Embedded Systems

Software Engineering in Embedded
Systems

Stephan Heidinger

Seminar: Software Engineering Fachbereich für Informatik und Informationssysteme Universität Konstanz

19. January 2012

Embedded Systems

Stephan Heidinger

Embedded Systems Design

Architectural patterns

Timing analysis

Real-time operating system

Software Engineering in Embedded Systems

Stephan Heidinger

Seminar: Software Engineering Fachbereich für Informatik und Informationssysteme Universtität Konstanz

19. January 2012



Embedded Systems

Embedded Systems - What's that? - I

Embedded Systems - What's that? - I

Defeation
'An embedded software system is part of a hardware/software system that reacts to events in its environment. The software is 'embedded in the hardware. Embedded systems are nominally real-time systems."

**The repurse of Not their to bit means to the stem.

Embedded Systems - What's that? - I

mbedded Systems

Stephan Heidinger

Embedded Systems Design

T'...'..........

I iming analys

Real-time operating system

Definition

"An **embedded software** system is part of a hard-ware/software system that reacts to events in its environment. The software is 'embedded' in the hardware. Embedded systems are nominaly real-time systems."

Software Engineering, p.561, Edited by Ian Sommerville, Ninth Edition

Embedded Systems

—Embedded Systems - What's that? - II

Embedded Systems - What's that? - II

» Embedded Systems: ...

Embedded Systems - What's that? - II

Embedded Systems

Stephan Heidinge

Embedded Systems Desig

T'...'.....

Real-time

• Embedded Systems: . . .

- ...respond to physical world
- ... respond in real time ("have a deadline")
- ... often have little resources
- ...run on special purpose hardware
- ...run in real-time operating system

Embedded Systems

—Embedded Systems - What's that? - II

Embedded Systems - What's that? - II

Embedded Systems

Stephan Heidinge

Embedded Systems Desig

patterns

Real-time

- Embedded Systems: ...
 - ...respond to physical world
 - ...respond in real time ("have a deadline")
 - ... often have little resources
 - ...run on special purpose hardware
 - ...run in real-time operating system

Embedded Systems - What's that? - II

- Embedded Systems:
- ... respond to physical world

Embedded Systems - What's that? - II

- Embedded Systems: ...
 - ... respond to physical world
 - ... respond in real time ("have a deadline")
 - ... often have little resources
 - ...run on special purpose hardware
 - ...run in real-time operating system

Embedded Systems

Embedded Systems - What's that? - II

- Embedded Systems What's that? II
 - » Embedded Systems:
 - ... respond to physical world
 - .. respond to physical world .. respond in real time ("have a deadline")
 - ... often have little resources

Embedded Systems - What's that? - II

Embedded Systems

Stephan Heidinge

Embedded Systems Design

Timing analys

Real-time operating system

- Embedded Systems: . . .
 - ... respond to physical world
 - ...respond in real time ("have a deadline")
 - ... often have little resources
 - ...run on special purpose hardware
 - ...run in real-time operating system

Embedded Systems

Embedded Systems - What's that? - II

Embedded Systems - What's that? - II

- Embedded Systems:
- mbedded Systems: ...
 respond to physical world
 - pond to physical world
- ... often have little resources
 ... run on special purpose hardware

Embedded Systems - What's that? - II

Embedded Systems

Stephan Heidinge

Embedded Systems Desig

T'...'

Real-time operating system • Embedded Systems: ...

- ...respond to physical world
- ...respond in real time ("have a deadline")
- ...often have little resources
- ...run on special purpose hardware
- ...run in real-time operating system

Embedded Systems

Embedded Systems - What's that? - II

Embedded Systems - What's that? - II

- Embedded Systems:
- mbedded Systems: ...
 respond to physical world
 - ipond to physical world inned in real time ("have a deadline")
- ...often have I
- ...run on special purpose hardware

Embedded

Stephan Heidinge

Embedded Systems Design

patterns

Real-time

• Embedded Systems: ...

- ...respond to physical world
- ...respond in real time ("have a deadline")
- ...often have little resources
- ...run on special purpose hardware
- ...run in real-time operating system

Embedded Systems - What's that? - II

Embedded Systems

—Embedded Systems - What's that? - III

Embedded Systems - What's that? - III

» Examples for Embedded Systems:

Embedded Systems - What's that? - III

Embedded Systems

Stephan Heidinge

Embedded Systems Desig

patterns

Real-time

- airbag
- cell phone / 'modern' phone
- burglar alarm
- (fully automatic) coffee machine
- danger detection
- . . .

Embedded Systems

—Embedded Systems - What's that? - III

Embedded Systems - What's that? - III

Examples for Embedded Systems:
 airbag

Embedded Systems - What's that? - III

Embedded Systems

Stephan Heidinge

Embedded Systems Desig

patterns

Real-time

- airbag
- cell phone / 'modern' phone
- burglar alarm
- (fully automatic) coffee machine
- danger detection
- . . .

Embedded Systems

Embedded Systems - What's that? - III

Embedded Systems - What's that? - III

• Examples for Embedded Systems:
• airbag
• cell phone / "modern" phone

Embedded Systems - What's that? - III

Embedded Systems

Stephan Heidinge

Embedded Systems Desig

patterns

Real-time

- airbag
- cell phone / 'modern' phone
- burglar alarm
- (fully automatic) coffee machine
- danger detection
- . . .

Embedded Systems

Embedded Systems - What's that? - III

-Embedded Systems - What's that? - III

Embedded Systems - What's that? - III

Embedded Systems

Stephan Heidinge

Embedded Systems Desig

cell phone / 'modern' phone
 burglar alarm

patterns

Real-time operating system

- Examples for Embedded Systems:
 - airbag
 - cell phone / 'modern' phone
 - burglar alarm
 - (fully automatic) coffee machine
 - danger detection
 - . . .

Embedded Systems

Embedded Systems - What's that? - III

Embedded Systems - What's that? - III

- » Examples for Embedded Systems:
- s cell phone / 'modern' phone
- burglar alarm
 (fully automatic) coffee machine

Embedded Systems - What's that? - III

Embedded Systems

Stephan Heidinger

Embedded Systems Design

patterns

Real-time operating system

- Examples for Embedded Systems:
 - airbag
 - cell phone / 'modern' phone
 - burglar alarm
 - (fully automatic) coffee machine
 - danger detection
 - . . .

Embedded Systems

Embedded Systems - What's that? - III

Embedded Systems - What's that? - III

- a Examples for Embedded Systems:
- cell phone / 'modern' phone
- burglar alarm
- (fully automatic) coffee machine danger detection

Embedded Systems - What's that? - III

- airbag
- cell phone / 'modern' phone
- burglar alarm
- (fully automatic) coffee machine
- danger detection
-

Embedded Systems

Embedded Systems - What's that? - III

Embedded Systems - What's that? - III

- a Examples for Embedded Systems:
- cell phone / 'modern' phone
- burglar alarm
- danger detection
- (fully automatic) coffee machine

Embedded Systems - What's that? - III

- airbag
- cell phone / 'modern' phone
- burglar alarm
- (fully automatic) coffee machine
- danger detection
-

Motivation

Embedded Systems



Motivation

Embedded Systems

Stephan Heiding

Embedded Systems Desig

patterns

Real-time

• We see:

- Embedded Systems are everywhere!
- There are probably more Embedded Systems than computers out there!
- We realize:
 - Man, they must be important
 - There sure is some money in this.

Embedded Systems - Motivation



Motivation

• We see:

- Embedded Systems are everywhere!
- There are probably more Embedded Systems than computers out there!
- We realize:

Motivation

Embedded Systems

Motivation

- We see:
- Embedded Systems are everywhere!
 There are probably more Embedded Systems than computers out there!
- Man, they must be important.

 There sure is some money in this

Motivation

Embedded Systems

Stephan Heidinger

Embedded Systems Desig

patterns

Real-time operating system

- We see:
 - Embedded Systems are everywhere!
 - There are probably more Embedded Systems than computers out there!
- We realize:
 - Man, they must be important
 - There sure is some money in this.

Embedded Systems

- Motivation

Motivation

We see:

▶ Embedded Systems are everywhere!

There are probably more Embedded Systems than computers out there!

Motivation

• We see:

- Embedded Systems are everywhere!
- There are probably more Embedded Systems than computers out there!
- We realize:
 - Man, they must be important.
 - There sure is some money in this.

Embedded Systems

- Motivation

Motivation

- We see:
- Embedded Systems are everywhere!
 There are probably more Embedded Systems than computers out there!
- We realize:
 Man, they must be important.

Motivation

Embedded Systems

Stephan Heidinger

Embedded Systems Desig

patterns

Real-time operating syster

- We see:
 - Embedded Systems are everywhere!
 - There are probably more Embedded Systems than computers out there!
- We realize:
 - Man, they must be important.
 - There sure is some money in this.

Embedded Systems

O-IOMotivation

Motivation

- We see:
- Embedded Systems are everywhere!
 There are probably more Embedded Systems than computers out there!
- We realize:

 Man, they must be important.

 There sure is some money in this.

Motivation

Embedded Systems

Stephan Heidinger

Embedded Systems Desig

patterns

Real-time operating syster

- We see:
 - Embedded Systems are everywhere!
 - There are probably more Embedded Systems than computers out there!
- We realize:
 - Man, they must be important.
 - There sure is some money in this.

Embedded Systems

Motivation

Motivation

- We see:
 Embedded Systems are everywhere!
 There are probably more Embedded Systems than computers out there!
- a We realize:
- Man, they must be important.
 There sure is some money in this.
 I did an internship producing an embedded system.

Motivation

Embedded Systems

Stephan Heidinger

Embedded Systems Desig

patterns

_ . .

• We see:

- Embedded Systems are everywhere!
- There are probably more Embedded Systems than computers out there!
- We realize:
 - Man, they must be important.
 - There sure is some money in this.
 - I did an internship producing an embedded system.

Embedded Systems
Outline

Outline

- Embedded Systems Design
- Architectural patterns
- Timing analysis
- Real-time operating systems

Outline

mbedded Systems

Stephan Heidinge

Embedded Systems Design

patterns

Timing analysi

Embedded Systems Design

2 Architectural patterns

3 Timing analysis

4 Real-time operating systems

Embedded Systems Design
 Architectural partners

Outline

Outline

Embedded Systems

Stephan Heidinger

Embedded Systems Design

Timing analys

Real-time

- Embedded Systems Design
- 2 Architectural patterns
- 3 Timing analysis
- 4 Real-time operating systems

5 / 22

Embedded Systems Design

Embedded Systems Design

Problems

Problems in embedded Systems:

Problems

Problems

Embedded Systems

Stephan Heidinge

Embedded Systems Design

patterns

I iming analys

- deadlines
- environment
- continuity
- direct hardware interaction
- safety & reliability



Problems

Problems in embedded Systems:

deadlinss

deadlines: every process has deadline until result must exist

hard systems: deadline not met, failure soft system: deadline not met, bad results

Problems

Embedded Systems

Stephan Heidinge

Embedded Systems Design

patterns

Tilling allalysis

- deadlines
- environment
- continuity
- direct hardware interaction
- safety & reliability

Embedded Systems

Embedded Systems Design

Problems

Problems • Problems in embedded Systems: • duadfines • embeddemsest

environment:

2012-01-04

is unpredictable embedded Software \Rightarrow must be concurrent

Problems

Embedded Systems

Stephan Heidinger

Embedded Systems Design

patterns

_ . .

- deadlines
- environment
- continuity
- direct hardware interaction
- safety & reliability

Embedded Systems

Embedded Systems Design

Problems

Problems • Problems in embedded Systems: • deadlines • environment • continuity

continuity:

2012-01-04

embedded Software \Rightarrow does not normally terminate software has to be reliable may need update while operating

Problems

Embedded Systems

Stephan Heidinger

Embedded Systems Design

patterns

Deal since

- deadlines
- environment
- continuity
- direct hardware interaction
- safety & reliability

Embedded Systems

Embedded Systems Design

Problems

2012-01-04

Problems

- Problems in embedded Systems:
 deadlines
 environment
- deadlines
 environment
 continuity
 direct hardware interaction

direct hardware interaction:

uncommon hardware (i.e. detonator in airbag) speed issues (hardware is faster)

Problems

Embedded Systems

Stephan Heidinger

Embedded Systems Design

atterns -.

I iming analysis

- deadlines
- environment
- continuity
- direct hardware interaction
- safety & reliability

Problems

- Problems in embedded Systems:
- environment
- continuity
 direct hardware interaction
 safety & reliability

safety & reliability:

cost of failure high either economical or in human life

next thing:

2012-01-04

Design steps not all are necessary, but most will be.

Problems

Embedded Systems

Stephan Heidinger

Embedded Systems Design

patterns

Timing analysi

operating system

- Problems in embedded Systems:
 - deadlines
 - environment
 - continuity
 - direct hardware interaction
 - safety & reliability

Embedded Systems Design

-Embedded Systems Design - I



Embedded Systems Design - I

Embedded Systems

Stephan Heidinge

Embedded Systems Design

patterns

i iming analys

- design steps
 - platform selection
 - special purpose hardware
 - stimuli:
 - periodic stimul
 - aperiodic stimul

```
Embedded Systems

Embedded Systems Design

Embedded Systems Design - I
```



Platform selection:

what hardware?
Real-time operating system (later)
What is to be implemented in software, what in hardware need to design special hardware?
power consumption (mobile device, backup)

Embedded Systems Design - I

mbedded Systems

Stephan Heiding

Embedded Systems Design

Architectu patterns

Timing analysi

Real-time operating system

- design steps
 - platform selection
 - special purpose hardware
 - stimuli:
 - periodic stimul
 - aperiodic stimul

Embedded Systems Design

Embedded Systems Design - I

special purpose hardware:

do we need special hardware? design special hardware? replace software by hardware?

Embedded Systems Design - I a design steps platform salection special purposes hardware

Embedded Systems Design - I

Embedded Systems

Stephan Heidinger

Embedded Systems Design

patterns

Timing analysi

Real-time operating systems

- design steps
 - platform selection
 - special purpose hardware
 - stimuli:
 - periodic stimul
 - aperiodic stimul

Embedded Systems

Embedded Systems Design

Embedded Systems Design - I

Embedded Systems Design - I

a design stage
y platform selection
y data propose hardware
y stimuli:

stimuli:

describe behavior of system by listing received stimuli and reactions $\mathsf{stimuli} = \mathsf{signals}$

Embedded Systems Design - I

mbedded Systems

Stephan Heidinge

Embedded Systems Design

patterns

I iming analysi

- design steps
 - platform selection
 - special purpose hardware
 - stimuli:
 - periodic stimuli
 - 2 aperiodic stimuli

Embedded Systems Design

Embedded Systems Design - I

periodic stimuli:

occur at predictable intervals predefined reaction per stimulus i.e. polling

Embedded Systems Design - I

a design steps
• platform selection
• special purpose hardware
• stimuli:
• periodic stimuli

Embedded Systems Design - I

Embedded Systems

Stephan Heidinger

Embedded Systems Design

Architectur patterns

Tilling analysis

- design steps
 - platform selection
 - special purpose hardware
 - stimuli:
 - periodic stimuli
 - 2 aperiodic stimuli

```
Embedded Systems

Embedded Systems Design

Embedded Systems Design - I
```

```
Embedded Systems Design - I

design steps

d
```

aperiodic stimuli:

occurr irregularly and unpredictably often interrupts i.e. alarms, failures, IO operation finished, etc**stimuli list:** best practice: stimuli list with **all** stimuli.

example next slide

Embedded Systems Design - I

mbedded Systems

Stephan Heidinger

Embedded Systems Design

Architectu patterns

Timing analysi

- design steps
 - platform selection
 - special purpose hardware
 - stimuli:
 - periodic stimuli
 - 2 aperiodic stimuli

Example: radiation warning system

Embedded Systems

Stephan Heidinge

Embedded Systems Design

Archite patteri

Example: radiation warning system

Timing analysi

Embedded Systems

Embedded Systems Design

Example: radiation warning system



Example: radiation warning system

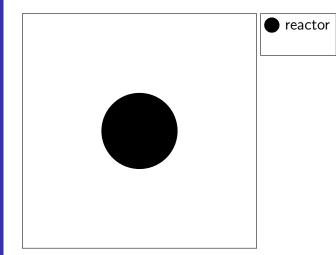
Embedded Systems

Stephan Heidinge

Embedded Systems Design

patterns

Real-time



Embedded Systems

Embedded Systems Design

Example: radiation warning system



Example: radiation warning system

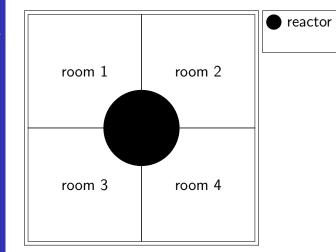
Embedded Systems

Stephan Heidinger

Embedded Systems Design

patterns

Real-time



Embedded Systems -Embedded Systems Design

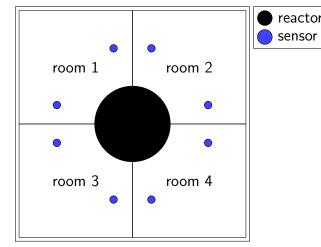
Example: radiation warning system



Example: radiation warning system

Example: radiation warning system

Embedded Systems Design



reactor

Embedded Systems

Embedded Systems Design

Example: Stimuli-List of a radiation warning system

Example: Stimuli-List of a radiation warning system

timulus Response

Example: Stimuli-List of a radiation warning system

Embedded Systems

Stephan Heidinger

Embedded Systems Design

Archited patterns

Timing analysis

Real-time operating systems

Stimulus Response

Embedded Systems
Embedded Systems Design

Example: Stimuli-List of a radiation warning system

Example: Stimuli-List of a radiation warning system

Stimulus Response single sensor positive flash yellow light around sensor

Example: Stimuli-List of a radiation warning system

Embedded Systems

Stephan Heidinger

Embedded Systems Design

atterns

Real time

StimulusResponsesingle sensor positiveflash yellow light around sensor

Embedded Systems Design

Example: Stimuli-List of a radiation warning system

Example: Stimuli-List of a radiation warning system

Stimulus Response single sensor positive flash yellow light around sensor both sensors in one flash red light around sensor area positive sound acoustic alarm aroun sensor

Example: Stimuli-List of a radiation warning system

mbedded Systems

Stephan Heidinge

Embedded Systems Design

Timine analys

Real-time operating sy

Stimulus	Response
single sensor positive	flash yellow light around sensor
both sensors in one	flash red light around sensor,
area positive	sound acoustic alarm around
	sensor

Embedded Systems -Embedded Systems Design

Example: Stimuli-List of a radiation warning system

Example: Stimuli-List of a radiation warning system

Example: Stimuli-List of a radiation warning system

Embedded Systems Design

Stimulus	Response
single sensor positive	flash yellow light around sensor
both sensors in one	flash red light around sensor,
area positive	sound acoustic alarm around
	sensor
Voltage drop of 10-	switch to backup power; run
20%	power supply test

Embedded Systems

Embedded Systems Design

Example: Stimuli-List of a radiation warning system

Example: Stimuli-List of a radiation warning system

Response ingle sensor positive flash yellow light around sensor positive flash yellow light around sensor positive flash yellow light around sensor area positive sound acoustic alarm area sensor which to backup power; in power supply sent of the power of the power

Example: Stimuli-List of a radiation warning system

mbedded Systems

Stephan Heidinge

Embedded Systems Design

Timing analysis

Real-time operating syst

Stimulus	Response
single sensor positive	flash yellow light around sensor
both sensors in one	flash red light around sensor,
area positive	sound acoustic alarm around
	sensor
Voltage drop of 10-	switch to backup power; run
20%	power supply test
Voltage drop of more	switch to backup; run power
than 20%	supply test; call maintainer

Embedded Systems Design

Embedded Systems Design - II

Embedded Systems Design - II

s design steps - continued

Embedded Systems Design - II

Embedded Systems

Stephan Heidinge

Embedded Systems Design

patterns

I iming analy

design steps - continued

- Timing analysis
- Process design
- Algorithm design
- Data design
- Process scheduling

Embedded Systems

Embedded Systems Design

Embedded Systems Design - II

Embedded Systems Design - II s design steps - continued s Timing analysis

Timing analysis:

For each stimulus and response \Rightarrow find timing constraints timing constraints \Rightarrow deadlines

Embedded Systems Design - II

Embedded Systems

Stephan Heidinge

Embedded Systems Design

patterns

- . . .

- design steps continued
 - Timing analysis
 - Process design
 - Algorithm design
 - Data design
 - Process scheduling

Embedded Systems Design

Embedded Systems Design - II



Process design:

aggregate the stimuli & responses into concurrent processes See Architectural design

Embedded Systems Design - II

Embedded Systems

Stephan Heidinge

Embedded Systems Design

patterns _.

i iiiiiig aliaiysi

- design steps continued
 - Timing analysis
 - Process design
 - Algorithm design
 - Data design
 - Process scheduling

Embedded Systems Embedded Systems Design Embedded Systems Design - II

Embedded Systems Design - II s design steps - continued Trining analysis Algorithm design

Algorithm design:

For each stimulus & response \Rightarrow design algorithm especially important for computationally intensive tasks (signal processing)

Do we need to implement these in hardware?

Embedded Systems Design - II

Embedded Systems

Stephan Heidinge

Embedded Systems Design

patterns

iiiiiig aliaiysis

- design steps continued
 - Timing analysis
 - Process design
 - Algorithm design
 - Data design
 - Process scheduling

Embedded Systems Embedded Systems Design Embedded Systems Design - II

```
Embedded Systems Design - II

, design steps - continued

Timing analysis

Process design

Out design
```

Data design:

How to store data, that will be exchanged semaphore & critical regions & monitors & . . .

circular buffer: producer & consumer may run at different speeds

Embedded Systems Design - II

Embedded Systems

Stephan Heidinge

Embedded Systems Design

patterns

. . .

- design steps continued
 - Timing analysis
 - Process design
 - Algorithm design
 - Data design
 - Process scheduling

Embedded Systems Embedded Systems Design

Embedded Systems Design - II

Embedded Systems Design - II

- design steps continued Timing analysis
 - Timing analysis Process design
- Algorithm design
 Data design
 Process scheduling

Process scheduling:

ensure, that processes meet their deadline probably among the hardest (own opionion) all shown:

not all need to be done, but most probably will which & order depends on what we design

after this design steps:

make sure system can meet deadlines static analysis simulation

Embedded Systems Design - II

Embedded Systems

Stephan Heidinger

Embedded Systems Design

patterns

Real-time

- design steps continued
 - Timing analysis
 - Process design
 - Algorithm design
 - Data design
 - Process scheduling

Embedded Systems Design

Embedded Systems Design

Embedded system modeling



Embedded system modeling

Embedded Systems

Stephan Heidinge

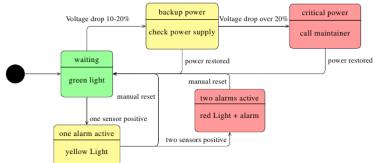
Embedded Systems Design

patterns

Real-time

• Embedded Systems are often built as state machines.

⇒ UML state diagrams



Embedded Systems Design

Embedded Systems Design

Embedded system modeling



Embedded system modeling

Embedded Systems

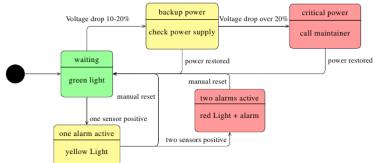
Stephan Heidinge

Embedded Systems Design

Timing analys

Real-time operating system • Embedded Systems are often built as state machines.

⇒ UML state diagrams



Embedded Systems Design

Embedded software programing

Embedded software programing

programm has to be...

 Consumer to a Consumer to

Embedded software programing

Embedded Systems

Stephan Heidinge

Embedded Systems Design

patterns

Timing analys

• programm has to be...

```
• ... fast (i.e. C, Asssembler)
```

speed looses importance

Embedded Systems

Embedded Systems Design

Embedded software programing

C, Assembler:

No concurrency no built-in system for shared resources



Embedded software programing

mbedded Systems

Stephan Heidinger

Embedded Systems Design

patterns

Timing analysi

Real-time operating syster

- programm has to be...
 - ... fast (i.e. C, Asssembler)
 - ... concurrent (i.e. C++, real time Java, ...)
- speed looses importance

Embedded software programing

Embedded software programing

programm has to be...

....concurrent (i.e. C++, real time Java, ...)

concurrent:

and manage shared resources

concurrent or speed??:

depends on what is more important simulate concurrency with frequent polling do something yourself about shared resources

Embedded software programing

mbedded Systems

Stephan Heidinger

Embedded Systems Design

patterns

Timing analysi

Real-time operating system

- programm has to be...
 - ... fast (i.e. C, Asssembler)
 - ... concurrent (i.e. C++, real time Java, ...)
- speed looses importance

Embedded Systems -Embedded Systems Design Embedded software programing

speed:

due to faster hardware ie monitoring device written in C++ ie cell phones in java, objective C, ...

Embedded software programing

- ... concurrent (i.e. C++, real time Java, ...)

Embedded software programing

Embedded Systems Design

- programm has to be...
 - ... fast (i.e. C, Asssembler)
 - ... concurrent (i.e. C++, real time Java, ...)
- speed looses importance

Outline

Embedded Systems

Stephan Heidinger

Embedded
Systems Desig

Patterns

Timing analysi

Real-time operating system

- 1 Embedded Systems Design
- 2 Architectural patterns
- 3 Timing analysis
- 4 Real-time operating systems

Embedded Systems
Architectural patterns

-Architectural patterns

Architectural patterns

• Anohitectural patterns are used to describe a system

in an abstract way and help to understand the
architecture.

note on 3:

The source described three rough design pattern there are finer patterns

Architectural patterns

Embedded Systems

Stephan Heidinger

Embedded
Systems Design
Architectural

patterns

i iming analysis

 Architectural patterns are used to describe a system in an abstract way and help to understand the architecture.

- Observe and react
- Environmental Control
- Process Pipeline

Embedded Systems
Architectural patterns
Architectural patterns

 Architectural patterns are used to describe a system in an abstract way and help to understand the architecture.
 Observe and react

Architectural patterns

Observe and React:

set of monitored sensors Something happens *Rightarrow* we do something ie incoming phone call

Architectural patterns

Embedded Systems

Stephan Heidinge

Embedded Systems Desig Architectural

patterns

Timing analysis

 Architectural patterns are used to describe a system in an abstract way and help to understand the architecture.

- Observe and react
- Environmental Control
- Process Pipeline

Embedded Systems -Architectural patterns -Architectural patterns

Environmental Control:

set of sensors and actuators can change environment ie flash light, when sensor fires Architectural patterns

a Architectural patterns are used to describe a system in an abstract way and help to understand the Observe and react A Environmental Control

Architectural patterns

Architectural

patterns

• Architectural patterns are used to describe a system in an abstract way and help to understand the architecture.

- Observe and react
- Environmental Control
- Process Pipeline

Embedded Systems
Architectural patterns
Architectural patterns

Architectural patterns are used to describe a system
 in an abstract way and help to understand the

Observe and react
 Freeinnemental Control

Architectural patterns

Process Pipeline:

data transformation series of processing steps

preferably concurrent

all of those: can be combined

often more than one pattern in the system

ie monitor the actuators design patterns: will lead to inefficient

system Rightarrow only for understanding system

Architectural patterns

Embedded Systems

Stephan Heidinge

Embedded Systems Design Architectural

. Timing analysis

patterns

Real-time

 Architectural patterns are used to describe a system in an abstract way and help to understand the architecture.

- Observe and react
- Environmental Control
- Process Pipeline

Embedded Systems

Architectural patterns

Observe and React

Observe and React

Observe and React

Observe and React

Embedded Systems

Stephan Heiding

Embedded
Systems Desig

patterns

Timing analys

Observe and React

- monitor the system with a set of sensors
- display something
- primarly used in: Monitoring systems

Embedded Systems

Architectural patterns

Observe and React

monitoring: as stated before

Observe and React

Observe and React
 monitor the system with a set of sensors

Observe and React

mbedded Systems

Stephan Heidinge

Embedded Systems Design Architectural

patterns

Tilling allarys

Observe and React

- monitor the system with a set of sensors
- display something
- primarly used in: Monitoring systems

Embedded Systems

Architectural patterns

Observe and React

Observe and React

Observe and React
 monitor the system with a set of sensors
 display something

display:

monitoring screen on exceptional behaviour: alarms, shutdown

Observe and React

Embedded Systems

Stephan Heidinge

Embedded Systems Design

Architectural patterns

I iming analysis

Observe and React

- monitor the system with a set of sensors
- display something
- primarly used in: Monitoring systems

Embedded Systems

Architectural patterns

Observe and React

Observe and React
 monitor the system with a set of sensors
 display something
 primarly used in: Monitoring systems

Observe and React

monitoring systems:

often consist of more than one O&R patterns, one for each sensor optimisation: combine something, ie display on one monitor

Observe and React

mbedded Systems

Stephan Heidinger

Embedded
Systems Desig

Architectural

patterns

Timing analysis

operating syster

- Observe and React
 - monitor the system with a set of sensors
 - display something
 - primarly used in: Monitoring systems

Embedded Systems -Architectural patterns

Environmental Control

Environmental Control Environmental Control

Environmental Control

Architectural patterns

Environmental Control

- monitor the system and react to any changes
- Used when there is no requirement for user
- ... or no time for the user to interact ...
- ... no way a user can interact ...
- ... or there is too much information for users to process.



Embedded Systems -Architectural patterns

Environmental Control

Environmental Control

 Environmental Control · monitor the system and react to any changes

Environmental Control

Architectural

patterns

Environmental Control

- monitor the system and react to any changes
- Used when there is no requirement for user
- ... or no time for the user to interact ...
- ... no way a user can interact ...
- ... or there is too much information for users to

Embedded Systems -Architectural patterns

Environmental Control

examples:

cruise control water level pressure control

. . .

Environmental Control

- Environmental Control · monitor the system and react to any changes

Environmental Control

Architectural patterns

Environmental Control

- monitor the system and react to any changes
- Used when there is no requirement for user interaction...
- ... or no time for the user to interact ...
- ... no way a user can interact ...
- ... or there is too much information for users to

Embedded Systems Architectural patterns

Environmental Control

examples:

break assist airbag

Environmental Control

- Environmental Control
 monitor the system and
- monitor the system and react to any changes
 Used when there is no requirement for user
- ... or no time for the user to interact ...

interaction... ... or no time for the user to interact ...

Environmental Control

Embedded Systems

Stephan Heidinge

Embedded Systems Desig Architectural

patterns

Real-time

Environmental Control

- monitor the system and react to any changes
- Used when there is no requirement for user interaction...
- ... or no time for the user to interact ...
- ... no way a user can interact ...
- ... or there is too much information for users to process.

Embedded Systems -Architectural patterns

Environmental Control

example:

CYPRES (parachute, Möllemann did not activate his in 2003) self desctruct of military/sensitive equipment

Environmental Control

- Environmental Control
- · monitor the system and react to any changes

Environmental Control

Architectural

patterns

Environmental Control

- monitor the system and react to any changes
- Used when there is no requirement for user interaction...
- ... or no time for the user to interact ...
- ... no way a user can interact ...
- ... or there is too much information for users to

Embedded Systems -Architectural patterns

Environmental Control

example:

Nuclear Power Plant Airplane Car virtually any big system with many subsystems

Environmental Control

- Environmental Control
- · monitor the system and react to any changes

Environmental Control

Architectural patterns

Environmental Control

- monitor the system and react to any changes
- Used when there is no requirement for user interaction...
- ... or no time for the user to interact ...
- ... no way a user can interact ...
- ... or there is too much information for users to process.

Embedded Systems
Architectural patterns
Process Pipeline

Process Pipeline

Process Pipeline

Process Pipeline

Embedded Systems

Stephan Heidinge

Embedded
Systems Desig

patterns
Timing analy

Real-time

Process Pipeline

- transform data
- often huge amounts of data to be converted in real time
- data aquisition system: storing of data may need to be fast

Embedded Systems
Architectural patterns
Process Pipeline

examples:

signal processing from sensors in other systems optical sensor convert digital data to audio

Process Pipeline * Process Pipeline * transform data

Process Pipeline

Embedded Systems

Stephan Heidinge

Embedded Systems Design Architectural

patterns

Real-time

Process Pipeline

- transform data
- often huge amounts of data to be converted in real time
- data aquisition system: storing of data may need to be fast

Embedded Systems
Architectural patterns
Process Pipeline

huge amount:

 ${\sf concurrency} + {\sf multicore} \ {\sf is} \ {\sf the} \ {\sf key}$



Process Pipeline

Embedded Systems

Stephan Heidinger

Embedded Systems Desig Architectural

patterns

Real-time

- Process Pipeline
 - transform data
 - often huge amounts of data to be converted in real time
 - data aquisition system: storing of data may need to be fast

Embedded Systems
Architectural patterns
Process Pipeline

example:

particle accelerator chemical reactions

. . .

if storing not fast, data will be lost

Process Pipeline

- Process Pipeline
- transform data
- often huge amounts of data to be converted in real
- data aquisition system: storing of data may need
 to be fact.

Process Pipeline

Embedded Systems

Stephan Heidinge

Embedded Systems Desig

patterns

Real-time

- Process Pipeline
 - transform data
 - often huge amounts of data to be converted in real time
 - data aquisition system: storing of data may need to be fast



Outline

Embedded Systems

Stephan Heidinger

Embedded Systems Desig

Timing analysis

,g analys

- Embedded Systems Design
- 2 Architectural patterns
- 3 Timing analysis
- 4 Real-time operating systems

Embedded Systems

Timing analysis

Timing Analysis - I

Timing Analysis - I

Timing Analysis - I

Embedded Systems

Stephan Heiding

Embedded Systems Desig

Timing analysis

Real-time operating system

timing analysis

- Correctness of systems depends not only on result, but also on the time at which the result is produced.
- How often does each process need to be executed?
- ullet aperiodic stimuly \Rightarrow make assumptions

Embedded Systems —Timing analysis ☐ Timing Analysis - I

Timing Analysis - I

Correctness of systems depends not only on result.

Timing Analysis - I

Timing analysis

- timing analysis
 - Correctness of systems depends not only on result, but also on the time at which the result is produced.
 - How often does each process need to be executed?
 - aperiodic stimuly ⇒ make assumptions

Timing Analysis - I

s timing analysis

correctness of systems depends ned only on result, but all an on the time at which the result is produced.

How other does such process need to be executed?

how often?:

then we check, if our system can deliver this this can be quite hard, when *mixture of aperiodic and periodic* stimuli or many aperiodic stimuli are expected

Timing Analysis - I

Embedded Systems

Stephan Heidinger

Embedded Systems Design

Timing analysis

Real-time

timing analysis

- Correctness of systems depends not only on result, but also on the time at which the result is produced.
- How often does each process need to be executed?
- aperiodic stimuly ⇒ make assumptions

Embedded Systems Timing analysis Timing Analysis - I

fast systems:

use only periodic stimuli poll frequently for aperiodic stimuli

Timing Analysis - I

- . Correctness of systems depends not only on result.
- . How often does each process need to be executed?

Timing Analysis - I

Timing analysis

- timing analysis
 - Correctness of systems depends not only on result, but also on the time at which the result is produced.
 - How often does each process need to be executed?
 - aperiodic stimuly ⇒ make assumptions

Embedded Systems

Timing analysis

Timing Analysis - II

Timing Analysis - II

a Consider:

Timing Analysis - II

Embedded Systems

Stephan Heiding

Embedded Systems Desig

Timing analysis

Real-time operating system

Consider:

- deadlines
- frequency
- execution time

Embedded Systems
Timing analysis
Timing Analysis - II

Timing Analysis - II

a Consider:
b deadlines

deadlines:

By which time must the process have ended.

Timing Analysis - II

mbedded Systems

Stephan Heidinge

Embedded Systems Design

Timing analysis

- Consider:
 - deadlines
 - frequency
 - execution time

Embedded Systems
Timing analysis
Timing Analysis - II

Timing Analysis - II

a Consider:
b deadlines
frequency

frequency:

The number of times a process must be executed in a given span, so that the *system* meets all deadlines

Timing Analysis - II

mbedded Systems

Stephan Heidinger

Embedded Systems Design

Timing analysis

- Consider:
 - deadlines
 - frequency
 - execution time

```
Embedded Systems

Timing analysis

Timing Analysis - II
```

Consider:
 deadlines
 frequency
 execution time

Timing Analysis - II

execution time:

How long does each single process take (average & worst case) hard: conditional execution, delays waiting, . . .

hard systems: always worst case

Timing Analysis - II

Embedded Systems

Stephan Heidinger

Embedded Systems Design

Timing analysis

- Consider:
 - deadlines
 - frequency
 - execution time



Embedded Systems

Stephan Heiding

Embedded Systems Design

Timing analysis

Real-time

Stimulus/ResponseTiming requirementsvoltage dropswitch to backup: 50ms



Embedded Systems

Stephan Heiding

Embedded Systems Desig

Timing analysis

Stimulus/Response	Timing requirements
voltage drop	switch to backup: 50ms
each sensor	poll twice a second





Stephan Heidinge

Embedded Systems Desig

Timing analysis

Stimulus/Response	Timing requirements
voltage drop	switch to backup: 50ms
each sensor	poll twice a second
turn on light	500ms



Stimulus/Response	Timing requirements
voltage drop	switch to backup: 50ms
each sensor	poll twice a second
turn on light	500ms
call maintainer	5000ms

Embedded Systems

Stephan Heidinge

Embedded Systems Desig

Timing analysis

Stimulus/Response	Timing requirements
voltage drop	switch to backup: 50ms
each sensor	poll twice a second
turn on light	500ms
call maintainer	5000ms

Outline

Outline

Embedded Systems

Stephan Heidinger

Embedded Systems Design

patterns

Timing analysis

Real-time operating systems

- Embedded Systems Design
- 2 Architectural patterns
- 3 Timing analysis
- 4 Real-time operating systems

Embedded Systems Real-time operating systems Real-time operating systems

too large, too bulky, too slow

normal operating systems:



Real-time operating systems

mbedded Svstems

Stephan Heidinger

Embedded Systems Design

patterns --- · · · · · · ·

Real-time operating systems

- nromal operating systems not feasible
- special "real-time operating systems" exist
- RTOS must include:
 - real-time clock
 - interrupt handler
 - process manager: scheduler & resource manager
 - dispatcher

Embedded Systems

Real-time operating systems

Real-time operating systems

real-time operating systems:

Windows/CE Vxworks RTLinux emdebian

they are small and damn fast



Real-time operating systems

Embedded Systems

Stephan Heidinger

Embedded Systems Design

patterns

- nromal operating systems not feasible
- special "real-time operating systems" exist
- RTOS must include:
 - real-time clock
 - interrupt handler
 - process manager: scheduler & resource manage
 - dispatcher

Embedded Systems Real-time operating systems

Real-time operating systems

Real-time operating systems nromal operating systems not feasible special "natime operating systems" code: « RTOS must include:

Real-time operating systems

Embedded Systems

Stephan Heidinger

Embedded Systems Desig

patterns

- nromal operating systems not feasible
- special "real-time operating systems" exist
- RTOS must include:
 - real-time clock
 - interrupt handler
 - process manager: scheduler & resource manager
 - dispatcher

Embedded Systems

Real-time operating systems

Real-time operating systems

Real-time operating systems

* monal operating systems not feable

* special "mail-time operating systems" exist

* ITO'S must include:

* real-time clock

real-time clock:

provides information required to schedule processes

Real-time operating systems

Embedded Systems

Stephan Heidinger

Embedded Systems Design

Tarterns

- nromal operating systems not feasible
- special "real-time operating systems" exist
- RTOS must include:
 - real-time clock
 - interrupt handler
 - process manager: scheduler & resource manager
 - dispatcher

Embedded Systems

Real-time operating systems

Real-time operating systems

Real-time operating systems a mornal operating systems nor fasable a special "real-time operating systems" exist a RTOS must include: - real-time contains - real-time contains

interrupt handler:

manages aperiodic requests for service may be inside process manager at least **2 levels:** *interrupt* for processes with fast response time & *clock level* fore regular processes often also background processes with low priority (self checks etc)

Real-time operating systems

Embedded Systems

Stephan Heidinger

Embedded Systems Design

-- · · ·

- nromal operating systems not feasible
- special "real-time operating systems" exist
- RTOS must include:
 - real-time clock
 - interrupt handler
 - process manager: scheduler & resource manager
 - dispatcher

Embedded Systems

Real-time operating systems

Real-time operating systems

Real-time operating systems

* monal operating systems not feasible
* special "real-time operating systems" exist
* RTOS must include:
* real-time foots

process manager: scheduler & resource manager

scheduler

examines processes and chooses one for execution processes need enough processor time to *finish before their deadline* **commonly used:**

non-pre-emptive & pre-emtive (execution of processes may be stopped)
round robin
rate monolithic scheduling (SJF)
shortes deadline first (HPF) resource manager:
allocates memory and processor resources scheduled for execution

Real-time operating systems

mbedded Systems

Stephan Heidinge

Embedded Systems Design

Timing analysis

- nromal operating systems not feasible
- special "real-time operating systems" exist
- RTOS must include:
 - real-time clock
 - interrupt handler
 - process manager: scheduler & resource manager
 - dispatcher

Embedded Systems Real-time operating systems Real-time operating systems

dispatcher:

starts execution of processes

Real-time operating systems

- a nromal operating systems not feasible
- special "real-time operating systems" exist
- · real-time clock
- · process manager: scheduler & resource manager dispatcher

Real-time operating systems

- nromal operating systems not feasible
- special "real-time operating systems" exist
- RTOS must include:
 - real-time clock
 - interrupt handler
 - process manager: scheduler & resource manager
 - dispatcher

Embedded Systems

Real-time operating systems

30 minutes in short

30 minutes in short

» What you should (at least)remember:

30 minutes in short

Embedded Systems

Stephan Heidinger

Embedded Systems Desig

patterns

Real-time operating systems

• What you should (at least)remember:

- Embedded Systems react to events in real time.
- Embedded Systems are a set of processes reating to stimuli
- State models help understanding the System.
- Architectural patterns can be used to help in designing the system.
- Always to timing analysis in (hard) Embedded Systems.



Embedded Systems

Real-time operating systems

30 minutes in short

30 minutes in short

> What you should (at least)remember:

• Embedded System result to events in real time.

30 minutes in short

Embedded Systems

Stephan Heidinger

Embedded Systems Desig

Timing and a

Real-time operating systems

• What you should (at least)remember:

- Embedded Systems react to events in real time.
- Embedded Systems are a set of processes reating to stimuli
- State models help understanding the System.
- Architectural patterns can be used to help in designing the system.
- Always to timing analysis in (hard) Embedded Systems.



Embedded Systems

Real-time operating systems

30 minutes in short

30 minutes in short

What you should (at least)remember:
 Embedded Systems react to events in real time.
 Embedded Systems are a set of processes reating to stimuli

 Architectural patterns can be used to i designing the system.
 Always to timing analysis in (hard) Em

30 minutes in short

mbedded Systems

Stephan Heidinger

Embedded Systems Desig

patterns

Real-time operating systems

- What you should (at least)remember:
 - Embedded Systems react to events in real time.
 - Embedded Systems are a set of processes reating to stimuli
 - State models help understanding the System.
 - Architectural patterns can be used to help in designing the system.
 - Always to timing analysis in (hard) Embedded Systems.



Embedded Systems

Real-time operating systems

30 minutes in short

30 minutes in short

- What you should (at least)remember:
 Embedded Systems react to events in real time.
- Embedded Systems react to events in real time.
 Embedded Systems are a set of processes reating
- State models help understanding the System.

30 minutes in short

Embedded Systems

Stephan Heidinge

Embedded Systems Desig

patterns

- What you should (at least)remember:
 - Embedded Systems react to events in real time.
 - Embedded Systems are a set of processes reating to stimuli
 - State models help understanding the System.
 - Architectural patterns can be used to help in designing the system.
 - Always to timing analysis in (hard) Embedded Systems.



Embedded Systems Real-time operating systems 30 minutes in short

30 minutes in short

- > What you should (at least)remember
- . Embedded Systems react to events in real time. . Embedded Systems are a set of processes reating
- . State models help understanding the System.
- . Architectural patterns can be used to help in
- designing the system.

30 minutes in short

Real-time operating systems • What you should (at least)remember:

- Embedded Systems react to events in real time.
- Embedded Systems are a set of processes reating to stimuli
- State models help understanding the System.
- Architectural patterns can be used to help in designing the system.
- Always to timing analysis in (hard) Embedded



Embedded Systems

Real-time operating systems

30 minutes in short

30 minutes in short

- What you should (at least)remember
 Embedded Surtame most to supply
- Embedded Systems react to events in real time.
 Embedded Systems are a set of processes reating.
- to stimuli
- Architectural patterns can be used to help in
- Architectural patterns can be used to help in designing the system.
 Always to timing analysis in (hard) Embedded
- Always to timing analysis in (hard) Embedded Systems.

30 minutes in short

Embedded Systems

Stephan Heidinger

Embedded Systems Design

patterns

Real-time operating systems

• What you should (at least)remember:

- Embedded Systems react to events in real time.
- Embedded Systems are a set of processes reating to stimuli
- State models help understanding the System.
- Architectural patterns can be used to help in designing the system.
- Always to timing analysis in (hard) Embedded Systems.