**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 analysi

Real-time operating system

# Software Engineering in Embedded Systems

Stephan Heidinger

Seminar: Software Engineering Fachbereich für Informatik und Informationssysteme Universität Konstanz

19. January 2012



Embedded Systems

-Embedded Systems - What's that? - I

Embedded Systems - What's that? - I

Dehention

"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."

\*\*Embedded systems are nominally real-time systems."

\*\*Embedded systems are nominally real-time systems."

# Embedded Systems - What's that? - I

Embedded 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

patter

Timing analys

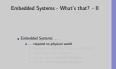
Real-time operating system

#### • 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

Stephan Heidinge

Embedded Systems Desig

patterns

i iiiiiig aliaiys

• 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 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

Stephan Heidinge

Embedded Systems Desig

patterns

I iming analys

• 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
  - 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 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:
- ... respond to physical world
  - respond to physical world
- ... run on special purpose hardware

# Embedded Systems - What's that? - II

#### Embedded Systems

Stephan Heidinge

Embedded Systems Design

-- · · ·

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:
- ... 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? - 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

a Examples for Embedded Systems:
b 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

» Examples for Embedded Systems:

airbag
 cell phone / 'modern' phone
 burglar alarm

# Embedded Systems - What's that? - III

Embedded Systems

Stephan Heidinge

Embedded Systems Desig

patterns

Real-time operating systems

- 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:
- cell phone / 'modern' phone
- burglar alarm
   (fully automatic) coffee machine

# Embedded Systems - What's that? - III

Embedded Systems

Stephan Heidinger

Embedded Systems Desig

patterns

Tilling allalys

- 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
- (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
- . . . .

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!

# Motivation

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

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

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.

└─ Motivation

**Embedded Systems** 

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** 2012-01-04

- Motivation

#### some money:

C-programing special skills

#### 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

#### • 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

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.
  - I did an internship producing an embedded system.

**Embedded Systems** 2012-01-04

-Outline

Outline

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

# Outline

Embedded Systems Design

Architectural patterns

Timing analysis

Real-time operating systems

Outline

Embedded Systems Design

Outline

Embedded Systems

Stephan Heidinger

Embedded Systems Design

Timing analysis

Real-time

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

5 / 21

Embedded Systems Design

Embedded Systems Design

Problems

a Problems in embedded Systems:

Problems

# **Problems**

Embedded Systems

Stephan Heidinge

#### Embedded Systems Design

patterns

Real-time

• Problems in embedded Systems:

- 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

Pool time

• Problems in embedded Systems:

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

Embedded Systems

Embedded Systems Design

Problems

# » Problems in embedded Systems: » deadlinss » environment

Problems

#### environment:

2012-01-04

is unpredictable embedded Software  $\Rightarrow$  must be concurrent

# **Problems**

Embedded Systems

Stephan Heidinger

#### Embedded Systems Design

patterns

Real-time

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

Embedded Systems

Embedded Systems Design

Problems

# Problems • Problems in embedded Systems: • deadlines • annicomment • 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

I iming analysis

Real-time

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

Embedded Systems

Embedded Systems Design

Problems

2012-01-04

Problems

#### direct hardware interaction:

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

# **Problems**

Embedded Systems

Stephan Heidinger

Embedded Systems Design

> atterns .. . . . . .

Tilling allalysis

• Problems in embedded Systems:

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

**Embedded Systems** -Embedded Systems Design -Problems

# safety & reliability:

2012-01-04

cost of failure high either economical or in human life

#### Problems

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

# Problems

#### Embedded Systems Design

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

Embedded Systems

Embedded Systems Design

Embedded Systems Design - I

# next thing:

not all are necessary, but most will be.



# Embedded Systems Design - I

mbedded Systems

Stephan Heidinger

#### Embedded Systems Design

patterns

I iming analys

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

```
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 Heidinge

#### Embedded Systems Design

Architectu patterns

Timing analysi

Real-time

- 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 stops y platform subction y special purpose bardware

# Embedded Systems Design - I

Embedded Systems

Stephan Heidinge

#### Embedded Systems Design

Architectur 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

Embedded Systems Design - I

design steps

platform salection

platform salection

distribution

#### stimuli:

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

# Embedded Systems Design - I

mbedded Systems

Stephan Heidinger

#### Embedded Systems Design

Architectu patterns

I iming analysis

- design steps
  - platform selection
  - special purpose hardware
  - stimuli:
    - periodic stimuli
    - 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
blatform selection
special purpose hardware
stimuli:
periodic stimuli

# Embedded Systems Design - I

Embedded Systems

Stephan Heidinger

#### Embedded Systems Design

Architectur patterns

Timing analysi

- 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

platform selection

special purpose hardware

stems

design steps

platform selection

special purpose hardware

stems

granified stimuti

granified stimuti
```

### 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 Heidinge

#### Embedded Systems Design

Architectu patterns

Real-time

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

Embedded Systems

Embedded Systems Design

Example: radiation warning system

## Example: radiation warning system

Embedded Systems

Stephan Heidinge

Embedded Systems Design

Archite

Example: radiation warning system

Timing analysi

Embedded Systems

Embedded Systems Design

Example: radiation warning system



## Example: radiation warning system

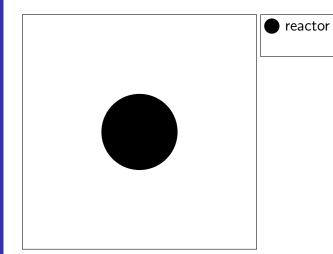
Embedded Systems

Stephan Heidinge

Embedded Systems Design

patterns

Real-time



Embedded Systems

Embedded Systems Design

Example: radiation warning system

room 1 room 2

Example: radiation warning system

## Example: radiation warning system

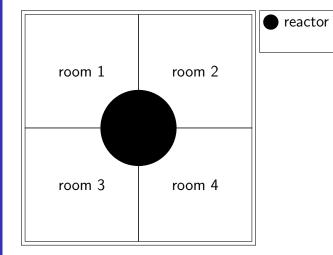
Embedded Systems

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

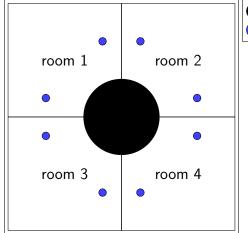
Embedded Systems

Stephan Heidinger

#### Embedded Systems Design

Timing analys

Real-time operating system





Embedded Systems

Embedded Systems Design

Example: Stimuli-List of a radiation warning system

Example: Stimuli-List of a radiation warning system

Stimulus Response

Example: Stimuli-List of a radiation warning system

Embedded Systems

Stephan Heidinge

Embedded Systems Design

Archite patterns

Timing analysis

Real-time operating systems

Stimulus Response

9 / 21

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

i iming analysis

Stimulus Response single sensor positive flash 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 Heidinger

Embedded Systems Design

patterns

Real-time

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 2012-01-04 Example: Stimuli-List of a radiation warning system

LAST CELL:

Example: Stimuli-List of a radiation warning system Embedded Systems Design

Example: Stimuli-List of a radiation warning system

Embedded Systems Design

Stimulus Response single sensor positive flash yellow light around sensor flash red light around sensor, both sensors in one sound acoustic alarm around area positive sensor Voltage drop of 10switch 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

Stimulus Response single sensor positive flash yellow light around sensor sin one area positive area positive sensor sound acoustic alarm around sensor voltage drop of 10-20%. Voltage drop of more which to backup; runn power han 20% supply test; runn power supply test supply test supply test supply test supply test.

## Example: Stimuli-List of a radiation warning system

Embedded Systems

Stephan Heidinger

Embedded Systems Design

patterns

Real-time

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

Real-time

- 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 b design steps - continued b Timing analysis

#### Timing analysis:

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

## Embedded Systems Design - II

mbedded Systems

Stephan Heidinger

#### Embedded Systems Design

patterns

Real-time

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

10 / 21

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 Heidinger

#### Embedded Systems Design

patterns

i iming analysi

- 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 steer - continued s Trining analysis Proceed design 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

mbedded Systems

Stephan Heidinge

#### Embedded Systems Design

patterns

Timing analysi

Real-time

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

10 / 21

# Embedded Systems Embedded Systems Design Embedded Systems Design - II

# Embedded Systems Design - II s design steps - continued \* Trining analysis \* Proces design \* Algorithm design \* Data 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

i iiiiiig ailaiysis

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

## Embedded Systems Embedded Systems Design

Embedded Systems Design - II

## 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

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

## Embedded Systems Design - II

#### Embedded Systems

Stephan Heidinger

#### Embedded Systems Design

patterns

Γiming analys

design steps - continuedTiming analysis

- Process design
- Process design
- Algorithm design
- Data design
- Process scheduling

Embedded Systems Design

Embedded Systems Design

Embedded system modeling



## Embedded system modeling

Embedded Systems

Stephan Heidinge

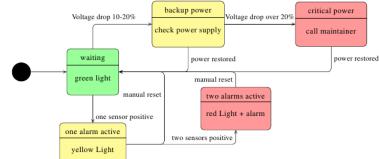
Embedded Systems Design

patterns

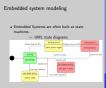
Real-time

• Embedded Systems are often built as state machines.

 $\Rightarrow$  UML state diagrams



**Embedded Systems** Embedded Systems Design Embedded system modeling

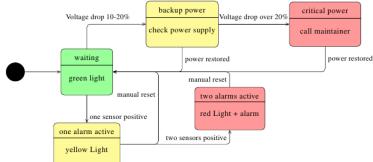


## Embedded system modeling

Embedded Systems Design

• Embedded Systems are often built as state machines.

⇒ UML state diagrams



Embedded Systems Design

Embedded software programing

Embedded software programing

b programm has to be...

## Embedded software programing

Embedded Systems

Stephan Heidinge

#### Embedded Systems Design

patterns

Timing analys

Real-time operating systems

- 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

#### C, Assembler:

No concurrency no built-in system for shared resources



### Embedded software programing

mbedded Systems

Stephan Heidinge

#### Embedded Systems Design

patterns

Timing analysis

• 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

#### 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 Svstems

Stephan Heidinger

#### Embedded Systems Design

patterns

. . . .

- 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

Architectural patterns

Outline

## Outline

Embedded Systems

Stephan Heidinger

Embedded
Systems Desig

patterns

Timing analys

Real-time

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

12 / 21

Embedded Systems
Architectural patterns
Architectural patterns

Architectural patterns

• Architectural 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 Design Architectural

patterns

Fiming 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

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

### Architectural patterns

Embedded Systems

Stephan Heidinger

Embedded
Systems Design
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 are used to describe a system in an abstract way and help to understand the

Observe and react

Architectural patterns

A Environmental Contro

### **Process Pipeline:**

data transformation series of processing steps preferably concurrent

all of those: can be combined

-Architectural patterns

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

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

Observe and React

Observe and React

Observe and React

### Observe and React

mbedded 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

Deal since

Observe and React

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

14 / 21

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

Timing 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 Svstems

Stephan Heidinger

Embedded
Systems Desig

Architectural

patterns

Timing analysis

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**

mbedded Systems

Stephan Heiding

Systems Desig

Timing analys

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.

15 / 21

Embedded Systems

Architectural patterns

-Environmental Control

Environmental Control

Environmental Control
 monitor the system and react to any changes

#### **Environmental Control**

Embedded Systems

Stephan Heidir

Embedded Systems Desig

Architectural patterns

Timing analys

- 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

#### 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

15 / 21

Embedded Systems
Architectural patterns

-Environmental Control

#### examples:

break assist airbag

Environmental Control

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

#### **Environmental Control**

Embedded Systems

Stephan Heidinge

Embedded Systems Desig

Architectural patterns

Real-time

- 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

- a Environmental Control
- monitor the system and react to any changes
   Used when there is no requirement for user
- $\blacktriangleright \ldots$  or no time for the user to interact  $\ldots$

### **Environmental Control**

Embedded Systems

Stephan Heidinger

Embedded Systems Design

Architectural patterns

Tilling analysis

- 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:

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

- 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
 Transform data
 often large amounts of data

## **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

Embedded Systems
Architectural patterns
Process Pipeline

## Process Pipeline \* Process Pipeline \* transform data

### examples:

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

## **Process Pipeline**

Embedded Systems

Stephan Heidinge

Embedded Systems Design Architectural

patterns

Tilling allalysis

### 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 Heidinge

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
- . data aquisition system: storing of data may need

## Process Pipeline

Architectural

patterns

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

Outline

Timing analysis

Timing analysis

I IIIIIIIg aliaiys

- Embedded Systems Design
- 2 Architectural patterns
- 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 not only on result,
but also on the time at which the result is
pendented.

• How of time 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 Heidinge

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 syster

### Consider:

- deadlines
- frequency
- execution time

Embedded Systems
Timing analysis
Timing Analysis - II

Timing Analysis - II

Consider:
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

Embedded Systems

Stephan Heidinger

Embedded Systems Design

Timing analysis

Real-time operating syster

- Consider:
  - deadlines
  - frequency
  - execution time

```
Embedded Systems

Timing analysis

Timing Analysis - II
```

Timing Analysis - II

a Consider.
b deadline
frequency
execution time

### 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 Desig

Timing analysis

Real-time

Stimulus/ResponseTiming requirementsvoltage dropswitch to backup: 50ms



Embedded Systems

Stephan Heidinge

Embedded Systems Design

Timing analysis

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



### LAST CELL:



Stephan Heidinger

Embedded Systems Design

patterns

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



Stephan Heidinge

Embedded Systems Desig

Timing analysis

Real-time

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

## normal operating systems:

too large, too bulky, too slow



## Real-time operating systems

#### mbedded 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 manager
  - dispatcher

Embedded Systems

Real-time operating systems

Real-time operating systems

# Real-time operating systems normal operating systems not facible special "mak-time operating systems" exist special "mak-time operating systems" exist

### 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

\* moroul operating systems not feasible 
\* special "real-time operating systems" exist 
\* KTOS must include:

## Real-time operating systems

Embedded Systems

Stephan Heidinger

Embedded Systems Desig

Timing analys

- 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 nromal operating systems not feasible special "real-time operating systems" exist

### real-time clock:

provides information required to schedule processes

## 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 on feasible second operating systems on feasible special "mail-time operating systems" exist FITO "mail-time dock interrupt handler

### 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

mbedded 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

a monal operating systems not feasible
a special real-time operating systems\* exist
a RTOS must include:

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

Timina analus

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 loas) remember:

• Embedded Systems must to events in real time.

### 30 minutes in short

Embedded Systems

Stephan Heidinge

Embedded Systems Desig

T'...'

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 it designing the system.
 Always to timing analysis in (hard) Em 30 minutes in short

Embedded Systems

Stephan Heidinge

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
- to stimuli

  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
- Architectural patterns can be used to help in designing the system.

### 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.

Embedded Systems

Real-time operating systems

30 minutes in short

30 minutes in short

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

mbedded Systems

Stephan Heidinge

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.