Lecture 10: EITF20 Computer Architecture

Anders Ardö

EIT – Electrical and Information Technology, Lund University

December 9, 2014

Outline

RAID types

	RAID level	Failures tolerated	Overhead 8 data disks	comment
0	striped	0	0	JBOD, common
1	mirrored	1-8	8	high overhead
2	ECC	1	4	not used
3	bit parity	1	1	synchronized drives
4	block parity	1	1	
5	block parity distributed	1	1	common
6	row-diagonal dual parity	2	2	high availability
01	mirrored stripes	1-8	8	
10	striped mirrors	1-8	8	

Summary I/O

I/O:

- I/O performance is important!
- The task of the I/O system designer:
  - meet performance needs
  - cost-effective
  - reliability, availability
- I/O system parts
  - CPU interface
  - Interconnect technology
  - Device performance

Disks:

- Disks have moving parts leading to long service times
- RAID disk arrays provide high bandwidth, high capacity disk storage at a reasonable cost
- SSD is faster and more expensive

Lecture 12 agenda

Appendix D in "Computer Architecture"

Outline

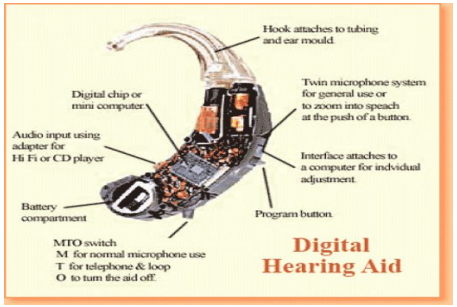
Embedded processors

A device that includes a (programmable) computer
But is not itself a general-purpose computer

- fastest growing segment
  - washing machines, cars, cell phones, TVs, ...
- wide range: low-end 8 bit $\rightarrow$ full size 32 bit
- price key factor
- performance, power, memory
- real time applications
- types
  - ASIC
  - SoC
  - DSP
  - General purpose CPU, microcontroller
  - ...

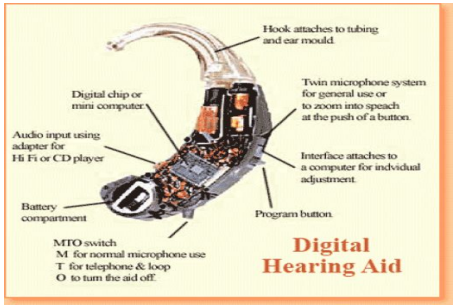
Modern (Embedded) computers

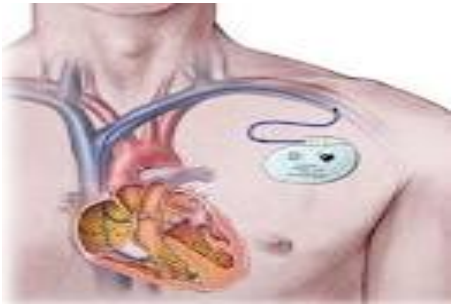
Modern (Embedded) computers



Microcontrollers

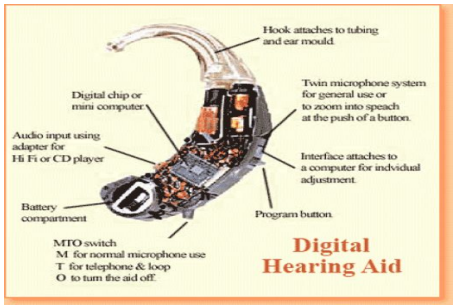
Modern (Embedded) computers

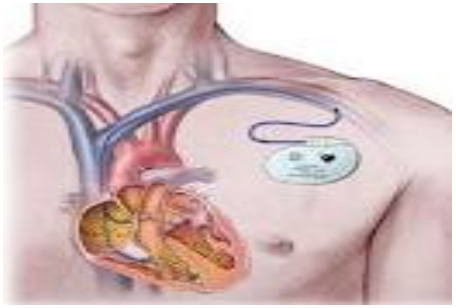




Microcontrollers

Modern (Embedded) computers

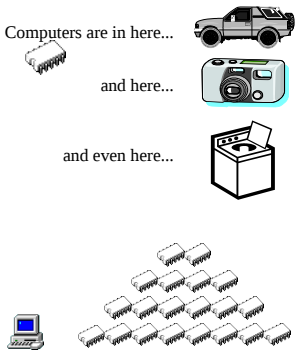




How computers took over our cars

Embedded systems overview

- Embedded computing systems
  - Computing systems embedded within electronic devices
  - Hard to define. Nearly any computing system other than a desktop computer
  - Billions of units produced yearly, versus millions of desktop units
  - Perhaps 50 per household and per automobile



A “short list” of embedded systems

Anti-lock brakes	Modems
Auto-focus cameras	MPEG decoders
Automatic teller machines	Network cards
Automatic toll systems	Network switches/routers
Automatic transmission	On-board navigation
Avionic systems	Pagers
Battery chargers	Photocopiers
Camcorders	Point-of-sale systems
Cell phones	Portable video games
Cell-phone base stations	Printers
Cordless phones	Satellite phones
Cruise control	Scanners
Curbside check-in systems	Smart ovens/dishwashers
Digital cameras	Speech recognizers
Disk drives	Stereo systems
Electronic card readers	Teleconferencing systems
Electronic instruments	Televisions
Electronic toys/games	Temperature controllers
Factory control	Theft tracking systems
Fax machines	TV set-top boxes
Fingerprint identifiers	VCR’s, DVD players
Home security systems	Video game consoles
Life-support systems	Video phones
Medical testing systems	Washers and dryers

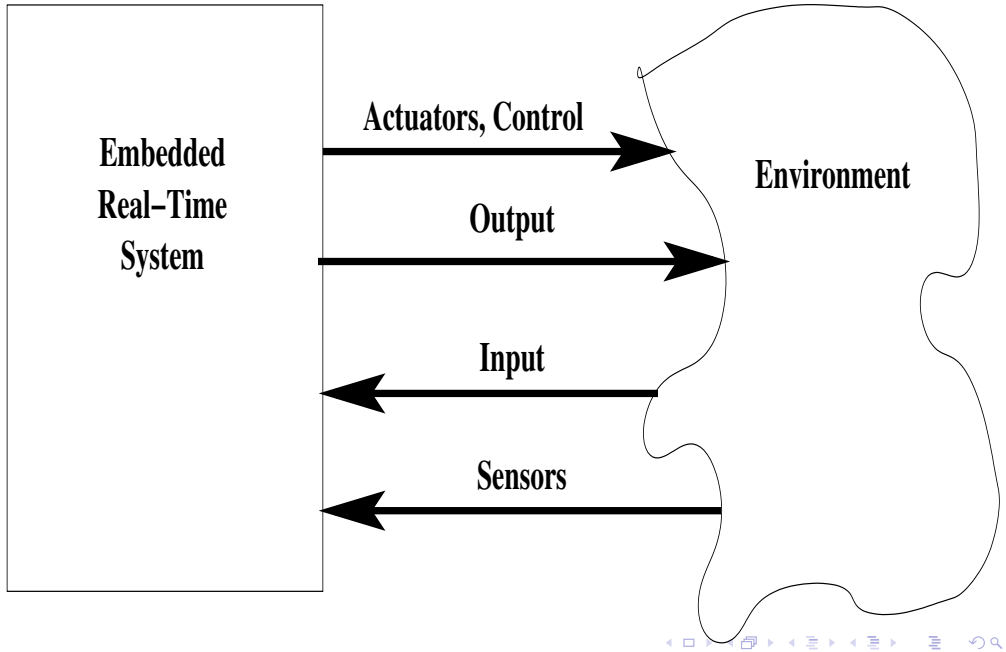
And the list goes on and on

Some common characteristics of embedded systems

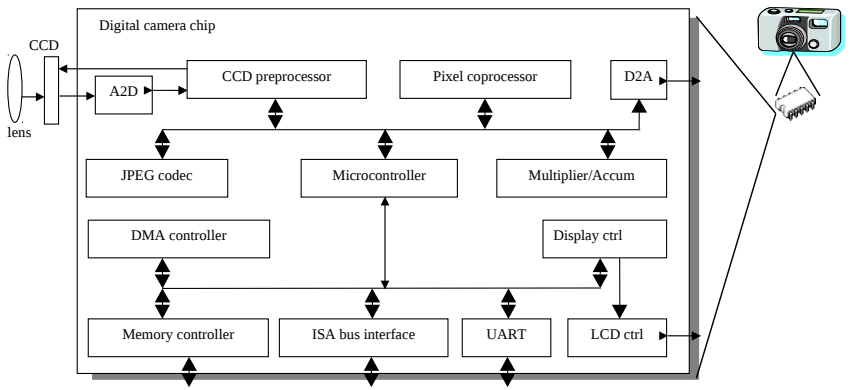
- Single-functioned
  - Executes a single program, repeatedly
- Tightly-constrained
  - Low cost, low power, small, fast, etc.
- Reactive and real-time
  - Continually reacts to changes in the system’s environment
  - Must compute certain results in real-time without delay

TA-150 Computer Controlled Stereo Receiver

Embedded system



An embedded system example -- a digital camera

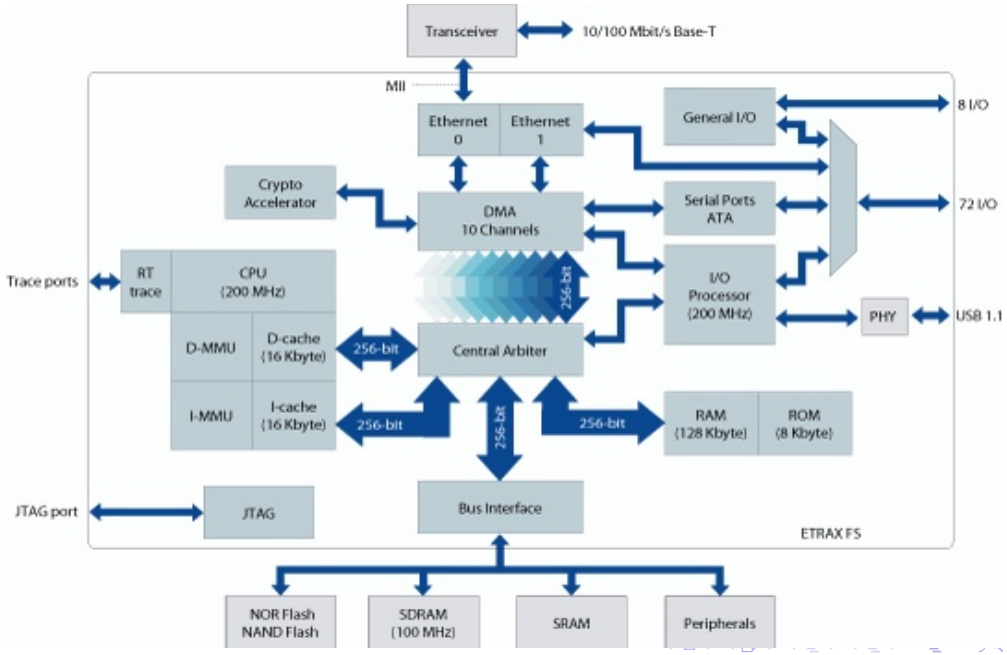


- Single-functioned -- always a digital camera
- Tightly-constrained -- Low cost, low power, small, fast
- Reactive and real-time -- only to a small extent

Embedded Systems Design: A Unified Hardware/Software Introduction, (c) 2000 Vahid/Givargis

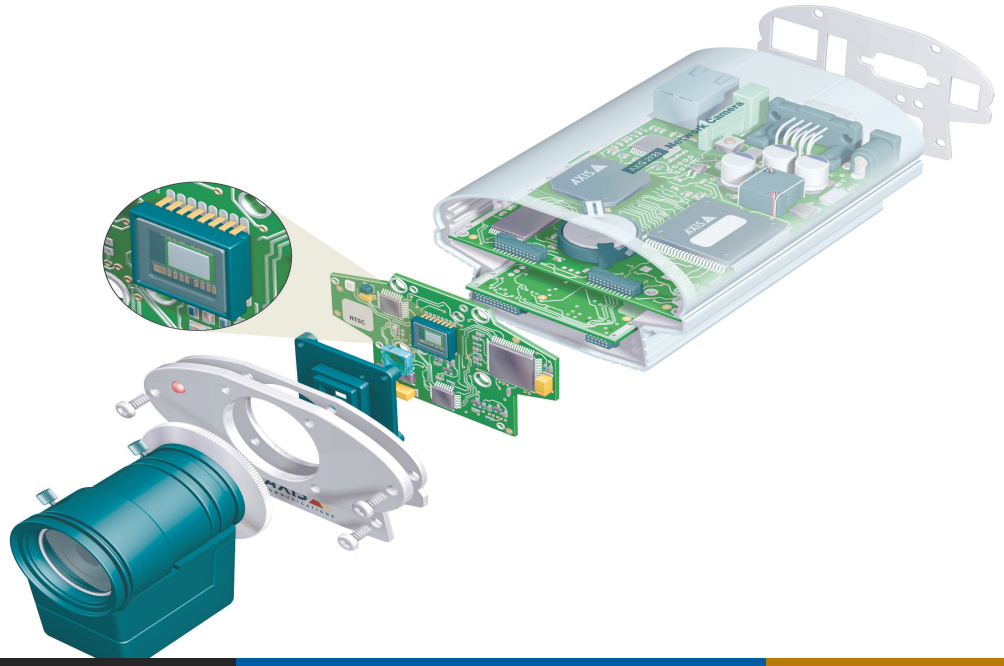
Outline

Axis Etrax FS



Axis Network Camera

Axis Network Camera



Outline

ARM

- ARM is a family of instruction set architectures (ARMv1 - ARMv8)
- 32/64(ARMv8) bit architecture
- based on a reduced instruction set computing (RISC)
- most widely used architecture in mobile devices
- most popular 32-bit one in embedded systems
- licensed: Companies that produce ARM products include Apple, Nvidia, Qualcomm, Samsung Electronics, and Texas Instruments.

ARMv8-A

- RISC ISA (+ extensions)
- load/store
- both big and little endian
- 64 bit address-space
- 31 x 64 bit registers
- 32 x 128 bit floating point registers
- 32-bit wide fixed length instructions
- (extensions support 16-bit wide instructions)

ARMv8 implementations

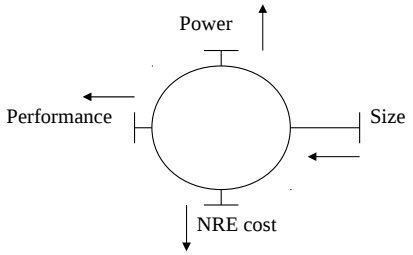
	Cortex A53	Cortex A57
process	20 nm	20 nm
pipeline	in-order	out-of-order
depth	8	15
		128 instr. in flight
issue	2	3 (8 issue slots)
FU	dual int + FP	multiple
L1 TLB	10	48 I / 32 D
L2 TLB	512	1024
L1 Cache	8-64 kB I / 8-64 kB D	48 kB I / 32 kB D
L2 Cache	128 kB - 2 MB	512 kB - 2 MB
L2 org		16 way associative
features	prefetch	prefetch
	non-blocking	non-blocking
		banked
		way prediction

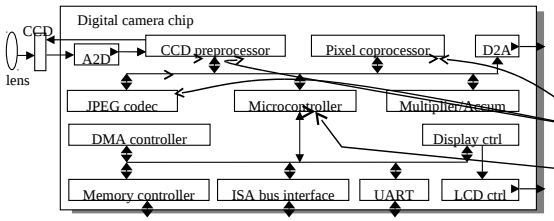
Outline

Design challenge – optimizing design metrics

- Common metrics
  - 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
  - Size: the physical space required by the system
  - Performance: the execution time or throughput of the system
  - Power: the amount of power consumed by the system
  - Flexibility: the ability to change the functionality of the system without incurring heavy NRE cost

Design metric competition -- improving one may worsen others





- Expertise with both **software and hardware** is needed to optimize design metrics
  - Not just a hardware or software expert, as is common
  - A designer must be comfortable with various technologies in order to choose the best for a given application and constraints

Hardware vs software

- hardware
  - performance
  - power
  - cost
- software
  - flexibility
  - reconfigurability
  - cost

Outline

Real time

- React to external environment
- Permanent interaction
- Endless execution
- External timing requirements

Real time performance

Special application areas

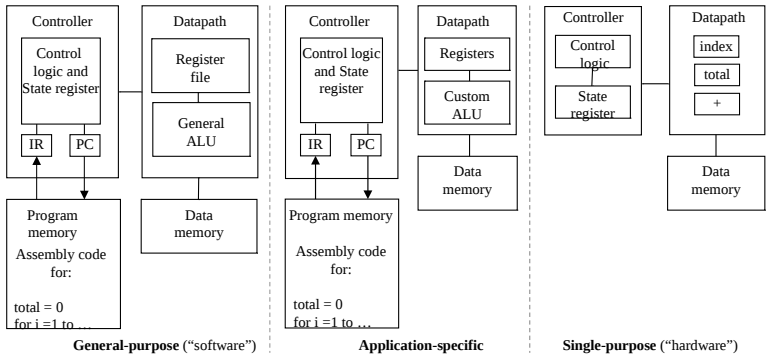
- video
- process control
- medical applications
- airplane control - JAS

- Hard vs soft real time requirements
- Analyzes WCET - Worst Case Execution Time

Outline

Processor technology

- The architecture of the computation engine used to implement a system's desired functionality
- Processor does not have to be programmable
  - "Processor" *not* equal to general-purpose processor



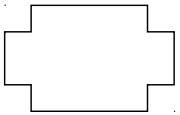
Processor technology

- Processors vary in their customization for the problem at hand

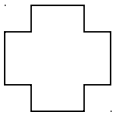
Desired functionality

```
total = 0
for i = 1 to N  loop
  total += M[i]
end loop
```

General-purpose processor



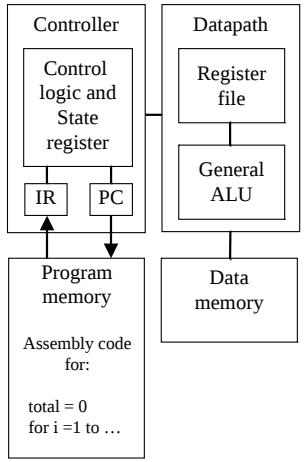
Application-specific processor



Single-purpose processor

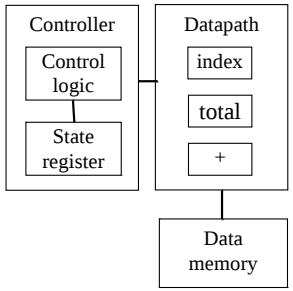
General-purpose processors

- Programmable device used in a variety of applications
  - Also known as "microprocessor"
- Features
  - Program memory
  - General datapath with large register file and general ALU
- User benefits
  - Low time-to-market and NRE costs
  - High flexibility
- "Pentium" the most well-known, but there are hundreds of others



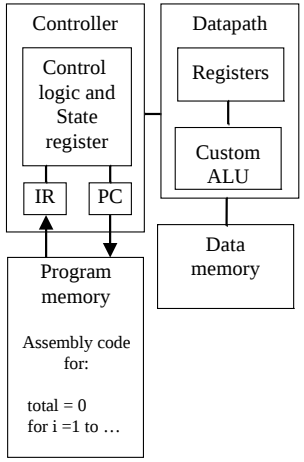
Single-purpose processors

- Digital circuit designed to execute exactly one program
  - a.k.a. coprocessor, accelerator or peripheral
- Features
  - Contains only the components needed to execute a single program
  - No program memory
- Benefits
  - Fast
  - Low power
  - Small size



Application-specific processors

- Programmable processor optimized for a particular class of applications having common characteristics
  - Compromise between general-purpose and single-purpose processors
- Features
  - Program memory
  - Optimized datapath
  - Special functional units
- Benefits
  - Some flexibility, good performance, size and power



Flexibility vs Efficiency hardware platforms

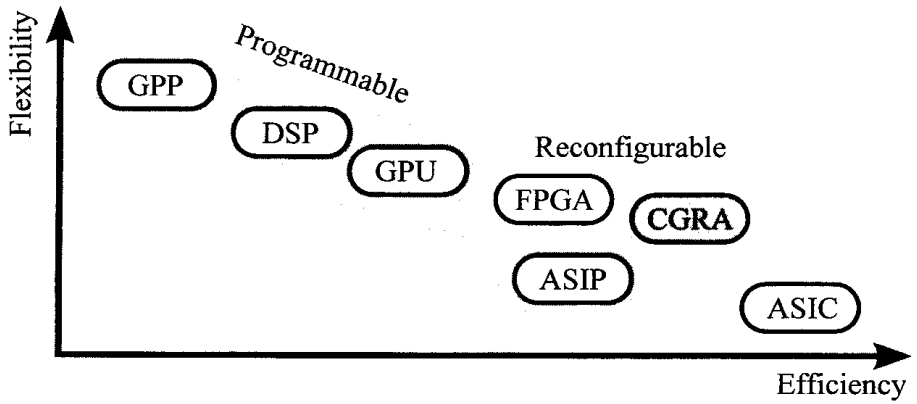
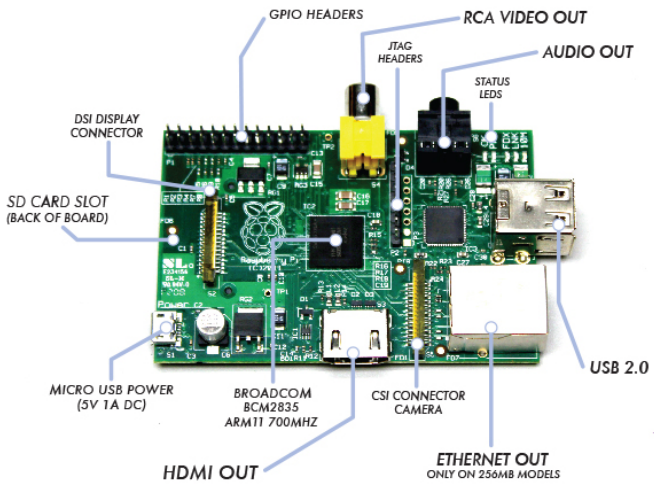


Figure 2.1: Comparison of flexibility and efficiency for various forms of hardware platforms.

(From Chenxin Zhang: "Dynamically Reconfigurable Architectures for Real-time Baseband Processing", PhD thesis, EIT, 2014)

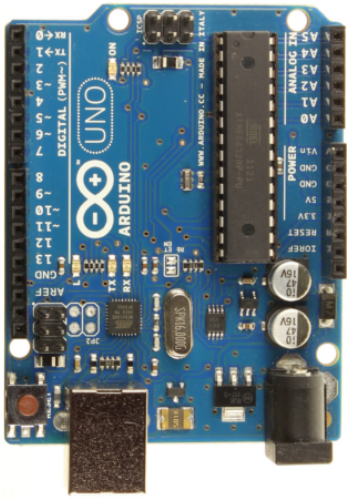
Outline

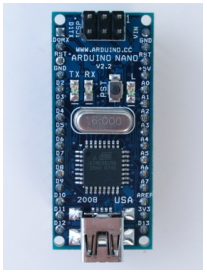
A simple Linux computer

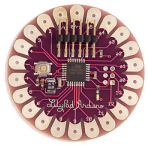


Raspberry Pi: 250 + SD-card + Box + (Power) $\approx 500 SEK$

Arduino

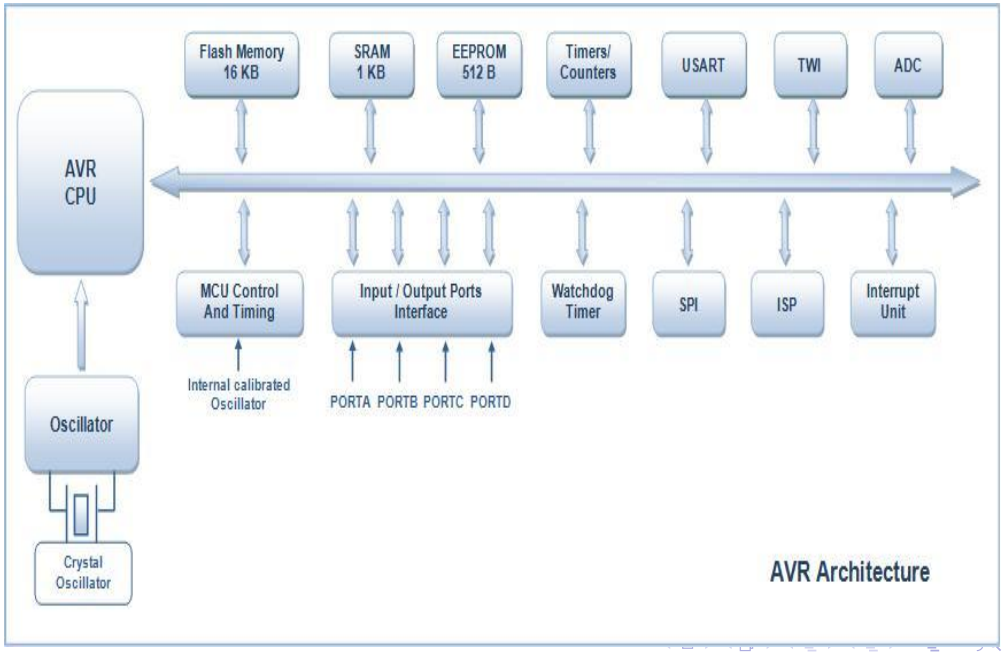




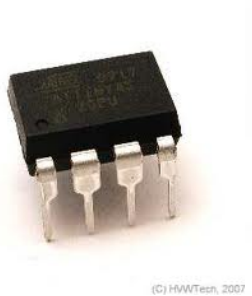


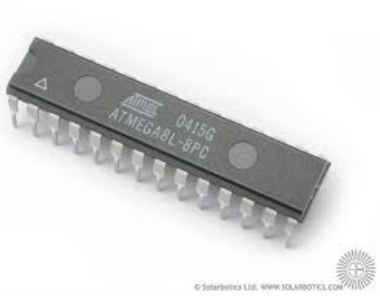
http://arduino.cc/en/

AVR overview



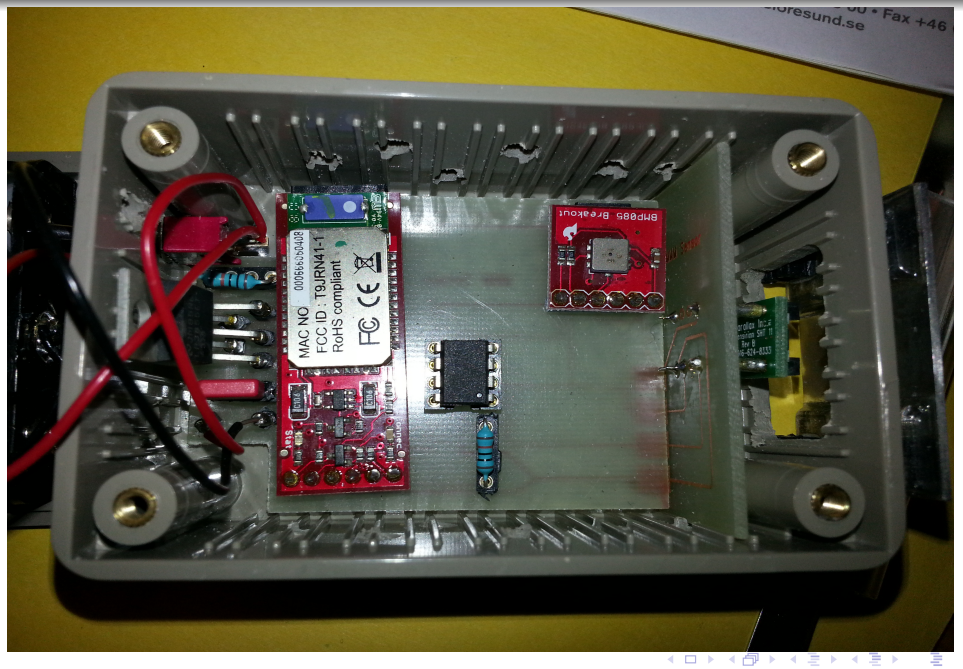
AVR micro-controller





http://www.atmel.com/

Application - temp, humidity, pressure



Application - temp, humidity, pressure

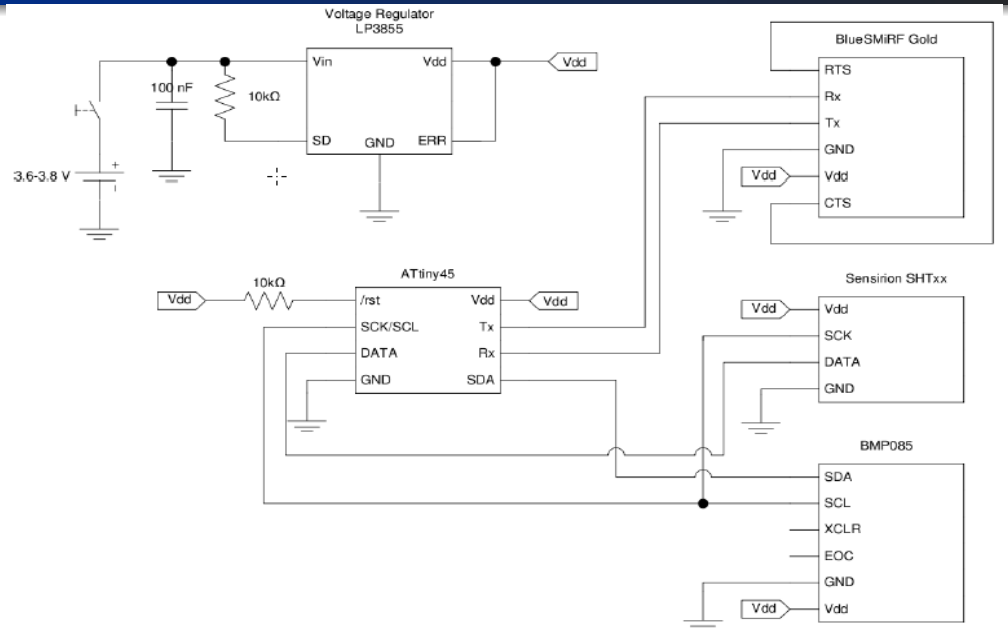
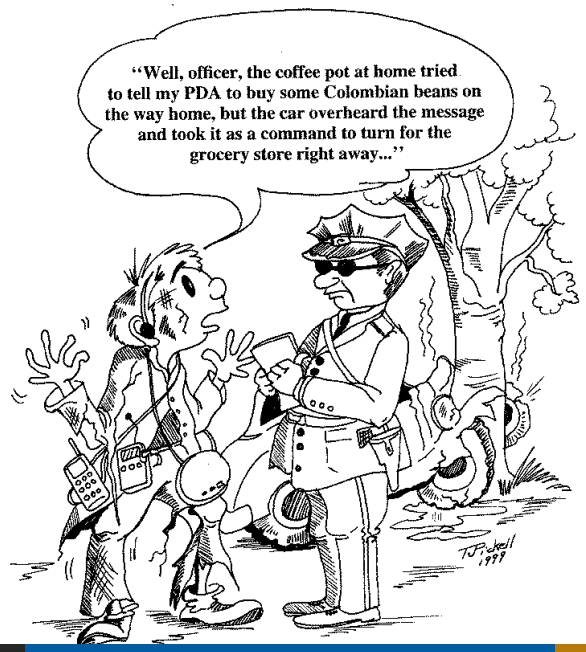


Figure 3.1: Schematic overview of the sensor unit.

Outline

Linux-Running 'Smart Fridge'

They Are Everywhere!



Summary

- Important, found everywhere, high volume
- General purpose, application specific, single purpose
- Design of hardware and software together
- Cover several areas
  - microelectronics
  - real time
  - software + hardware
  - SoC