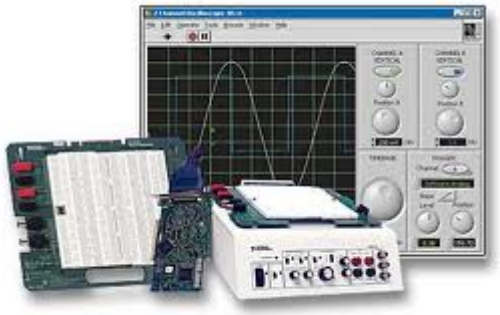
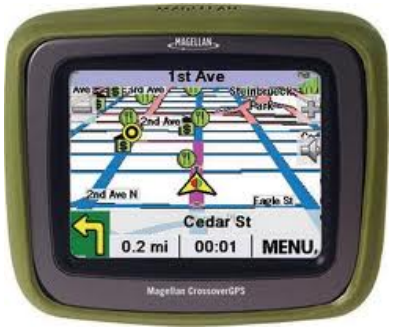
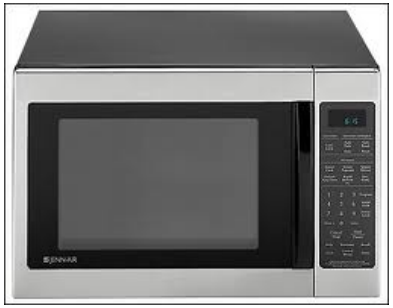
Introduction to Embedded System

Chapter 1

They are Embedded Systems







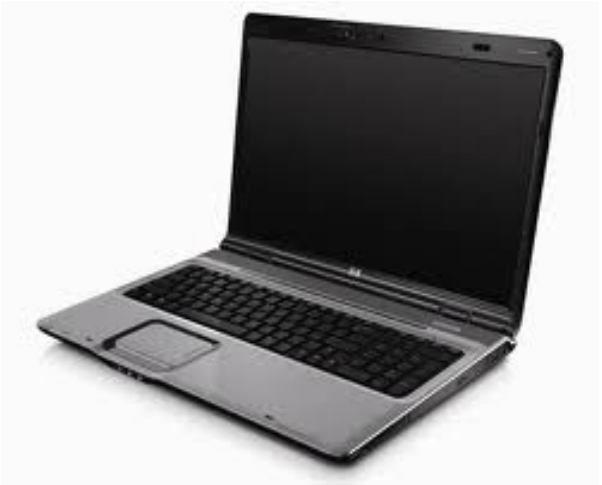
Advanced Embedded Systems

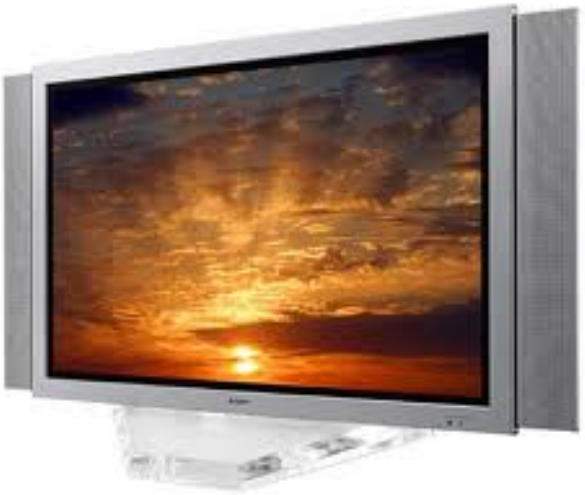
- Robots
  - Complexity: computing, mechanic, intelligence
- Fly (Festo)
  - https://www.youtube.com/watch?v=nnR8fDW3Ilo
- Climb (Boston Dynamic, ETH)
  - https://www.youtube.com/watch?v=XEMlkonimvQ
  - https://www.youtube.com/watch?v=ps06qU9-Dpc
- Jump (MIT, Boston Dynamic)
  - https://www.youtube.com/watch?v=_luhn7TLfWU
  - https://www.youtube.com/watch?v=WcbGRBPkrps
- Swarm (ETH)
  - https://www.youtube.com/watch?v=RCXGpEmFbOw
- Many other advanced embedded systems

Definition

- An embedded system is a computerized system that is purpose-built for its application.
  - **Computerized: computation is core**
  - **Purpose-built: software and hardware components**
  - **Application: functionality, requirement, specification**
- Comparison
  - General purpose computer
    - Not purpose-built
    - No specific applications
  - Super computer
    - Purpose-built: high performance computing
    - Not for specific applications, but for providing computational capability

Is it an embedded system?





Characteristics (I)

- Designed to perform dedicated functions for specific applications
  - LED TV
  - A remote controller, a signal receiver, a panel
  - Although the controller has many buttons, the TV only has a fixed menu with a fixed set of functions.
    - Change channel
    - Recording
    - Video on demand
    - Change color
  - So, it is feasible to have a manual with a TV, which describes all functions provided by the TV.

Characteristics (II)

- Resource constraints in hardware
  - Sensor
  - Hardware : 4KByte ram, 1MIPS CPU
    - Voice recording
      - 8 bit per sample, 9.6K samples per second (only fair quality)
      - Without compression : 0.41 seconds of recording
      - With compression, each sample should be compressed within 104 instructions.
  - Battery :
    - 1 AAA battery : 1000mAhr
    - Run for 1 year : 6360 hours (0.15mA in average)
      - 20mA of operation and 10uA of sleep
      - 20x + 0.01(6360-x) = 1000, so 46.84 hrs of operation
      - 0.74% of time for operation
      - fewer than a half minute per hour of operation (26.6s)

Characteristics (III)

- Designed with specific quality requirements
  - Deterministically : exactly the same each time
    - Sensor
  - Real time : always reacting to an event fast enough
    - Brake
  - Fault tolerant : tolerable quality degradation in the face of errors
    - Unmanned aviation vehicle (UAV)
  - Agile : instant change of operations on a sudden signal
    - Uninterruptible power supply (UPS)

Characteristics (III)

- Examples:
  - Car's brake controller
    - ABS, ESC, ...
    - 60 mph = 87.5 feet per second
    - A response time of 0.1 second in the brake system means a distance of 8.75 feet.
    - High real-time requirement
  - Car's engine controller
    - Gas and ignition control in response to current speed
    - High reliability requirement : cannot fail over years
  - Airplane : higher
  - Digital TV : lower

Characteristics ?

- Modern general purpose embedded systems
  - PDAs, cell phones, ...
  - They were NOT general purpose embedded systems, but are now.
  - Limited resources
    - ?
  - Dedicated functionality
    - ?
  - Specific quality requirements
    - ?

System Design

- General design principles
  - Start with applications
  - What functionality (Characteristic I)
    - Determine the function units of software and hardware
  - What requirements (Characteristic III)
    - Determine the performance and quality of software and hardware
  - What restraints (Characteristic II)
    - Determine the limit and cost of software and hardware
- Design case
  - Application: obtain power consumption data in computer
  - Embedded system: power sensor in computer

Design Case

- Functionality of application
  - Get raw data
    - Power consumed by CPU, memory, hard drive, …
    - About 30 power lines to get data in a computer
      - Need a MCU of more than 30 ADC channels
  - Process raw data
    - Processing integer only with arithmetic computation
      - No need of floating point or digital signal processing (DSP)
      - Need a standard MCU
    - Data could be send via USB or network
      - Need a MCU to support USB and network
      - Need software libraries to support USB and network

Design Case

- Requirement of application
  - Data precision: 8-bit or above and 5% error margin
    - Need a 16-bit MCU or better
  - Processing rate: 1 ~ 200 samples per channel per second (6400 samples per second maximum)
    - Assume each data sample processing needs 1000 clocks
      - This is also a software implement requirement
    - Real-time: need a MCU working at 10MHz or above
    - No data caching: no data memory requirement
  - Power requirement
    - Sensor powered by computer chassis
    - In-chassis temperature: 20-60 degree celsius
    - Need a MCU of <5W at 3.3V

Design Case

- Functionality of application
  - Get raw data
    - Power consumed by CPU, memory, hard drive, …
    - About 30 power lines to get data in a computer
      - Need a MCU of more than 30 ADC channels
  - Process raw data
    - Processing integer only with arithmetic computation
      - No need of floating point or digital signal processing (DSP)
      - Need a standard MCU
    - Data could be send via USB or network
      - Need a MCU to support USB and network
      - Need software libraries to support USB and network

Design Case

- Restraints of application
  - What you are going to pay (money, time, people)
  - Production cost: <$100 per sensor set
    - Need a MCU at $10 or less
  - Development cost:
    - Software license: free or open-source
    - Prototyping cost: prototype manufacturing
  - Development time: <12 months
    - Need well established libraries, documents and tools
  - Deploy, update and maintenance
    - One or two students to deploy
    - Remote access only for update and maintenance
      - Need a MCU supporting remote control and update
      - Need software including remote control and update

Challenges in Development

- Resources constraints:
  - Memory and code space
  - Processor cycles and speed
  - Power consumption
  - Peripherals
  - Compromise in design: tradeable resources
    - Code space vs processor cycles
    - Processor speed vs power consumption
- Uncertainty in hardware
  - Hardware is buggy and not reliable

Challenges in Development

- Cost
  - Hardware cost
    - Hardware design
    - PCB manufacture
    - Board assembly
    - Parts cost
    - Shipping
    - Custom and tax
  - Software cost
    - Software development
    - License
    - Testing
  - Support and upgrade

Confront Challenges

- Flexible design
  - Modularity, encapsulation
    - Break system into parts
  - Testing
    - Write test code for each part
  - Documenting
    - What if you will revisit code after a year
  - Tune later
    - Make it happen first
    - Optimize later