SFU

SCHOOL OF INTERACTIVE ARTS + TECHNOLOGY

IAT 267 Introduction to Technological Systems

Instructor: Helmine Serban

Teaching Assistants:
Ying Deng
Daniel Feuereissen

Agenda

- Introductions
- Class overview
  - Course Outline (-> posted on WebCT)
  - Resources
  - Expectations and Grading
- What are technological systems?
- Organization of the course
- Unit 1

Instructor

- Helmine Serban – Senior Lecturer at SIAT
  - Contact: helmine@sfu.ca
  - Office: SFU Surrey Campus, Podium 2, Room 2727
  - Office Hours: TBA

Teaching Assistants

- Ying Deng
  - Contact: yingd@sfu.ca
  - Office: grad space
  - Office hours: TBA
- Daniel Feuereissen
  - Contact: dfeuerei@sfu.ca
  - Office: grad space
  - Office hours: TBA

Classes

- **Lecture:**
  - Wednesday – attendance strongly recommended
- **Workshops:**
  - Friday
    - Each student should attend the workshop with the section in which they have registered.
    - Sections limited at 24 students

Course Evaluation Structure

- Assignments (3): 15% (individual)
  - Assignment 1: computer systems
  - Assignment 2: sensors
  - Assignment 3: networking
- Workshop participation, quizzes, lab activities, in-class worksheets: 20% (individual and team based)
- Course project: 30% (individual and team based)
- Final exam: 35% (individual)

Expectations

- Lecture attendance is strongly recommended
  - You are responsible for all content delivered during lecture (course material, announcements, solutions to assignments, etc)
- Attend the workshop and **come prepared** – review the lectures
- Read through all content on WebCT
  - Weekly lectures
  - Announcements
- If you cannot attend a workshop, let us know **before** the workshop
- Grades will be posted on WebCT

Lecture / Workshop Focus

- **Lecture**
  - Focus on theoretical concepts
  - Explain techniques used in sensor-microcontroller-computer communication
  - Explain / walk-through many code samples
  - Analyze circuits

- **Workshop**:
  - Practice the techniques covered during lecture
  - Build circuits
  - Write code
  - Lab exercises cover a variety of situations of computer – sensor system communication

Course Resources

- *"Using Information Technology: A Practical Introduction to Computers and Communications"* (2007) by Brian K. Williams and Stacey Sawyer; 7th/8th/ 9th Edition; (Soft cover); 592 pages; ISBN 0072260718 – some course content / readings from this book

- Lecture slides, workshops, course information: WebCT

- Library equipment (Arduino boards, sensors, breadboards, etc)

- Software / programming languages used in the course:
  - Processing language (Java-based)
  - Wiring (for Arduino)
  - Java

Programming

- Processing – (based on Java) (IAT265)
- Arduino language (Wiring)
- Case studies of computer networking – Java

Processing Language

- *Processing* is an open source programming language and environment for people who want to program images, animation, and interactions.

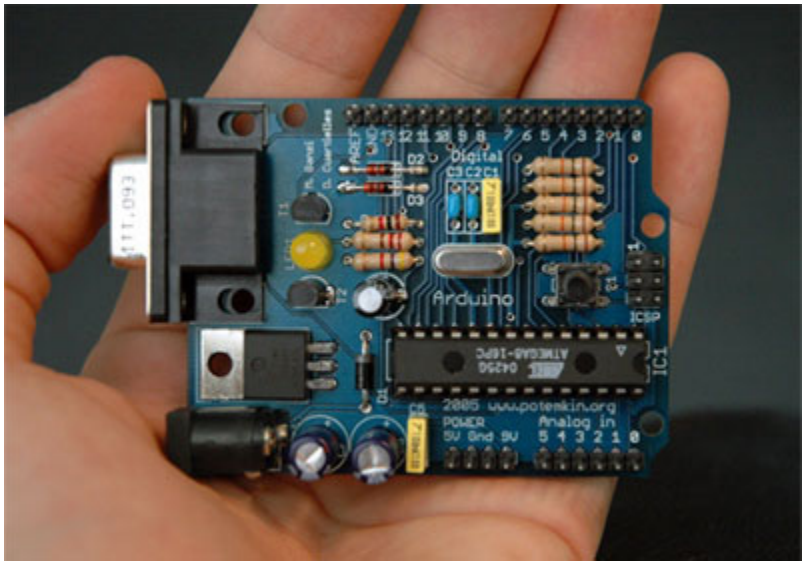
  - Download from http://processing.org/



Wiring / Arduino Programming Environment

- Download the Arduino software from http://arduino.cc/en/Main/Software
- Arduino is an open-source physical computing platform based on a simple microcontroller board, and a development environment for writing software for the board.

The microcontroller board:



Wiring / Arduino Programming Environment

- Arduino can be used to develop interactive objects, taking inputs from a variety of switches or sensors, and controlling a variety of lights, motors, and other physical outputs. Arduino projects can be stand-alone, or they can be communicate with software running on your computer (e.g. Flash, Processing, MaxMSP.)

Wiring / Arduino Programming Environment

- The Arduino programming language is an implementation of *Wiring*, a similar physical computing platform, which is based on the Processing multimedia programming environment.

- See sample projects from the past sessions of this class

Equipment Used In The Course

- During workshops: equipment will be provided

- Outside class work: students can sign-out Arduino boards from the library and other equipment (sensors, starter kits, etc).

Arduino And Small Electronic Parts

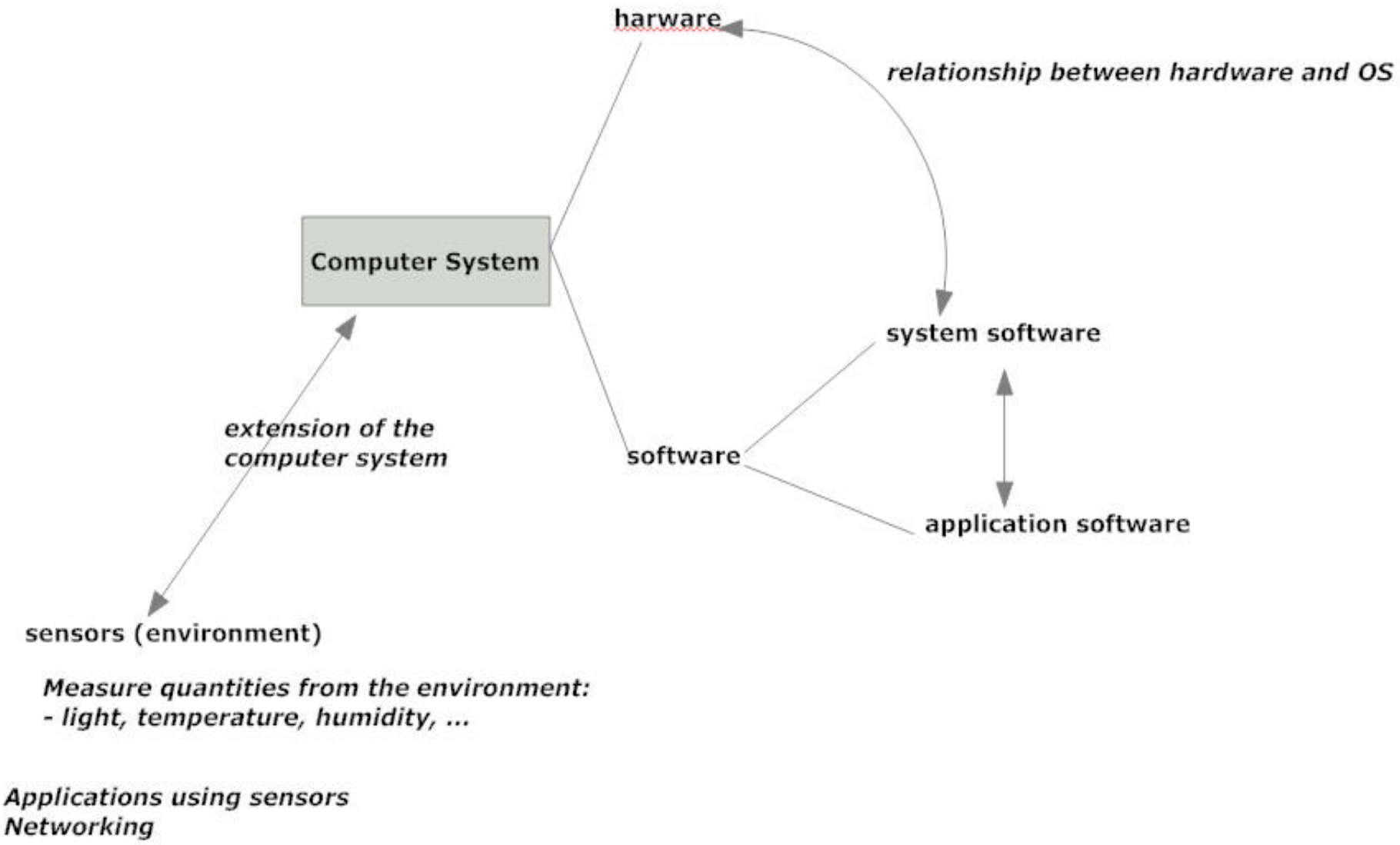
- It is **strongly recommended** that students buy their own Arduino board – used in this class and in several other upper-level SIAT classes

Arduino And Small Electronic Parts

- Where to buy the Arduino board and / or starter kit:
  - Lee's Electronics (Vancouver)
  - Canakit Electronic Kits and Modules
  - Robotshop.ca
  - Other online sources

  - Price of an Arduino board is approx. 30$; starter kit including small breadboard and some components for circuits is around 45$

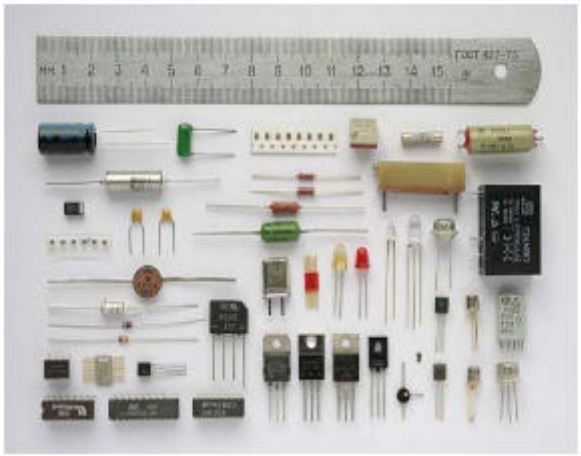
Course topics and relationships between them

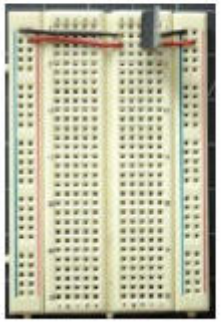


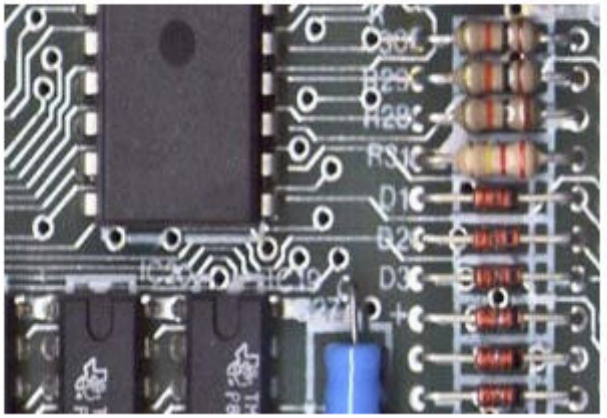
Topics In The Course

- Computer hardware
  - Relationship between computer hardware and operating system
- Software: system software and application software, the relationship between them
  - Operating system
  - Application software

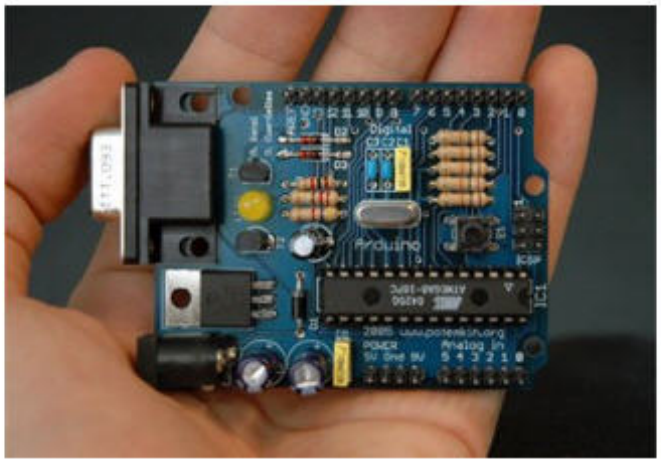
- Sensors: how sensors integrate with a computer system, reading / processing data from sensors, generating output based on sensor data
  - Simple sensor circuits (using discrete components)







– Sensor circuits using a microcontroller board (Arduino)



Arduino : a physical computing platform for a single-board microcontroller, with embedded I/O support and a standard programming language - Wire

Phidgets : a system of low-cost electronic components and sensors that are controlled by a PC

• Applications can be developed in Mac OS X, Linux, Windows CE and Windows operating systems

- Develop applications using sensors
  - Microcontroller (Arduino) – sensor – computer communication
  - Stand-alone Arduino applications
  - Processing applications, involving animation, mouse and keyboard interactions, etc
- Networking topics
  - Network protocols, applications and programming
- Networking applications
  - Java networking applications (case studies)

If You Need Help

- Discussion area in WebCT – monitored daily - fast response time

- Office Hours – by appointment (if no appointments are requested by 8pm the day before office hours, office hours are automatically cancelled)

- Email us if your question is not of general interest to the class – response time is 48 hours during week days.

Lecture 1
Technological Systems

Topics

- Technological systems: why study them?
- Directions of study
- Computer systems: classification and key concepts
- Key concepts for technological systems

Why Study Technological Systems?

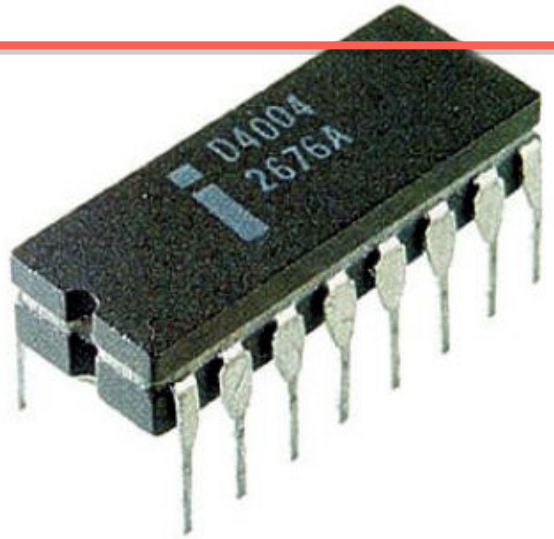
- Know what technology and technological systems can do for you
- Know the limitations of certain technologies
- Know how to solve technological problems
- Know when & how to get help and from what sources

Technological Systems - Examples

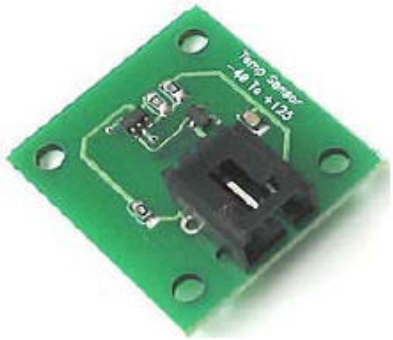
- Technology… Can be of many different kinds: electrical, mechanical, computer-based, hydraulic, etc.

- The systems of interest to us in the context of this course are **computer-based systems**
  - Can also be embedded systems (the processor is hidden – microcontroller systems)

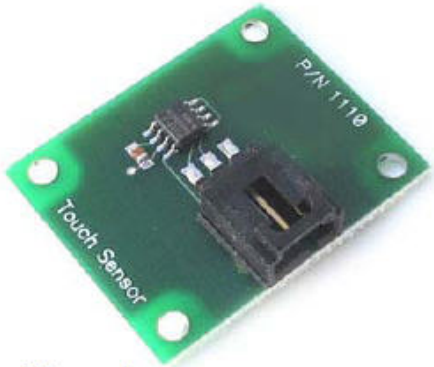
Embedded systems



- System controlled by an embedded processor
  (airplane, microwave, heart monitor, traffic lights)
  - **Microprocessor:** a CPU on a single IC
  - **Sensor**: a device that measures a physical quantity and converts it into a signal which can be read by an observer or by an instrument
  - **Actuators:** a mechanical device for moving or controlling a mechanism or system



Temperature sensor



Touch sensor

Force sensor

light sensor

Embedded Systems

Why study technological systems?

- **Abstraction**
  - Productivity enhancer – don't need to worry about details…

    Can drive a car without knowing how the internal combustion engine works.
  - …until something goes wrong!

    Where's the dipstick? What's a spark plug?
  - It is important to understand the components and how they work together.
- **Hardware vs. Software**
  - It's not either/or – both are components of a system.
  - Even if you specialize in one, you should understand capabilities and limitations of both.

What we need to know

- How to use the tool (e.g., entire system, programming language…)
- How to design systems
- How a certain system works
- How data is represented
- How an algorithm works on a computer / microcontroller
- How to collect data from sensors (inputs), process it (programs) and send it to actuators (output)

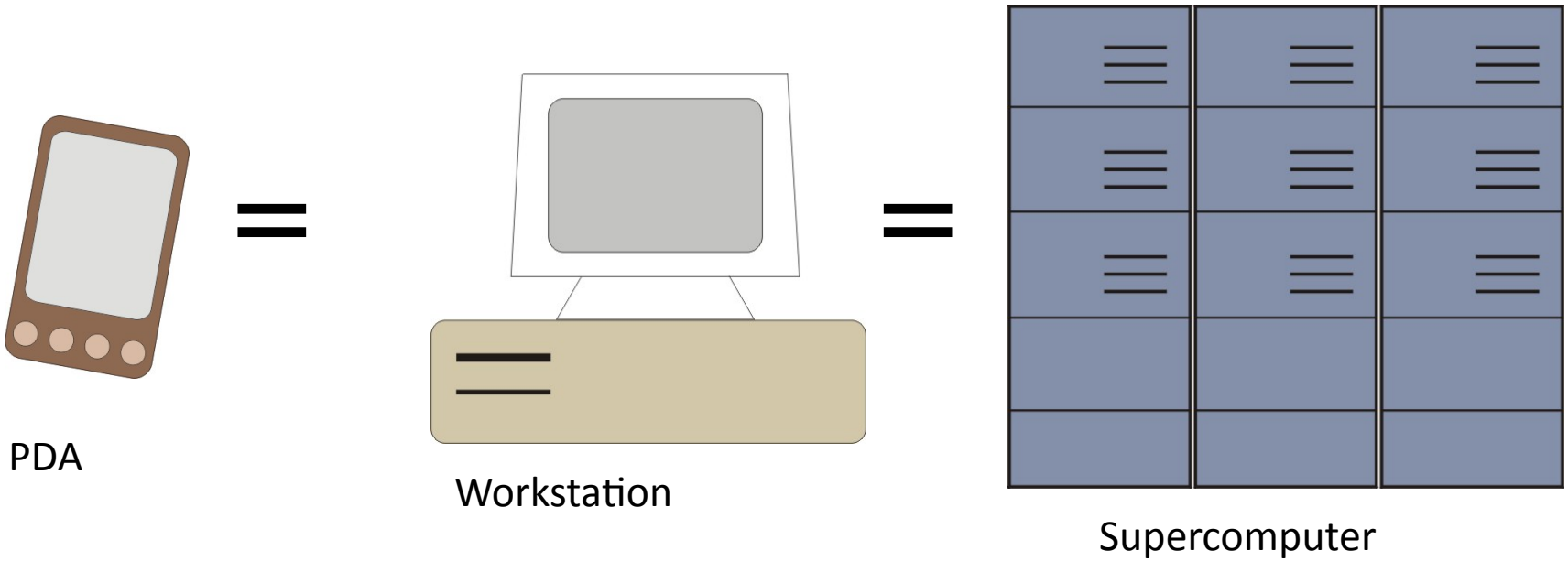
- Three directions of study:
  - Computer Technology – computer systems
  - Embedded Systems
  - Communication Technology - networking

Computer Systems

- Tools used to solve problems
- Different types of computer systems
- **ALL** computer systems are characterized by two general principles
  - **Universal computing device**
  - **Transformation between layers**

Big Idea #1: Universal Computing Device

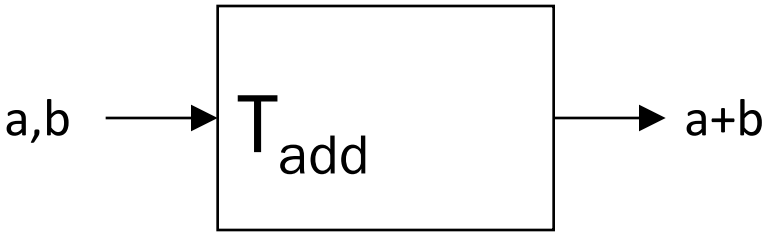
- All computers, given enough time and memory, are capable of computing exactly the same things.



Turing Machine

- Mathematical model of a device that can perform any computation – Alan Turing (1937)
  - ability to read/write symbols on an infinite "tape"
  - state transitions, based on current state and symbol

- Every computation can be performed by some Turing machine. *(Turing's thesis)*



Turing machine that adds

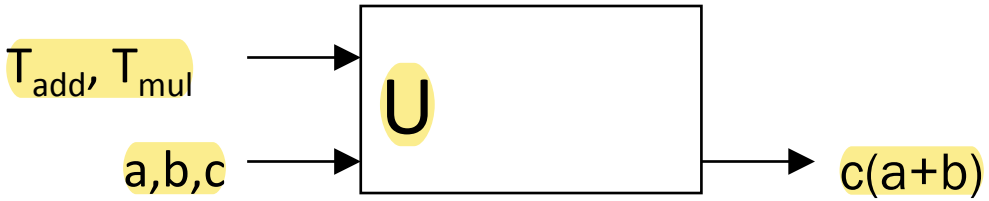
Turing machine that multiplies

Universal Turing Machine

A machine that can implement all Turing machines

-- this is also a Turing machine!

– inputs: data, plus a description of computation (other TMs)



Universal Turing Machine

U is programmable – so is a computer!

- instructions are part of the input data
- a computer can emulate a Universal Turing Machine

A computer is a universal computing device.

Problem Solving

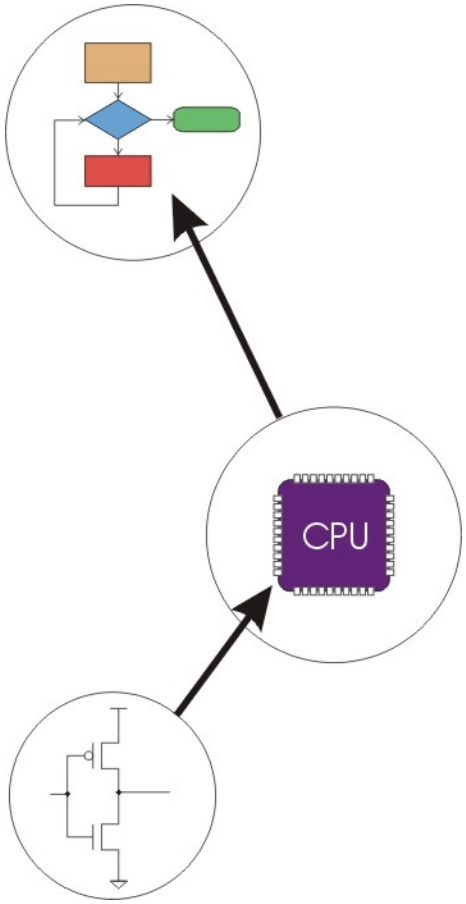
- Computer programs
- Knowing the base technology
- Describing the problem algorithmically
- Meaningful dialog between designer and computer programmer

From Theory to Practice

- In theory, a computer can compute anything that's possible to compute
  - given enough memory and time

- In practice, solving problems involves computing under constraints.
  - time
    - weather forecast, next frame of animation, ...
  - cost
    - cell phone, automotive engine controller, ...
  - power
    - cell phone, handheld video game, ...

Big Idea #2:
Transformations Between Layers



Algorithms

Language

Instruction Set Architecture

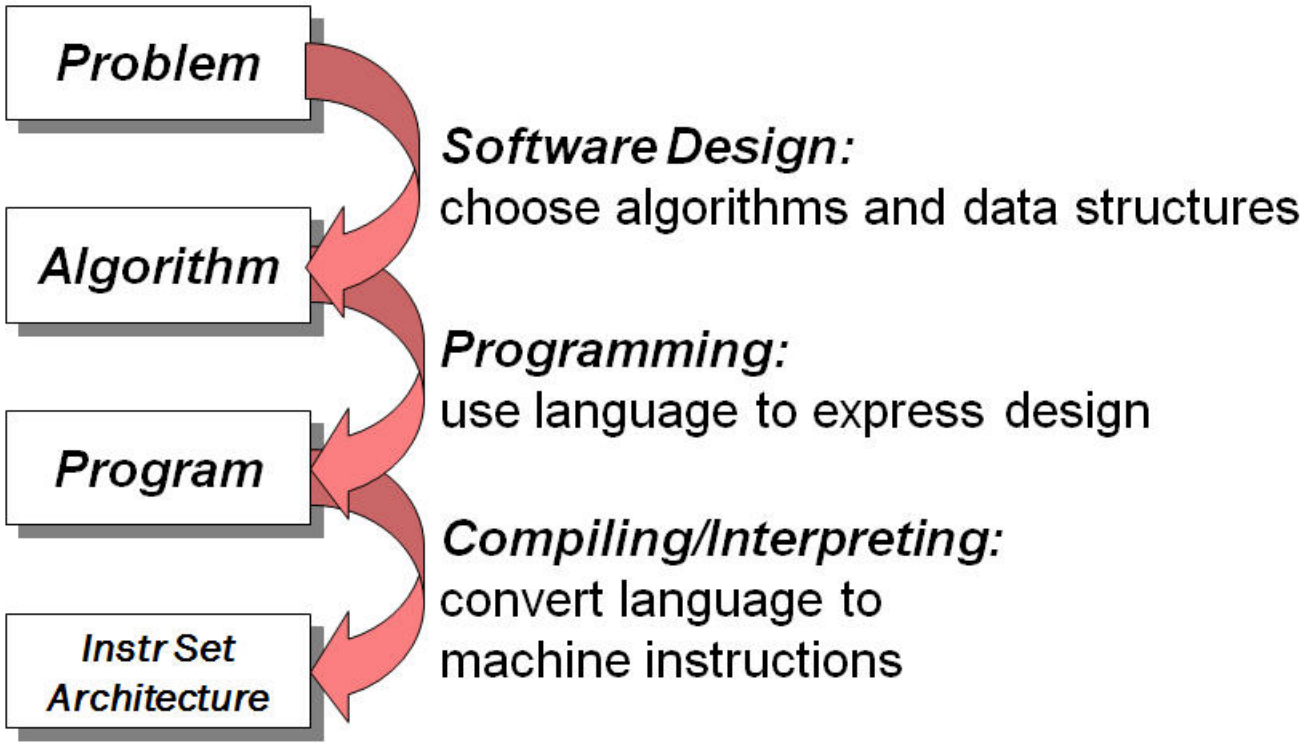
Microarchitecture

Circuits

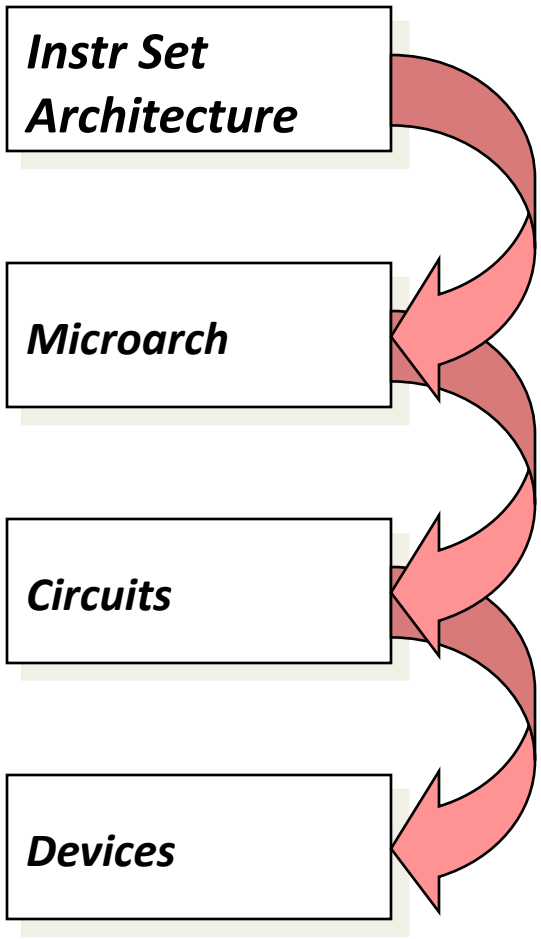
Devices

How do we solve a problem using a computer?

- A systematic sequence of transformations



At deeper levels…



Processor Design:
choose structures to implement ISA

Logic/Circuit Design:
gates and low-level circuits to implement componnts

Process Engineering & Fabrication:
develop and manufacture lowest-level components

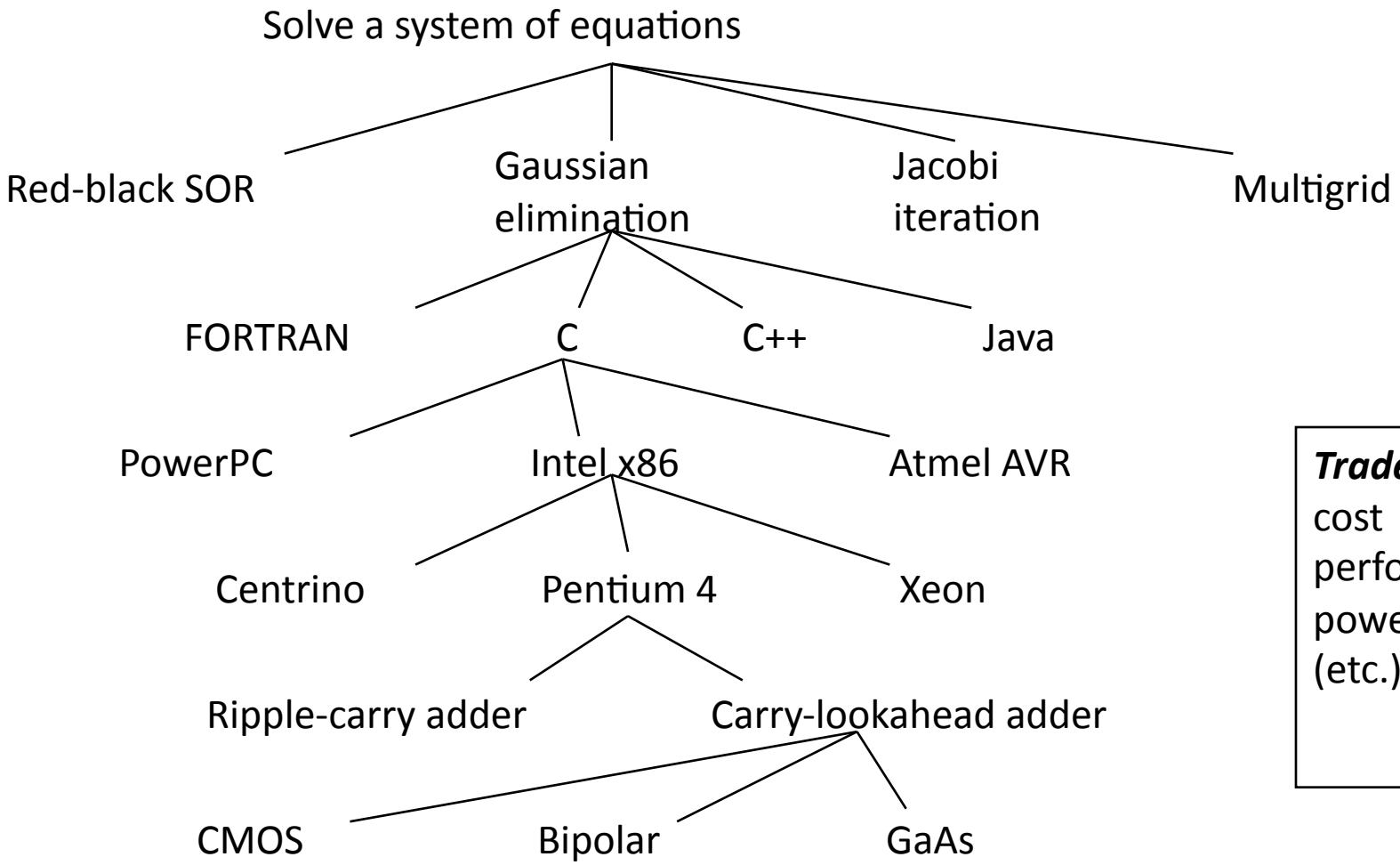
Description of Each Level

- Problem Statement
  - stated using "natural language"
  - may be ambiguous, imprecise
- Algorithm
  - step-by-step procedure, guaranteed to finish
  - definiteness, effective computability, finiteness
- Program
  - express the algorithm using a computer language
  - high-level language, low-level language
- Instruction Set Architecture (ISA)
  - specifies the set of instructions the computer can perform
  - data types, addressing mode

Description of Each Level (cont.)

- Microarchitecture
  - detailed organization of a processor implementation
  - different implementations of a single ISA
- Logic Circuits
  - combine basic operations to realize microarchitecture
  - many different ways to implement a single function (e.g., addition)
- Devices
  - properties of materials, manufacturability

Many Choices at Each Level

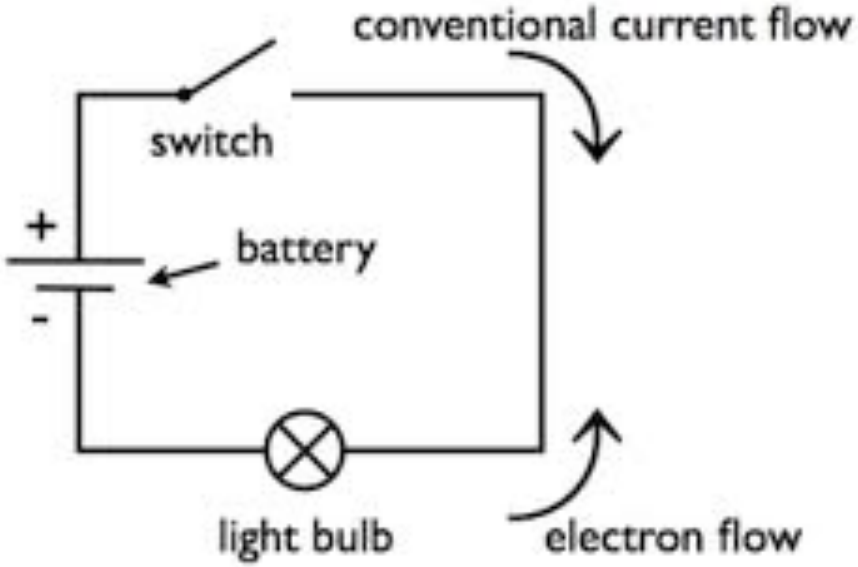


Tradeoffs:
cost
performance
power
(etc.)

In the workshop:

we will start our study at the lowest level:

circuit level



Computer Systems: 3 Key Concepts

- Purpose of a computer
  - Turn **data** into **information**
  - Data: the raw facts and figures
  - Information: data that has been summarized and manipulated for use in decision making

- Hardware and Software
  - Hardware is the machinery and equipment in the computer
  - Software is the electronic instructions that tell the computer how to perform a task

Computer Systems: 3 Key Concepts

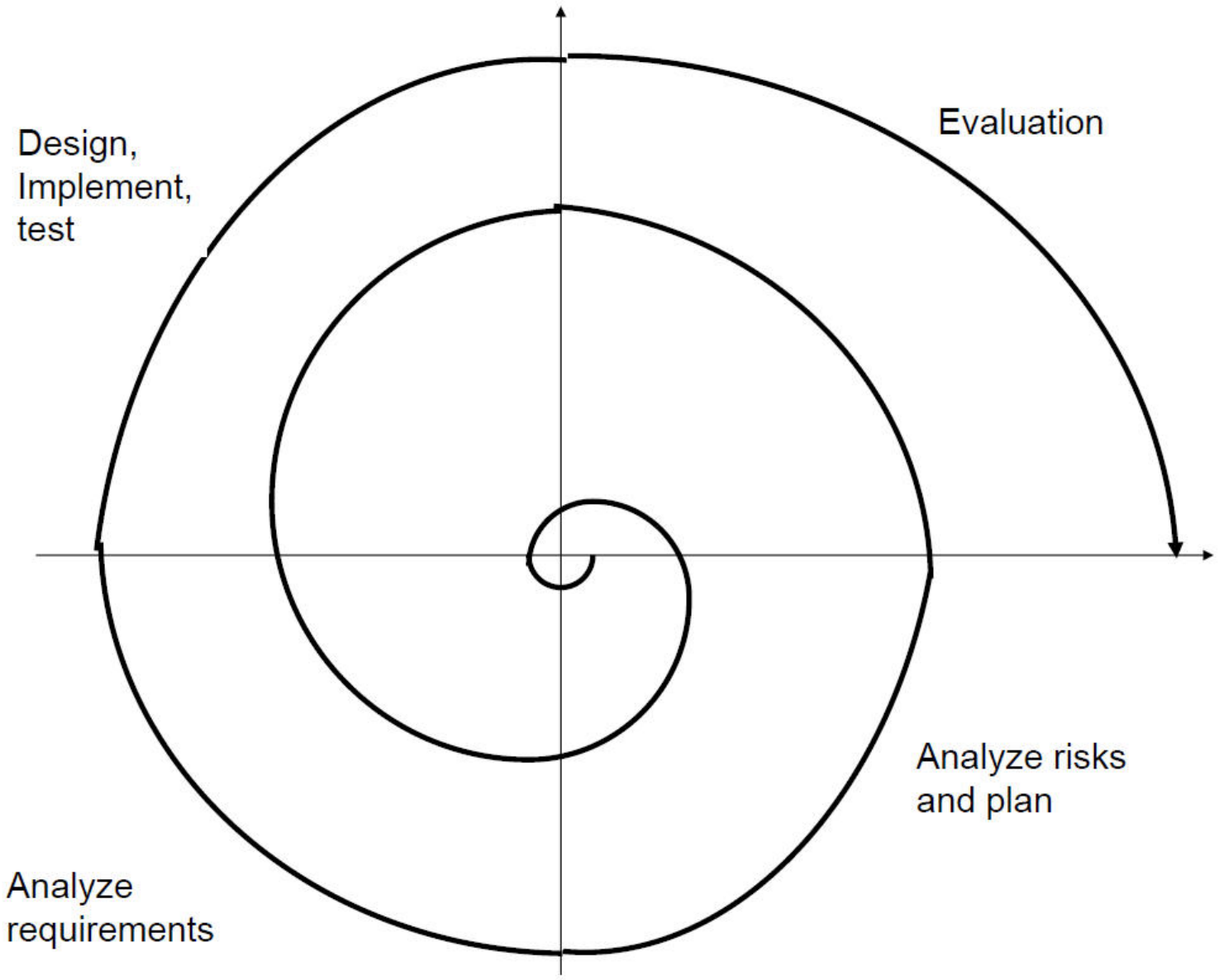
- The basic operations
  - **Input**: What goes in to the computer system
  - **Processing**: The manipulation a computer does to transform data into information
  - **Storage**:
    - Temporary storage: Memory is primary storage
    - Permanent storage: Disks and media such as DVDs and CDs are secondary storage
  - **Output**: What comes out
    - Numbers or pictures on the screen, printouts, sounds
  - **Communications**: Sending and receiving data

Technological Systems: Key Concepts

- Iterative Design
- Trade-offs
- Real-world constraints
- Feedback
- Complexity management techniques

Iterative Design

- Imagine something → design it → build it → get it to work
- Exposure to the design cycle
- Even the process of choosing a project requires several iterations


Design,
Implement,
test
Evaluation
Analyze risks
and plan
Analyze
requirements

Trade-offs

- Multiple interacting domains and subsystems in a project → solving one problem can create others
  - Space/time trade-off: By compressing an image you can reduce transmission time/costs at the expense of CPU time to perform the compression and decompression.
- Important aspect of trade-offs: they make it clear that there is no single 'right' or 'best' solution; rather each solution comes with its own benefits and drawbacks.

Real-world Constraints

- Design of a system: in many cases involves multiple interacting parts and domains, each with its own issues
  - Example: comparison between the 'bin-sorting' problem in computer science and the task of designing a device that would sort marbles in different bins

Single Domain vs. Multiple Domains

- **Bin-sorting problem (computer science perspective)**

- Purely computational - developing an algorithm that solves the problem, using the least amount of time and computer memory

- Also called **Bucket-Sorting**: works by partitioning an array into a number of buckets:
  - Set up an array of initially empty "buckets.“
  - Scatter: Go over the original array, putting each object in its bucket
  - Sort each non-empty bucket
  - Gather: Visit the buckets in order and put all elements back into the original array

Physical sorting of marbles into bins: several issues

- Sensor system that would differentiate the marbles, based on certain parameters
- Marbles have mass and volume and need to be transported to the correct bin
- Determine when the correct bin has been reached
  - How to insert the marble into the bin
  - Computation, but time and space requirements will not be primary concerns

Feedback

- Common in natural systems, engineered devices
- Examples:
  - **Negative feedback**: ball on the botton of a hill: if perturbed from this position, it will roll back to the bottom (**back to the initial state**)
  - **Positive feedback**: ball on the top of a hill: if perturbed from this position, it will roll further away from the top of the hill (**away from the initial state**)

Feedback

- Control systems involving negative and positive feedback
  - Example: a light-seeking robot, with photocell sensors serving as 'eyes'
    - If the right eye senses more light than does the left, the robot rotates towards the right; similar for the left eye
    - If the two eyes see the same amount of light, the robot does not rotate
  - This is an example of **negative feedback** (deviation from a state drives the system back to that state).
  - The same example can be used to illustrate **positive feedback** (deviations from a state drive the system further away from that state): start with an orientation of both eyes to a point where they detect the same amount of darkness

More examples of feedback

Positive feedback: The Tacoma Narrows Bridge collapsed in 1940, due to a design flaw that allowed positive feedback to dominate.

Negative feedback: Thermostat
When the temperature in a heated room reaches a certain upper limit the room heating is switched off so that the temperature begins to fall. When the temperature drops to a lower limit, the heating is switched on again. Provided the limits are close to each other, a steady room temperature is maintained.

Positive feedback: Alarm or panic can spread by positive feedback among a herd of animals to cause a stampede

Source: Wikipedia

Controlling Complexity

- Abstraction: 'black-box' entities with simple interfaces
  - Programmable devices (microcontrollers, field-programmable logic devices) that can be used effectively without having to completely understand the details of how they work.
- Modularity: composing systems of reusable, mix-and-match parts
  - E.g., small circuit design to perform a specific function: microcontroller + temperature sensor + servomotor used to control temperature – can be reused in several courses and different projects

Summary of key concepts

- Technological Systems:
  - Iterative design
  - Trade-offs
  - Managing complexity
  - Real-world constraints
  - Feedback
- Computer Systems
  - Universal computing device
  - Transformations between layers

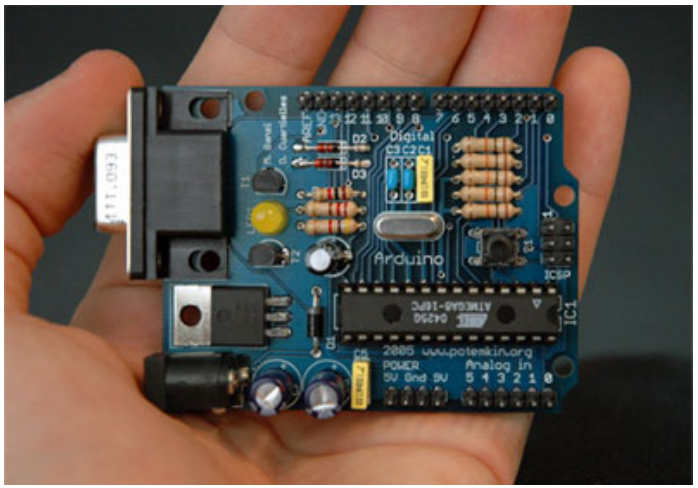
Directions of study in this course

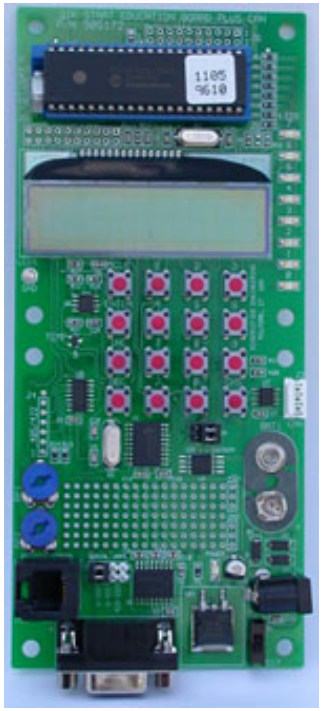
- Start with computer systems
  - Study hardware
  - Software (system, application)
  - Relationship between the above

- Extend the concept to more complex systems that will include the computer system as a central part → make use of the processing power of the computer
  - Connect sensors, for example → computer system gains the ability to sense the environment
  - Input from sensors can be processed using the CPU of the computer system
  - Add a microcontroller (Arduino for example) to gather information from sensor, process this information with a computer system and generate some output

Directions of study (cont'd)

- Technological system: does not necessarily need to have a computer (workstation / microcomputer) in its structure
  - Processing of data can be done for example by an embedded computer
  - Example: use of a microcontroller that will take input data from sensors and process it
    - Home temperature and light control system – (prototype shown on the next slide)


Digital
Arduino
2005 www.potemkin.org
POWER
Analog In



Your first quizzes

- On WebCT :
  - Read through the Plagiarism Tutorial

- Quiz: Citing
- Quiz: Writing Skills
  - (can be found under 'Assessments')

- These two quizzes are part of your participation mark.

- Due: next Wednesday (September 14th).

Thank you

Questions?