[dcc]

EMBEDDED SYSTEMS INTRODUCTION

Pedro Brandão

Sistemas Embutidos 2020/21

U.PORTO

FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

1

[dcc]

References

- Slides are from Wayne's Wolf Computer as Components 2nd Ed companion overheads, ISBN: 978012374397, Morgan Kaufmann Pub
- Some slides from Prof. Ryan Kastner Embedded System Design class (CSE 237D) class
- Some slides from Edward A. Lee & Sanjit Seshia, UC Berkeley, EECS 149 Fall 2013

U.PORTO

2

[dcc] U.PORTO

What is an Embedded Computing System?

- A miniature computation system developed for low power, high performance devices"
- "A system where different things are brought together to perform a particular application"
- "...defined as any system which does a fixed number of predefined set of tasks with deadlines"
- "A system that is specialized"
- "A system that contains a micro computer controller"

[dcc] U.PORTO

What is an Embedded Computing System? (cont'd)

- "An electronic device with computing capability, but whose main purpose isn't computing (i.e., cellphone, appliance, ..., not a laptop)"
- "Everything I use now is practically an embedded system"
- "A system that users cannot install their own application software on"
- "Miniaturized system that's a combination of HW/SW/Firmware for a specific application or cause."

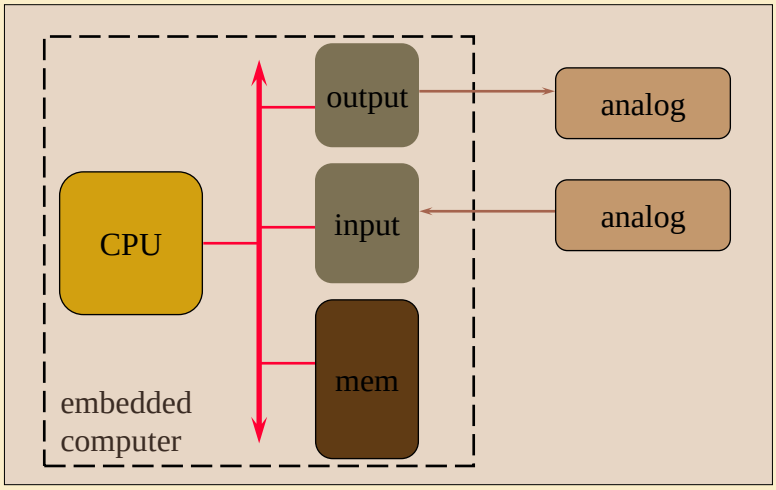
What is an Embedded Computing System? (cont'd)

- "A system designed for a specific task, not a multipurpose computer, which often has restraints such as memory, power, size, cost. Usually all components needed are local (memory, i/o, processor, application specific ICs)
- "Latest computing, signal processing, networking, components packaged into a small form factor. Unique resource and power constraints."
- "A digital system interfacing with hardware."
- "A system on the border between the physical and digital worlds."

Definition

- Embedded computing system: any device that includes a programmable computer but is not itself a general-purpose computer.
- Take advantage of application characteristics to optimize the design:
  - *don't need all the general-purpose bells and whistles.*

Embedding a computer



Examples

- Cell phone.
- Printer.
- Automobile: engine, brakes, dash, etc.
- Airplane: engine, flight controls, nav/comm.
- Digital television.
- Household appliances.

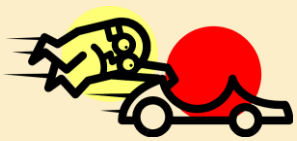
Characteristics of Embedded Systems

- Computational – but not first-and-foremost a computer
- Integral with physical processes – sensors, actuators
- Reactive – at the speed of the environment
- Heterogeneous – hardware/software, mixed architectures
- Networked – shared adaptive

Source: Edward Lee®

Not new…

- Automobiles used microprocessor-based engine controllers starting in 1970's.
  - *Control fuel/air mixture, engine timing, etc.*
  - *Multiple modes of operation: warm-up, cruise, hill climbing, etc.*
  - *Provides lower emissions, better fuel efficiency.*



[dcc]

Key Recent Trends

Source: Mani Srivastava®

- Increasing computation demands
  - *e.g., multimedia processing in set-top boxes, HDTV*
- Increasingly networked
  - *to eliminate host, and remotely monitor/debug*
  - *embedded Web servers*
    - e.g., video-cameras, printers
  - *embedded Java virtual machines*
    - e.g., smart cards, printers
  - *cameras, disks etc. that sit directly on networks*
- Increasing need for flexibility
  - *time-to-market under ever changing standards!*

U.PORTO

[dcc]

Application examples

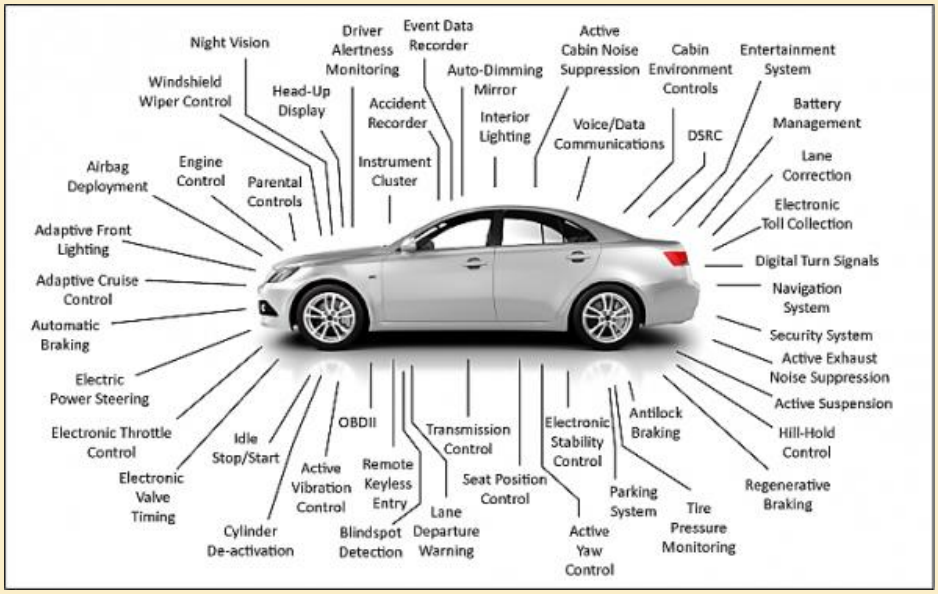
- Simple control: front panel of microwave oven, etc.
- Canon EOS 3 (1998, 35 mm film) had three microprocessors.
  - *32-bit RISC CPU runs autofocus and eye control systems.*
- Digital TV: programmable CPUs + hardwired logic for video/audio decode, menus, etc.

U.PORTO

[dcc] U.PORTO

Electronics and the Car

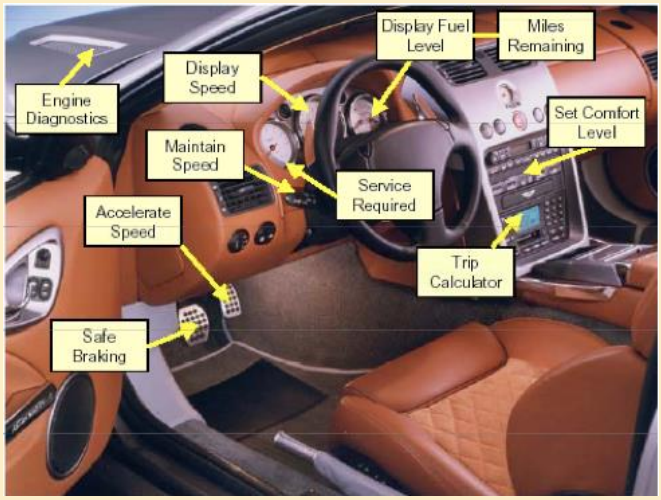
Image from: eeNews automotive.



[dcc] U.PORTO

Typical Car Controls

- Configure
- Sense
- Actuate
- Regulate
- Display
- Trend
- Diagnose
- Predict
- Archive



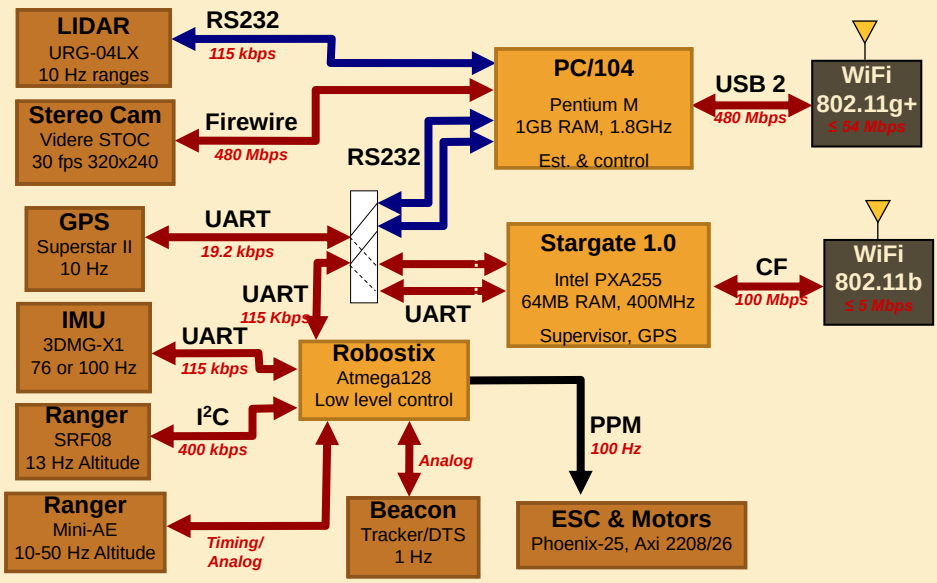
Source: Alberto Sangiovanni-Vincentelli®

Motivating Example of a Cyber-Physical System

STARMAC quadrotor aircraft (Tomlin, et al.)

15

STARMAC Design Block Diagram



16

[dcc]

Microprocessor varieties

- Microcontroller: includes I/O devices, on-board memory.
- Digital signal processor (DSP): microprocessor optimized for digital signal processing.
- Typical embedded word sizes: 8-bit, 16-bit, 32-bit and now 64 bit.

U.PORTO

[dcc]

Characteristics of embedded systems

- Sophisticated functionality.
- Real-time operation.
- Low manufacturing cost.
- Low power.
- Designed to tight deadlines by small teams.

U.PORTO

[dcc]

Functional complexity

- Often must run sophisticated algorithms or multiple algorithms.
  - *Cell phone, laser printer.*
- Often provide sophisticated user interfaces.

U.PORTO

19

[dcc]

Real-time operation

- Must finish operations by deadlines.
  - *Hard real time: missing deadline causes failure.*
  - *Soft real time: missing deadline results in degraded performance.*
- Many systems are multi-rate: must handle operations at widely varying rates.

U.PORTO

20

[dcc]

What does "performance" mean?

- In general-purpose computing, performance often means average-case
  - *may not be well-defined.*
- In real-time systems, performance means meeting deadlines.
  - *Missing the deadline by even a little is bad.*
  - *Finishing ahead of the deadline may not help.*

U.PORTO

[dcc]

Non-functional requirements

- Many embedded systems are mass-market items that must have low manufacturing costs.
  - *Limited memory, microprocessor power, etc.*
- Power consumption is critical in battery-powered devices.
  - *Excessive power consumption increases system cost even in wall-powered devices.*

U.PORTO

[dcc]

Why use microprocessors?

- Alternatives: field-programmable gate arrays (FPGAs), custom logic, etc.
- Microprocessors are often very efficient: can use same logic to perform many different functions.
- Microprocessors simplify the design of families of products.

U.PORTO

[dcc]

Integrated Circuits

ASIC BitCoin Miner from Wikipedia Targaryen

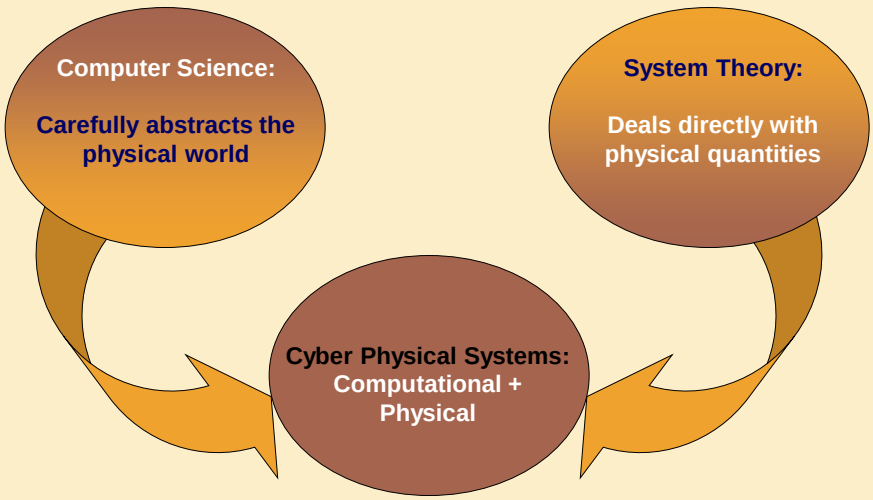
- ASIC – Application Specific Integrated Circuit
  - *IC is specific for an application*
  - *Ex.: ROM, RAM*
- FPGAs – Field-Programmable Gate Arrays
  - *Can be configured after being built*
- ASIP – Application Specific Instruction-set Processor
  - *Instruction set is specific for an application*
  - *Ex.: graphics processors, network processors*
- CPU – Central Processing Unit
  - *General purpose*

U.PORTO

The physics of software

- Computing is a physical act.
  - *Software doesn't do anything without hardware.*
- Executing software consumes energy, requires time.
- To understand the dynamics of software (time, energy), we need to characterize the platform on which the software runs.

This Subject is Multidisciplinary



[dcc]

REQUIREMENTS FORM

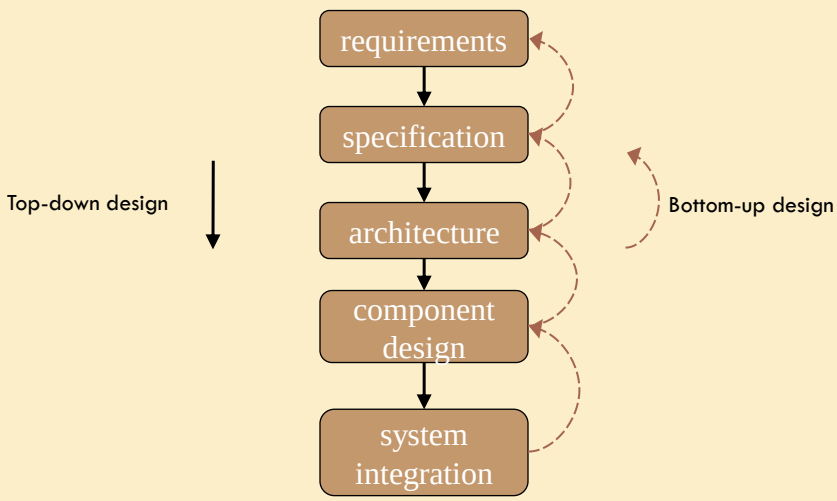
U.PORTO

FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

27

[dcc]

Levels of abstraction



U.PORTO

28

28

[dcc]

Top-down vs. bottom-up

- Top-down design:
  - *start from most abstract description;*
  - *work to most detailed.*
- Bottom-up design:
  - *work from small components to big system.*
- Real design uses both techniques.

U.PORTO

[dcc]

Requirements

- Plain language description of what the user wants and expects to get.
- May be developed in several ways:
  - *talking directly to customers;*
  - *talking to marketing representatives;*
  - *providing prototypes to users for comment.*

U.PORTO

Functional vs. non-functional requirements

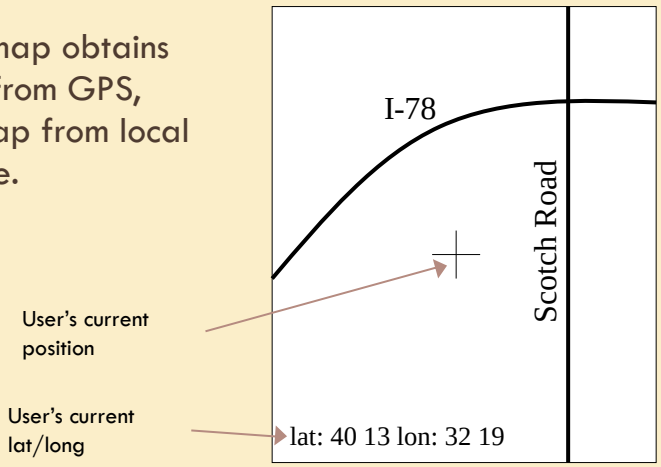
- Functional requirements:
  - *output as a function of input.*
- Non-functional requirements:
  - *time required to compute output;*
  - *size, weight, etc.;*
  - *power consumption;*
  - *reliability;*
  - *etc.*

Our requirements form

- name
- purpose
- inputs
- outputs
- functions
- performance
- manufacturing cost
- power
- physical size/weight

Example: GPS moving map requirements

- Moving map obtains position from GPS, paints map from local database.



GPS moving map needs

- Functionality: For automotive use. Show major roads and landmarks.
- User interface: At least 400 x 600 pixel screen. Three buttons max. Pop-up menu.
- Performance: Map should scroll smoothly. No more than 1 sec power-up. Lock onto GPS within 15 seconds.
- Cost: €120 street price = approx. € 30 cost of goods sold.
- Power consumption: Should run for 8 hours on four AA batteries.
- Physical size/weight: Should fit in hand.

[dcc]

GPS moving map requirements form

Name	GPS moving map
Purpose	Consumer-grade moving map for driving
Inputs	power button, two control buttons
Outputs	back-lit LCD 400 X 600
Functions	5-receiver GPS; three resolutions; displays current lat/lon
Performance	updates screen within 0.25 sec of movement
Manufacturing cost	€ 100 cost-of-goods-sold
Power	100 mW
Physical size/weight	no more than 2: X 6:, 12 oz.

U.PORTO

35

[dcc]

SPECIFICATION

U.PORTO

FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

36

[dcc]

"A design without specifications cannot be right or wrong, it can only be surprising!"

(paraphrased from Young et al. (1985):

U.PORTO

[dcc]

Specification

- A more precise description of the system:
  - *should not imply a particular architecture;*
  - *provides input to the architecture design process.*
- May include functional and non-functional elements.
- May be executable or may be in mathematical form for proofs.

U.PORTO

[dcc] U.PORTO

GPS specification

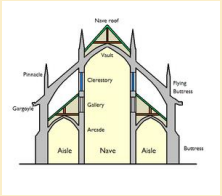
Should include:

- What is received from GPS;
- map data;
- user interface;
- operations required to satisfy user requests;
- background operations needed to keep the system running.

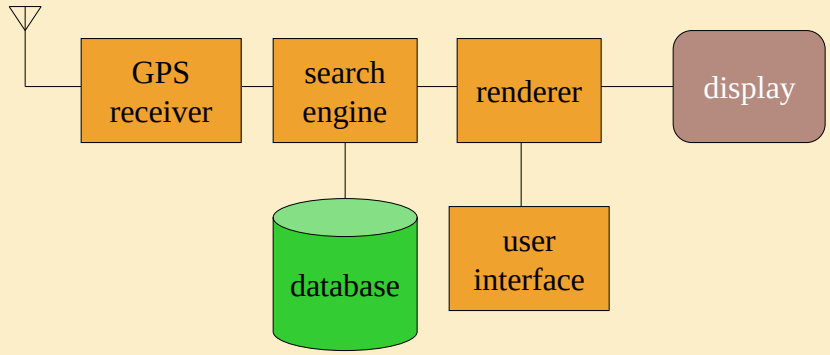
[dcc] U.PORTO

Architecture design

- What major components to satisfy the specification?
- Hardware components:
  - *CPUs, peripherals, etc.*
- Software components:
  - *major programs and their operations.*
- Must consider functional and non-functional specifications.

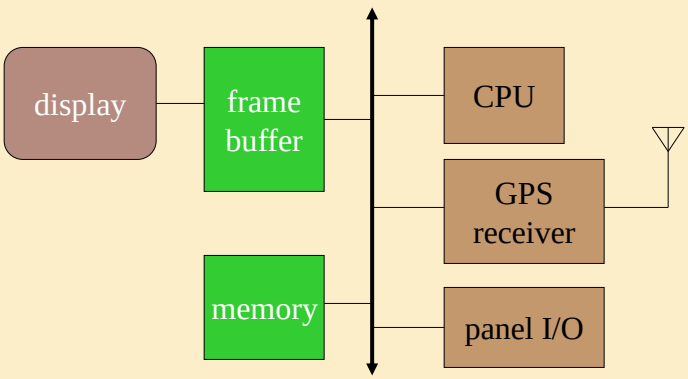


GPS moving map block diagram



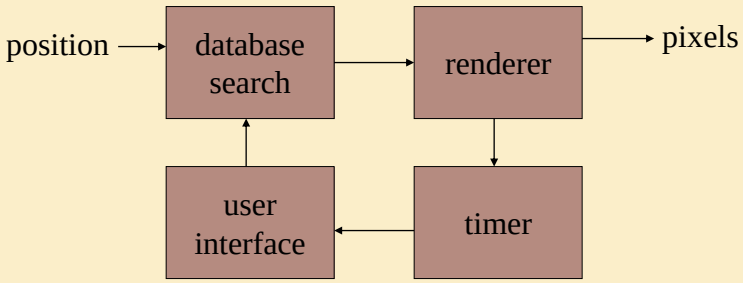
41

GPS moving map hardware architecture



42

GPS moving map software architecture



Designing hardware and software components

- Must spend time architecting the system before you start coding.
- Some components are ready-made (ex.: GPS receiver), some can be modified from existing designs (ex.: panel IO), others must be designed from scratch (ex.: display).

[dcc]

System integration

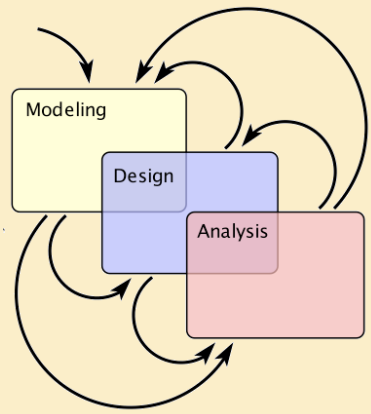
- Put together the components.
  - *Many bugs appear only at this stage.*
- Have a plan for integrating components to uncover bugs quickly, test as much functionality as early as possible.

U.PORTO

[dcc]

Modeling, Design, Analysis

- Modeling is the process of gaining a deeper understanding of a system through imitation. Models specify what a system does.
- Design is the structured creation of artifacts. It specifies how a system does what it does.
- Analysis is the process of gaining a deeper understanding of a system through dissection. It specifies why a system does what it does (or fails to do what a model says it should do).



U.PORTO

[dcc]

Think Critically

- Any course that purports to teach you how to design embedded systems is misleading.

The technology will change!

- Goal is understand how things are done today, and why that is not good enough.
  - *So not to be surprised by the changes that are coming.*

U.PORTO

47

[dcc]

Summary

- Embedded computers are all around us.
  - *Many systems have complex embedded hardware and software.*
- Embedded systems pose many design challenges: design time, deadlines, power, etc.
- Design methodologies help us manage the design process.
- Model, Design, Analysis

U.PORTO

48