considerations

HW - a model



HW - short (one slide)

Memory

- Error detection / error correction (ECC)

CPU

- Error detection
 - watchdog
 - redundancy in space and time

Storage

- Mechanical device :- (really no comments)

I/O

- harsh environments

OUTSIDE

- We have to follow the world precisely (keep up speed)

Some figures - for illustration

Raw speed (MHz, MIPS, ...)

- 6->3000 MHz

Memory

- 4 -> 1.000.000 kB

I/O

- AD, DA, DD, PWM, ... 1 -> 400 in/out, up to several kV, ...
- rs232, 485, ...
- fieldbus 1, 2, 3, 4, ...
- general network
- Human interface

Operating System

- from 200 B to 2.000.000 B or more in size
- many approaches

Some figures - for illustration II

Raw speed (Mhz, MIPS, ...)

- 6 → 3000 Mhz

Memory

- 4 → 1.000.000 kB

Power consumption

- 0.?? → 100 W attormore

I/O

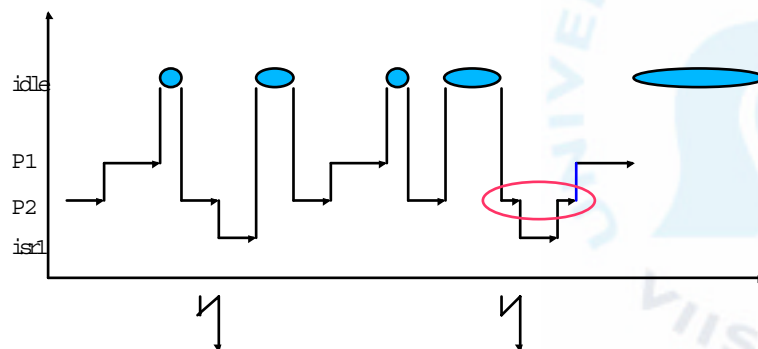
- AD, DA, DD, PWM, ... "0" → 400
- rs232, 485, ...
- fieldbus 1, 2, 3, 4, ...
- general network

IT IS VERY DIFFICULT TO GENERALIZE

Simple Modeling Approach I

Code execution

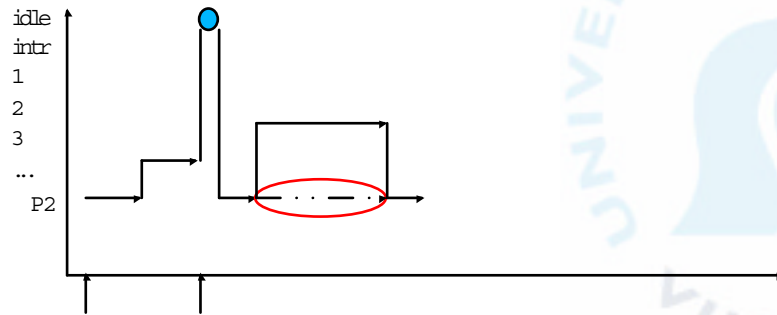
- Threaded code
- Cyclic/event
- simple/complex (measurable)



Simple Modeling Approach II

Code execution

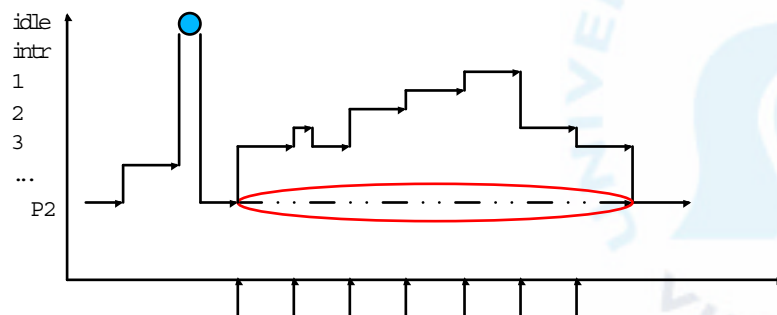
- Threaded code
- Cyclic/event
- simple/complex (measurable)



Simple Modeling Approach II

Code execution

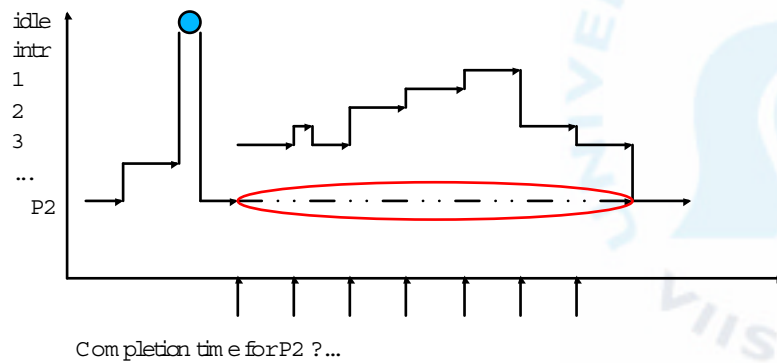
- Threaded code
- Cyclic/event
- simple/complex (measurable)



Simple Modeling Approach II

Code execution

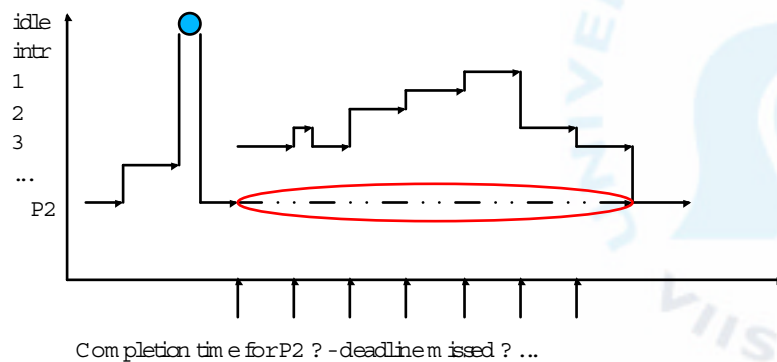
- Threaded code
- Cyclic/event
- simple/complex (measurable)



Simple Modeling Approach II

Code execution

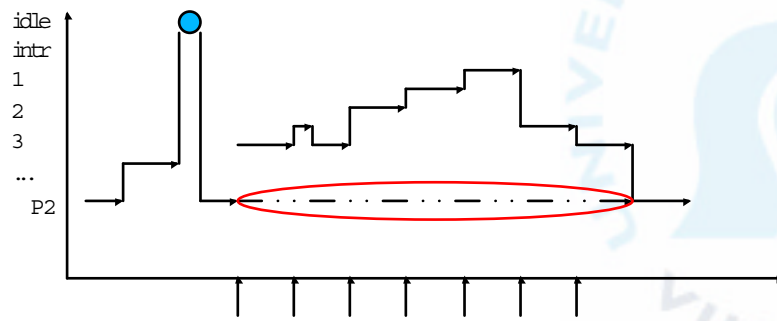
- Threaded code
- Cyclic/event
- simple/complex (measurable)



Simple Modeling Approach II

Code execution

- Threaded code
- Cyclic/event
- simple/complex (measurable)



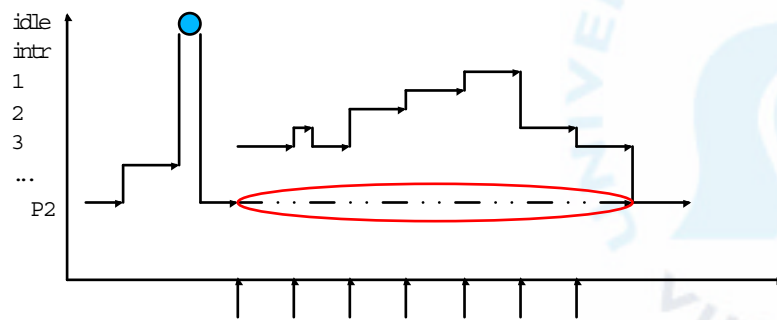
Completion time for P2? - deadline missed? - system unstable?

...

Simple Modeling Approach II

Code execution

- Threaded code
- Cyclic/event
- simple/complex (measurable)



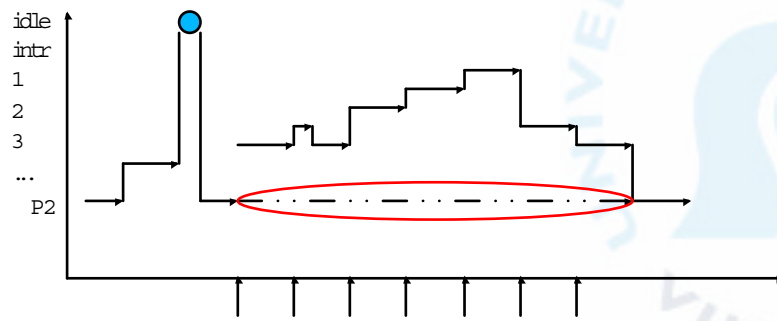
Completion time for P2? - deadline missed? -

system unstable? ... this was just a very small example

Simple Modeling Approach II

Code execution

- Threaded code
- Cyclic/event
- simple/complex (measurable)



Hundred of such systems onboard ships !

Simple Modeling Approach II

Code execution

- Threaded code
- Cyclic/event
- simple/complex (measurable)



Hundred of such systems onboard ships !

Simple Modeling Approach II

Code execution

- Threaded code
- Cyclic/event
- simple/complex (measurable)



Hundred of such systems onboard ships !
 AND THEY WORKS VERY WELL 24/7/365 :-)

Simple Modeling Approach III

Code/functional modelling

...

Simple Modelling Approach III

Code/functional modelling

Cyclic code (properties like)

- period of execution, execution time
- delay borderlines, ...
- communication
- mutual exclusions

...

Simple Modelling Approach III

Code/functional modelling

Cyclic code (properties like)

- period of execution, execution time
- delay borderlines, ...
- communication
- mutual exclusions

Calculations can be made for

- minimum mean delays
- properties fulfilled or not

...

Simple Modeling Approach III

Code/functional modelling

Cyclic code (properties like)

- period of execution, execution time
- delay borderlines, ...
- communication
- mutual exclusions

Calculations can be made for

- minimum mean delays
- properties fulfilled or not

Scheduling methods to be selected

- static priority based etc
- dynamic deadline approaches

!

Simple Modeling Approach IV

Code/functional modelling II

Events (interrupts)/non periodic code

- behavior/pr "definition" often unknown
- You must assume
 - worstcase (no of interrupts/interval)
- You must know
 - Time length of interrupt service code
 - if it is interruptible or not

Interrupt scheme

- priority based / flat / dynamic/static priority

Some small calculations

Interrupt

- Interrupt latency 5-25 usec "not" CPU dependent
 - no big difference between 40 Mhz (arm) and 2000 Mhz (PC)!!!
- Larger delay up to "user space"
- 100 usec for an interrupt gives MAX 10.000 interrupts per second
- M odelable?
- V erifiable ???

...

Some small calculations

Interrupt

- Interrupt latency 5-25 usec "not" CPU dependent
 - no big difference between 40 Mhz (arm) and 2000 Mhz (PC)!!!
- Larger delay up to "user space"
- 100 usec for an interrupt gives MAX 10.000 interrupts per second
- M odelable?
- V erifiable ???

Interrupt II

- Prioritized or equal
- Static or dynamic
- Inside ISR blocking times

Some small calculations II

A mind spin

- An interrupt can be delayed by other interrupts
- 1 isr of (20 usec + 40 usec code)
- System call and kick to userspace: 20 usec
- Influence from scheduling
- Influence from HW like position of harddisk (500 usec - ...)
- Influence from (PC) chipset???

A question

- Do you really want to have time of period less than 1 m sec ????
- ...
- Test your PC : <http://ssaris.org/ta/>
- Life ain't always "happy days"

How do you get

Evidence for functionality?

...

How do you get

Evidence for functionality?



Ask your local dealer :-)

KISS



KISS

Quality of verification/simulation/calculating is dependent of

- Model quality of code/system
- HW model quality
- Measurements

!!!



NETWORKING

NETWORKING

To interconnect computing devices

Some characteristics

- 5 to 5.000 m length
- 9600 to 1.000.000.000 bits/sec
- 8 to 9000 B user data frame
- 2 - ??? nr of nodes

Topology

- bus / ring / star
- switched or not

Access protocol:

- arbitrary with collision detection/avoidance (csma, {cd,ca})
- token slotted
- time based
- (non) prioritized

Modeling network

Some prerequisites

- Technology
- Topology
- Access protocols
- Higher layer protocols (if they are modelable)

AND

...

An example - canbus



- CANBUS has real time behavior :-)
- and very shortcabling :-)
- Scheduling theory can be used :-)

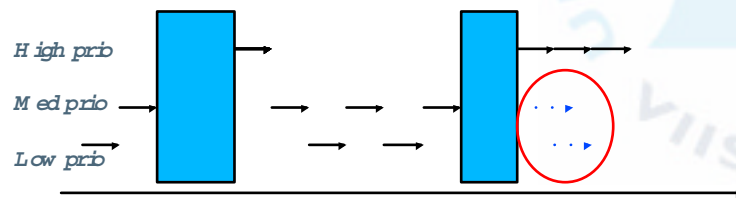
An example - canbus

CANBUS has real time behavior

- *for the high (est) priorities*
- *and very shortcabling*
- *Scheduling theory can be used*

Scheduling of traffic

- *layout Highest priority telegrams*
- *next 2nd priority where there are space*
- *...*
- *In many ways equal to rate monotonic schedule*



canbus

Deterministic:

- Collision avoidance on a packet priority paradigm
- Every node can transmit one package at each token passing
- A package is up to 8 B + head & tail
- Max delay = ?? ...
 - Dependent of all the other packet priorities
- ...

canbus

Deterministic:

- Collision avoidance on a packet priority paradigm
- Every node can transmit one package at each token passing
- A package is up to 8 B + head & tail
- Max delay = ?? ...
 - Dependent of all the other packet priorities
- Queuing theory is applicable
- Results can be discussed when packet timing is close to network timing.
- Easy to simulate
- ...

canbus

Deterministic:

- Collision avoidance on a packet-priority paradigm
- Every node can transmit one package at each token passing
- A package is up to 8 B + head & tail
- Max delay = ?? ...
 - Dependent of all the other packet priorities
- Queuing theory is applicable
- Results can be discussed when packet timing is close to network timing.
- Easy to simulate
- Quality rely on traffic description quality
- ...

canbus

Deterministic:

- Collision avoidance on a packet-priority paradigm
- Every node can transmit one package at each token passing
- A package is up to 8 B + head & tail
- Max delay = ?? ...
 - Dependent of all the other packet priorities
- Queuing theory is applicable
- Results can be discussed when packet timing is close to network timing.
- Easy to simulate
- Quality rely on traffic description quality
- DO YOU KNOW YOUR TRAFFIC SCHEDULE ?
- DO YOU KNOW YOU RUN PLAN FOR YOUR CODE ?

canbus – two comments

Agricultural world:

Functional standards like ISO 11783 describing

- *Communication between my tractor and peripherals*
- *Functional profiles of equipment*
 - *Conformance classes*

which means a John Deere tractor can manage a Kongskilde item without any "reprogramming" ...

SIL world: SIL level 3 !!

- *CAN open safety protocol by CAN chip CSC 01*
- *20 m sec safety cycle*
- *2 m sec for safety application (your code)*
- *Code size < a few kB*
- *A primitive but very safe system (<http://can-cia.org/csc>)*

Token based Systems

Deterministic:

- *Token traveling paradigm is well known*
- *IBM token ring 16 M bit*
- *ARCNET 0,1 ... 4 M bit*
- *rs485 sw/hw implements*

Traffic scheme

- *Token circulate on equal basis*
- *A node is obliged to send 1 package each token passing*
- *worstcase time to wait a token rotation*
- *Routing kills the system real time behavior*

Modeling

- *Easy*
- *Worstcase estimate very negative*
- *Routing is just combination of independent systems (no RT)*
- *Markov chains, ... - not in scope for today*

Ethernet

LOWER LAYERS:

Stochastic

- 10 M bit bus based

Deterministic

- 100/1000 M bit P2P / switch based

Wireless

Optical

Ethernet

Difficult to be general due to large span in parameters

Bus based Ethernet for control: nope

Wireless Ethernet for control: nope

Switch based Ethernet (100 M b / 1000 M b): YES

- 100 to 1000 higher bit rate than field buses
- Several 1000 packages per node per second
- Switching technology "eliminate" rejection
- Easy queue modeling (in arkov chains)
- Routing efficient
- Fail resilient systems (like < 1 sec spanning tree)
- Redundant systems available