Module Introduction

Rudi Villing (Room 3.03, rudi.villing@mu.ie)

Online at http://moodle.maynoothuniversity.ie

EE302 – Real time and embedded systems

1

Module descriptor summary

- Learning outcomes
  - Design simple embedded systems
  - Implement and test basic embedded applications
  - Implement and test real-time embedded applications
- Duration of elements and marks breakdown
  - 24 x 1hr lectures
  - 6 x 3hr labs/tutorial labs – **18%**
  - 1 x 15hr Assignment – **12%**
  - 1 x 2hr end of semester exam – **70%**

2

To succeed…

- **All** material covered during contact hours and in online discussion is part of the course, in particular…
  - All examples given in lectures or labs are relevant
  - All extra verbal explanation given in class is relevant
  - The bullet point notes on their own are not sufficient
- If you don't understand something, ask questions during class (or in online forum)
- Beware!
  - Penalties for late submission of labs/assignments
  - There is generally no facility for catch up on missed labs (without a certified absence)
  - As normal, continual assessments can not be repeated

3

1.10 Introduction to Embedded Systems

EE302 – Real time and embedded systems

4

Overview

- Aims 介绍嵌入式系统的概念及其主要特性
  - Introduce the concept of an embedded system and its main properties
- Learning outcomes – you should be able to…
  - Explain the concept of an embedded system
  - Recognise and give examples of real world embedded systems
  - Differentiate embedded systems from desktop systems
  - Give examples of the practical consequences the difference between embedded systems and desktop systems

解释嵌入式系统的概念识别
并给出真实世界嵌入式系统的例子
区分嵌入式系统和桌面系统
举例说明嵌入式系统和桌面系统之间的差异所带来的实际后果

What is an embedded system?

Embedded system:

(a) a computing system designed to perform a dedicated task

(b) often embedded within systems or devices which are not themselves computers

- Example 1
  - a network router is a computing system dedicated to the task of routing packets in a network
  - It is a standalone embedded system
- Example 2
  - a microwave oven is an electronic appliance which happens to contain a computing system to control its user interface and cooking time
  - The embedded system is the user interface/cooking controller
  - It is embedded within the microwave oven appliance

Main elements of an embedded system

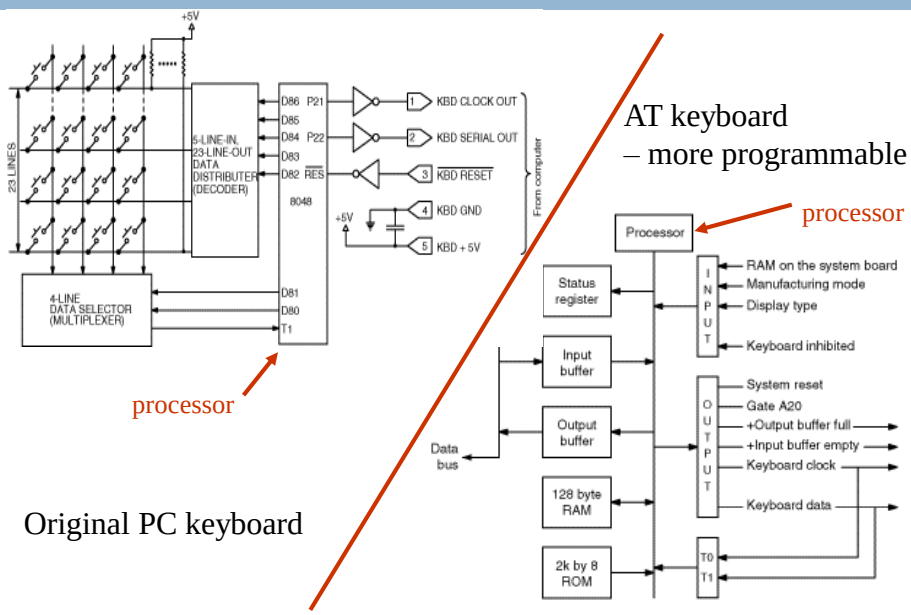
- Input/output
  - to communicate with sensors, actuators, displays etc
- Processor
  - the computing engine
- [Optional] Hardware acceleration
  - E.g. ASICs, FPGAs, AI accelerators, etc
- Firmware
  - embedded software

Why we say "embedded system"

- Consider a BMW 745i
  - it's not a computer – it's a car
- It contains many embedded systems
  - The embedded systems are application specific computers (e.g. engine management etc)
  - 53 x 8-bit microprocessors, 11 x 32-bit microprocessors, 7 x 16-bit microprocessors

Example: PC Keyboard

9

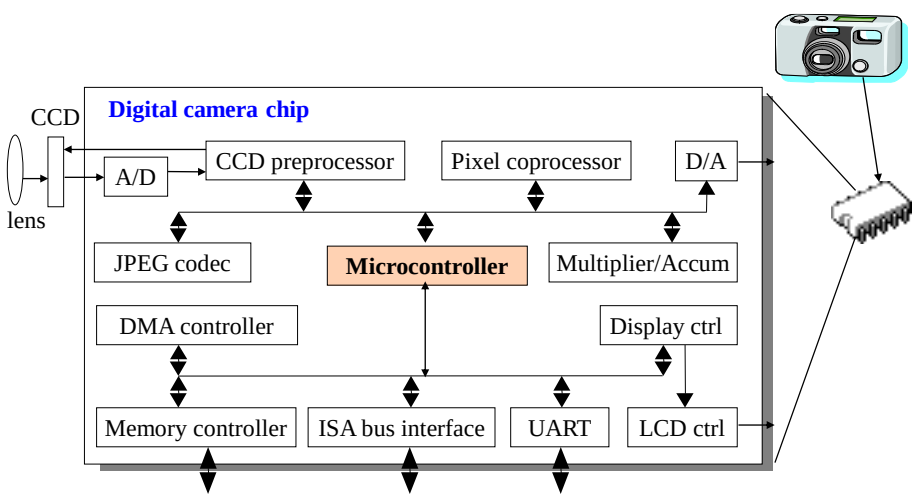


29 September 2020

9

Example: Digital Camera

11



29 September 2020

11

Example question

Q1. Explain the concept of an embedded system.

- *What is the significance of the word "embedded"?*
- *Give 3 examples of embedded systems.*

Q2. Which of the following are embedded systems and why?

- *A robot*
- *A digital radio*
- *A hoverboard*
- *Wireless headphones*
- *A smartphone*
- *A TV remote*
- *An ATM*
- *A vending machine*

What makes embedded systems different (from other computer systems)?

1. Task specific
2. Real time constraints
3. Fault handling and implications
4. Power constraints
5. Operating environment
6. Cost sensitivity
7. Wide range of processor types
8. Resource constraints
9. Program code in ROM
10. Specialized development/debugging tools

Q. How do these differences affect your system design in practice?

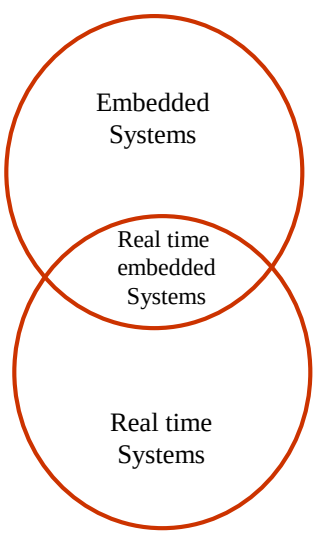
Differences 1: Tasks specific

- Task specific, not general purpose
  - A digital camera is always a digital camera
  - A microwave is always a microwave
  - A bit fuzzier for PDAs, game consoles, and modern smart phones

Q. How does this affect the system design in practice?

Differences 2: Real time (depends on system)

- *Note: Some, but not all embedded systems are real time*
- Real time constraints
  - Has deadlines for system response to an input
  - Outputs must be both correct and on time
  - Predictable vs. fast
- Real Time Operating System
  - Do you need one?
  - Common services
    - Communications stack (e.g. TCP/IP)
  - VxWorks, QNX, LynxOS, RTLinux, many more…



Q. How does this affect the system design in practice?

Differences 3: fault handling

- What are the implications of software failure?
  - Safety critical systems
    - mobile phone failure is just annoying
    - Failure of a car's ABS or an aircraft's flight control software may be fatal!
  - Affects how system is specified, designed, implemented, debugged and validated
  - Is the system fault tolerant, fail safe, etc?
  - Affects system design and perhaps even programming language: E.g. US Dept of Defense mandates use of ADA language for many projects
    - ADA has language features which make certain kinds of software mistakes difficult or impossible to make

Q. How does this affect the system design in practice?

Differences 4 and 5: power and operating environment

- Power constraints
  - Limits on total power consumption
  - Battery lifetime
  - cooling requirements
- Operating environment
  - heat/cold
  - Humidity
  - Pressure/vacuum
  - Shock and vibration
  - Radiation, electromagnetic interference, etc.

Q. How does this affect the system design in practice?

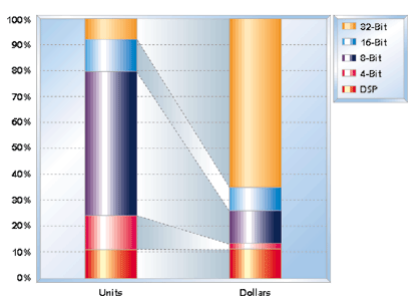
Differences 6: cost

- Cost sensitivity
  - Unit cost: the monetary cost of manufacturing each copy of the system, excluding NRE cost
  - NRE cost (Non-Recurring Engineering cost): The one-time monetary cost of designing the system
  - Time-to-prototype: the time needed to build a working version of the system
  - Time-to-market: the time required to develop a system to the point that it can be released and sold to customers
  - Flexibility: the ability to change the functionality of the system without incurring heavy NRE cost

Q. How does this affect the system design in practice?

Differences 7: range of processor types

- Wide range of processor types
  - 4, 8, 16, 32 bit
  - About 98% of all 32-bit processors are used in embedded systems, not PCs.



Semiconductor units and revenue by type
Source: http://www.embedded.com/story/OEG20021217S0039

Q. How does this affect the system design in practice?

Differences 8-10: constraints, ROM, tools

- Resource constrained
  - E.g. PIC16F876A has just 368 Bytes of data memory
  - Often few input buttons and simple displays
  - Greatly affects software design
- Program code in ROM
  - Affects data initialization, debugging
- Specialized development/debugging tools
  - Cross compiler/assembler
  - Emulator
  - In Circuit Debugger

Q. How does this affect the system design in practice?

Example question

Q1. What do you think are the 3 most significant/important ways in which an embedded system differs from a desktop or server computer system?

- *Illustrate each difference with a suitable embedded system example.*

Q2. State at least one practical consequence of each of the following differences on your system design?

- *Fault handling*
- *Power constraints*
- *Range of processor types*
- *Program code in ROM*