ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA ĐIỆN-ĐIỆN TỬ
BỘ MÔN KỸ THUẬT ĐIỆN TỬ

Embedded System Design

Chapter 2: Develop a project of embedded system design

1. Design Process
2. Design Issues
3. Project plan

1. Challenges in embedded system design

How much hardware do we need?
How do we meet deadlines?
How do we minimize power consumption?
How do we design for upgradeability?
Does it really work?

Challenges

- NRE (Non-recurring Engineering Cost)
- Size
- Power
- Performance
  - Latency, through put
- Flexibility

Throughput vs Latency

- "How fast do the samples need to be acquired?" usually translates to throughput
- **Throughput** is the rate at which the system can process inputs. It is an amount of measurements per a given time.
- Measure by samples per second (S/s), though it is useful to note that many computer components are rated in B/s, MB/s, GB/s, etc.
- One common synonym is the term bandwidth.

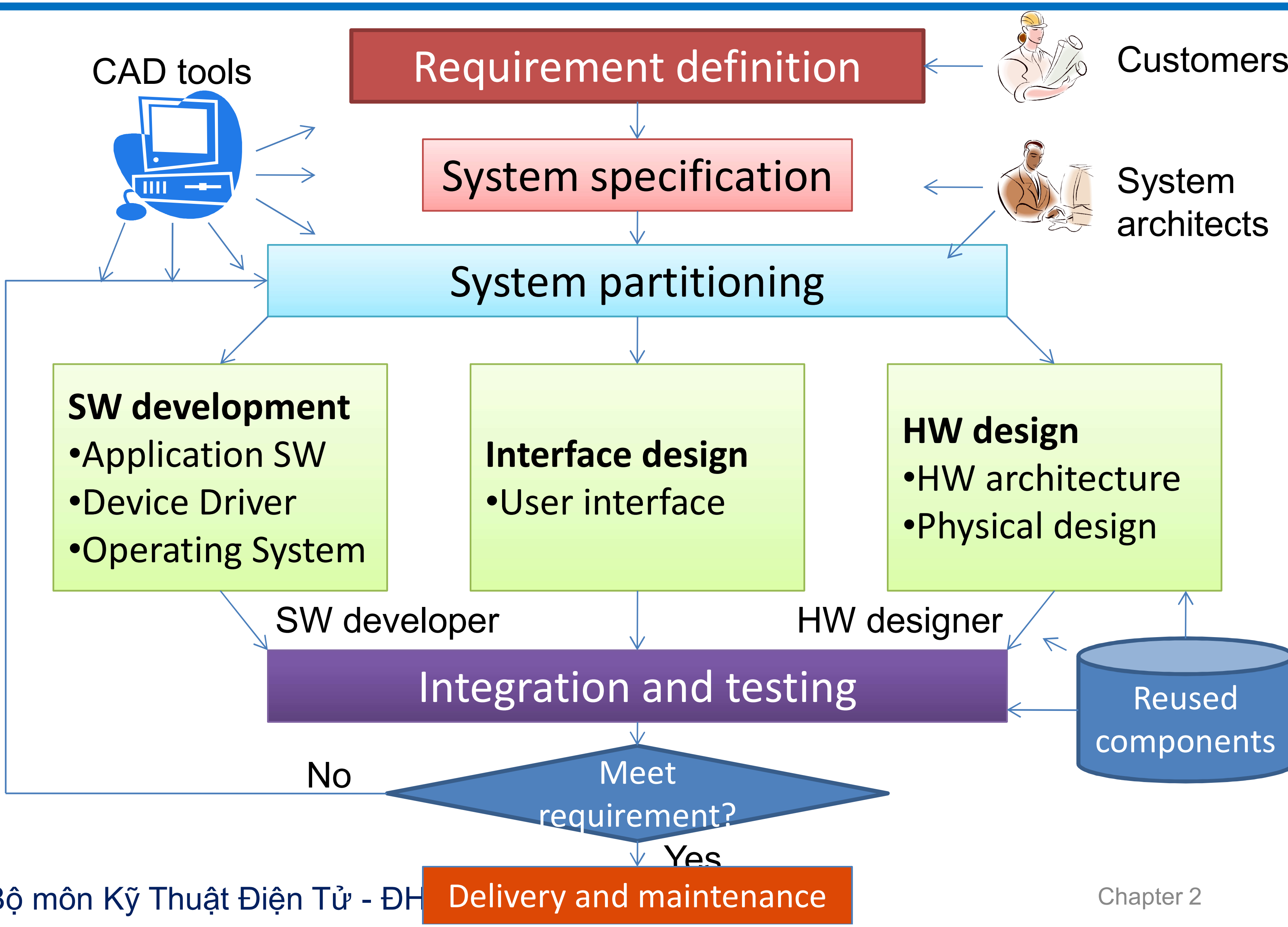
Throughput vs Latency

- "How fast does a result need to be available after a sample?" usually translates to latency.
- **Latency** is the amount of time it takes to complete an operation. It is measured in units of time (s, ms, ns).
- In many applications such as control applications, it is the time from an input to its corresponding output

Difficulties in Design and Development

- Complex testing
  Run a real machine to have proper data
  System must be tested in the embedded machine
- Limited observability and controllability
  Sometimes no keyboard or screen!
  In real-time systems it's not easy to stop the system to see what is going on
- Restricted development environments
  Much more limited than in PCs
  Usually compile code in PC and download it to embedded system

1. Embedded System Design Process



1.1. System Specification

Documents for System Specification

1. Product Requirement	Describe how the product will be.
⇩	
2.Design Specification	Describe hardware modules, subsystem, and functions will be used.
⇩	
3. Hardware Specification	Describe how the board will be implemented and how it works.
⇩	
4. Software Specification	Describe how the software will be implemented.
⇩	
5. Test Specification	Describe how the system will be tested.

Design Goals

Performance

Overall speed, deadlines

Functionality and user interface

Manufacturing cost

Power consumption

Other requirements (physical size, etc.)

1. Product Requirement

⇓

2.Design Specification

⇓

3. Hardware Specification

⇓

4. Software Specification

⇓

5. Test Specification

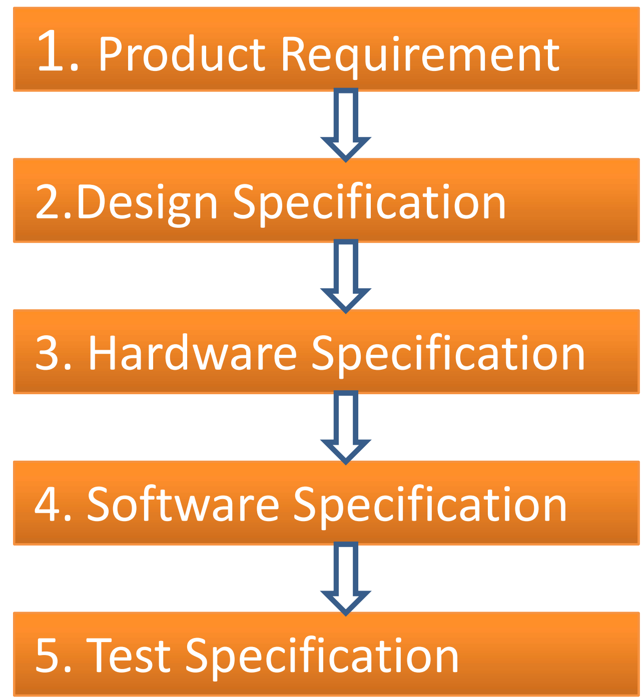
Stepwise Refinement

- At each level of abstraction, we must

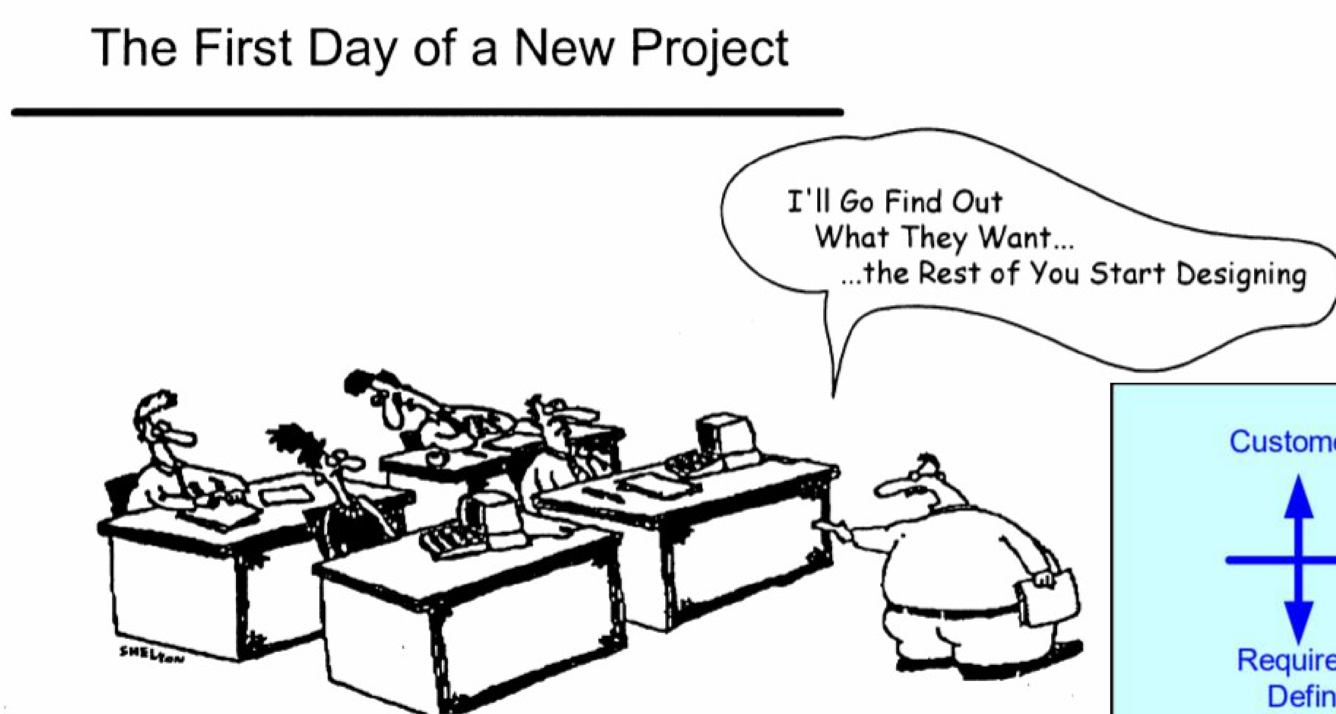
  analyze the design to determine characteristics of the current state of the design

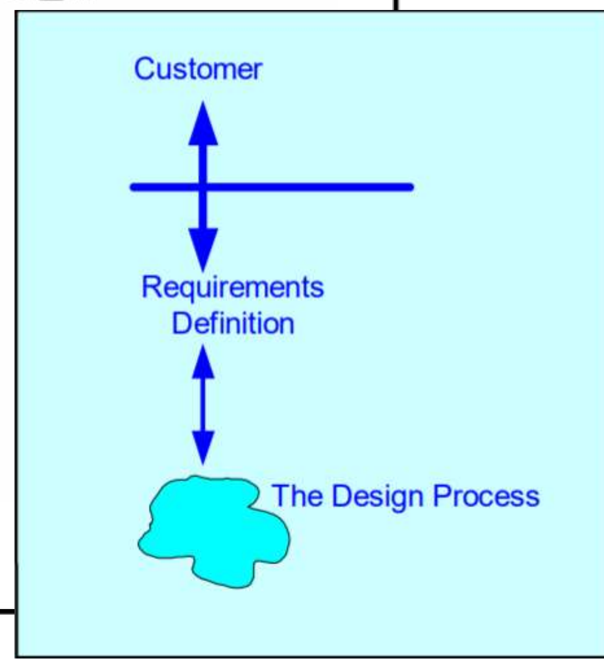
  refine the design to add detail

  verify that it meets all system goals cost, speed, . . .



Identifying the Requirements





Requirements

A user requirement is a statement that specifies WHAT a product should do, but it does not define HOW it should do it

WHAT:

The device shall decrease the temperature of the skin

HOW:

The device shall apply moisture to cool down the skin.

Requirements

May be developed in several ways

- Talking directly to customers
- Talking to marketing representatives
- Providing prototypes to users for comment

Requirements end up being the specification

- Containing enough information to begin designing the system architecture

Functional vs. Non-functional Requirements

Functional requirements
- output as a function of input

Non-functional requirements
- time required to compute output
- cost
- size, weight, etc.
- power consumption
- reliability

Non-functional Requirements

Performance

Major consideration for the usability of the system and its ultimate cost

May be a combination of soft performance metrics and hard deadlines

Cost

Manufacturing costs (e.g. components, assembly)

Nonrecurring engineering (NRE) costs (e.g. personnel, designing the system)

Environment and Installation

The environment that the device will operate

How to install the device

Those requirements are important for designing the PCB, the enclosure.

Non-functional Requirements

Physical size and weight

Depends on the application

Power consumption

Important not only in battery-powered systems

Specified in terms of battery life or Wh

Certification

For international market, device need to have certification to enter the market.

Several important certifications are UL, FCC, CE, ROHS, IP

Validating The Requirements

- Requires understanding what people want and how they communicate it
- User interface requirements can be refined by using a mock-up

    may be executed on a PC

- Physical, nonfunctional models of devices can also help

    better idea of size and weight

Sample Requirements Form (1/3)

- name
- purpose
- inputs and outputs
- Use cases
- functions
- performance
- manufacturing costs
- power
- physical size/weight
- Installation
- Certification

Sample Requirements Form (2/3)

- name
- purpose

  one- or two-line description
- inputs and outputs

  types of data: analog? digital? mechanical?. . .

  data characteristics: periodic? occasional? how many bits?. . .

  types of I/O devices: buttons? A/D converters? video displays?. . .

Sample Requirements Form

- Use case

  How user will interact with the system

- functions

  more detailed description of the system

  when the system receives an input, what does it do?

  how do interface inputs affect these functions?

  how do different functions interact?

- performance

  must be identified earlier to ensure that the system works properly

Sample Requirements Form

- manufacturing costs
  - cost has substantial influence on architecture
  - work with some idea of the cost range
- power
  - battery powered? plugged into a wall?
- physical size/weight
  - more or less flexibility in the components to use
- Installation
  - Device is fixed, wall mounted or on desk, etc

Sample Requirements Form

- Certification
  - Need to meet standards for safety, compatibility.
  - Some Certification is UL, CE, FCC, IP

Validating The Requirements

- The requirements form should be the introductory of a longer document
- After writing the requirements you should check for internal inconsistency
    - forget to assign functions to an input/output?
    - considered all modes of operation?
    - unrealistic number of features into a battery-powered, low-cost machine?

Requirements

- name
- purpose
- inputs and outputs
- Use case
- functions
- performance
- manufacturing costs
- power
- physical size/weight
- Installation
- Certification

Digital Alarm Clock Requirements

- Description:
  - This specification describes and defines the basic requirements of a digital alarm clock. The clock resolution is second, with error rate of 1 sec per day. The clock can have 10 Alarm time.
- External Environment:
  - The Clock will be used indoor, put on desk or bed side

inputs

- inputs:

  4 small buttons: MENU, UP,DOWN, ENTER

  1 big button: Stop Alarm

Output

- Display:
  - LCD with back light
  - Date is display as Day/Month/Year
  - Time can be display as am/pm or 24h, up to second.

Output

- Display (cont)

  The display should be readable in direct sunlight and from wide angle.

  The display will appear similar to this one

Output

- Display (cont)

  Power Saving Mode

  - Day/Night Mode
  - Adaptive Brighness mode

- Sound

  The speaker can generate a sound at the volume of 87dB, measured by a microphone placed 1 meter away

Use Cases

“Specifies a sequence of actions, including Specifies a sequence of actions, including variants, that the system can perform and that variants, that the system can perform and that yields an observable result of value to an yields an observable result of value to an actor ”

Use Cases

- Normal run

Brief Description

This mode is the normal mode of the clock, where continuously update the time/date on the display and check for alarm.

Basic Flow

The display is updated every 1 second.

If the alarm time is reached, the clock will sound and vibrate according to the configuration.

Requirement

None

Use Cases

- Change mode

  Brief Description

  Perform the mode changing.

  Basic Flow

  When power on, the clock enter run mode. If user press MENU, it display SETTIME mode.

  If user Press ENTER, it will enter SETTIME mode. If user press MENU again, it display SETDATE mode. If user press MENU again, it display SETTIMEMODE mode. If user press MENU again, it display SETALARM MODE mode. . If user press MENU again, it return to RUN MODE mode

  Requirement

  Time to press the button is longer than 0.5s.

Use Cases

- Set Time

Brief Description

Perform the set current time function.

Basic Flow

First, the Hours display will blink. If user press Up button, it will increase. If user press Down button, it will decrease. If user press Enter Button, the current value is saved and the Minute display is blink. The same thing happen and if user press Enter, the Minute value is saved and the Second display is Blink. If user press Enter, the Hour display will blink again.

If user press MENU, it go to the SETTING mode

Requirement

Time to press the button is longer than 0.5s.

Use Cases

- Set Time

  Brief Description

  ....

  Basic Flow

  ....

  Requirement

  ....

System Functional Specification

- Display time

  Description

  Depend on the display mode, time is displayed by 24h or am/pm mode

  Requirement

  Time is updated on the LCD every second

System Functional Specification

- Alarm

  Description

    If the Alarm time is reached, the Clock will sound and vibrate with the set volume.

  Requirement

    Sound is 87 Db at 1m away

    Vibrate is 0.6g

System Functional Specification

- Time Count

  Description

  ....

  Requirement

  ....

Non Functional Specification

- performance:
  - Can save 100 Alarm tone, maximum length of each tone is 10 sec.
- Manufacturing Cost

  40 USD per piece for 1000 in quantity

Non Functional Specification

- Physical size/weight
  - Maximum 250g (including battery, enclosure)
  - Maximum outer size of the clock is 20x20 mm.
- Power supply
  - Adaptor 5V
  - Operate for a minimum of 7 days on a fully charged battery.
- Installation and working environments
  - Put on desk and bed.
  - Indoor
  - Temperature range: 10 – 80 Celcius

Non Functional Specification

- Reliability and Safety Specification :

  The Clock shall comply with the appropriate standards

  - FCC, CE.

  MTBF: 3 years

  (Mean time between failures)

Requirements Validation

- **Validity** *checks*: The functions proposed by stakeholders should be aligned with what the system needs to perform.
- **Consistency** *checks*: Requirements in the document shouldn't conflict or different description of the same function
- **Completeness** *checks*: The document should include all the requirements and constrains.
- **Realism** *checks*: Ensure the requirements can actually be implemented using the knowledge of existing technology, the budget, schedule, etc.
- **Verifiability**: Requirements should be written so that they can be tested. This means you should be able to write a set of tests that demonstrate that the system meets the specified requirements.

EXERCISE

- Group discussion
  - Finish the Digital Alarm Clock requirement

Design Specification

- System description
  - Written from a designer point of view
  - System architecture
  - Block diagram if appropriate
- First order functional decomposition
- System behavior
  - Control flow
  - Data flow
  - Signal flow
  - Timing requirements
  - State diagrams
  - Constraints

Design Specification

- Format, structure, and type of the system inputs and outputs
  - Signal levels
  - Timing / critical timing
  - Command, Control, Data Formatting
- Memory Maps
- Major modules and their interfaces

1.2. System partitioning

- System partitioning: divide the system into three parts:
  - Hardware (HW): microcontrollers, memories, peripherals,
  - Software (SW): OS, program, application software
  - Interface: SW driver, HW interface, user interface

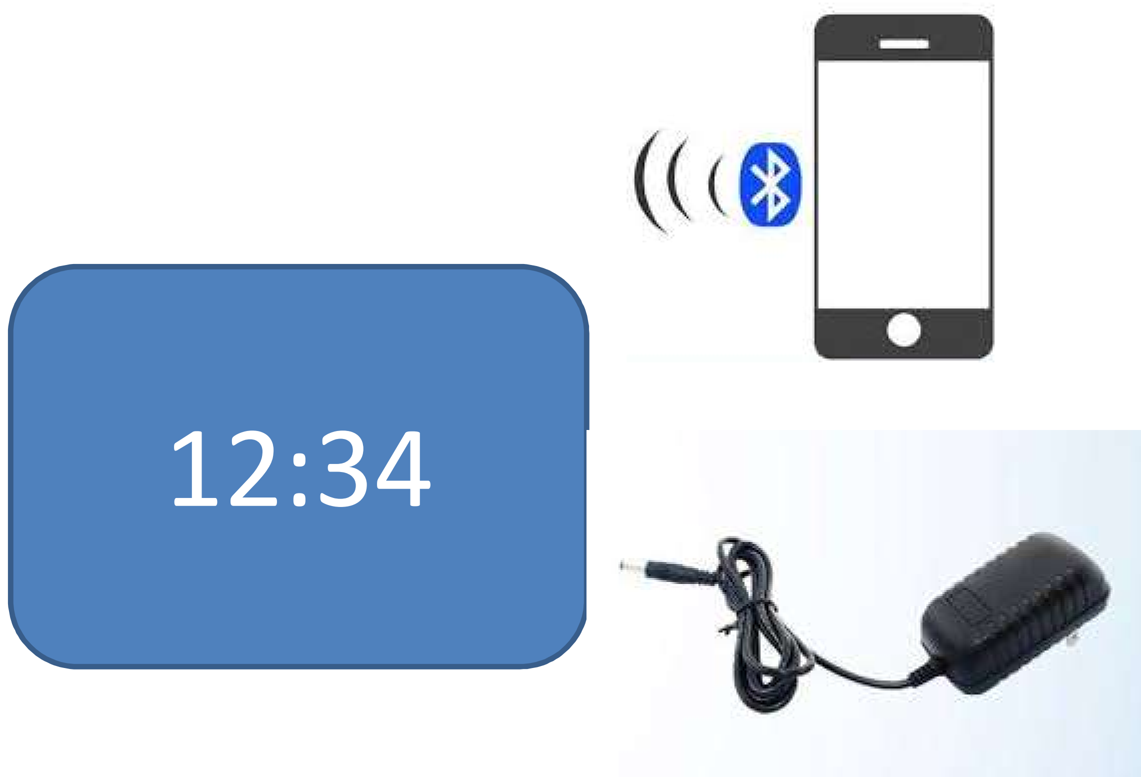
- Alternative way: divide the system into HW and SW
  - which functions should be performed in hardware, and which in software?
  - the more functions in software, the lower will be the product cost

Design Specification Example

- Description:
  - This specification describes and defines the basic requirements of a digital alarm clock. The clock resolution is second, with error rate of 1 sec per day. The clock can have 10 Alarm time. It supports sound alarm.
- External Environment:
  - The Clock will be used indoor, put on desk or bed side
  - Temperature range is 10-35 Celcius

System Overview

The Clock can be powered by an 5V adaptor or by internal battery. User can use cellphone to download alarm tones or configure the clock.



Functional design

- Display time

  Description

  Depend on the display mode, time is displayed by 24h or am/pm mode.

  Requirement

  Time is updated on the LCD every second

Functional design

- Display time

	Hardware	Requirement and note
	LCD TFT 2.8 inch	Can choose SPI interface to reduce pin count
	Backlight Control Circuit	Allow dimming

	Software and spec	Note
	LCD driver routines	

Layout

Depend on the display mode, time is displayed by 24h or am/pm mode.

Functional design

- Alarm

  ### Description

  If the Alarm time is reached, the Clock will sound with the set volume.

  ### Requirement

  Sound is 87 Db at 1m away

Functional design

- Alarm (cont)

	Hardware	Requirement and note
	Speaker 1W 8Ohm	Sensitivity 87dB Size maximum 5mmx5mm
	Audio power amplifier	Can support 1W
	DAC to generate sound	8 bits
	EEPROM to store sound	

	Software and spec	Note
	Alarm time compare	Can be hardware
	Sound generate routines	
	DAC driver	

Non-Functional design

- performance:
  - Can save 10 Alarm tone, maximum length of each tone is 10 sec.

	Hardware	Requirement and note
	ROM to store sound	ROM capacity is 10*10*8KB

Hardware Considering

Item	Hardware	Description	Importance	Cost
	Flash	1Mbyte Flash	Must have	0.5USD
	Audio Codec	High quality audio processor	Nice to have	2USD
	RTC	Realtime clock	Must have	1 USD
	MCU	(Depend on the hardware and software requirement)		

Micro Controller Selection

	Peripheral	Interface	Pincount	Requirement
1	Buttons	GPIO	4	
2	DAC	Analog	1	8 bits DAC
3	Flash	SPI	3	Can read 8KB per sec
…..				

	Functions	Speed	RAM	Special Hardware Requirement (FPU, etc)
1	SPI read	8KByte/s	8 KByte	
2				
3				
…..				

Requirements vs Specification

Translating from requirements to specification (from the consumer's language to the designer's)

- Capturing a consistent set of requirements from the customer
- Massaging those requirements into a more formal specification

Requirements vs Specification

Consumers:

- are not embedded system designers
- see mostly users' interactions
- most of the time have unrealistic expectations as to what can be
- done within their budgets
- have a different language

System Specification

- Problems of unclear specifications
  - implementation of wrong functionality
  - system architecture may be inadequate to meet the needs of the implementation

1.1. System Specification

- Group discussion
  - Finish the System Specification for the digital clock

2. Embedded System Design Issues

- **Design issues** are problems that make it difficult to design an embedded system

1. **Constraint issues**
   - cost may matter more than speed
   - long life cycle
   - Reliability/safety
   - Low-power
   - Size / weight

Examples: Portable heart-beat monitor

- Long life cycle (10 years)
- Reliability (accuracy 99%)
- Low-power (5 using days)
- Light weight (<1kg)

2. Embedded System Design Issues

1. Constraints

Examples for smart home system

No.	Constraints	Note
1	Low price (< 1.000.000 dong)	Correct
2	Ability to detect smoke and fire	Wrong
3	Low power (100mW when idle, 3W when active)	Correct
4	Response time for control < 1ms	?
5	Support remote control by smartphones	?
6	Easy to install	?

Constraints are limitations or restrictions of some parameters of the system

2. Embedded System Design Issues

2. Functional issues

– safety-critical applications

– damage to life, health, economy

– affect to environment, society, politics

Example:

- Message LED for a shop
  - Display message for customers
  - Malfunctions result in small damage
- Message LED for a stock market
  - Display stock data
  - Malfunctions could damage to economy
- Battery charger
  - ?

2. Embedded System Design Issues

2. Functional issues

Examples for battery charger

No.	Issues	Note
1	Battery can be over-heat, it need to be detected by a sensor	Correct
2	Display a charging current and battery status	Wrong
3	The system must have a fuse for protection of over current	?
4	Support 3 charging modes	?
5	Apply efficient algorithm for fast charging and increase battery life cycle	?

Functional issues are problems which can affect to life, health, economy, environment, society, politics, ethics.

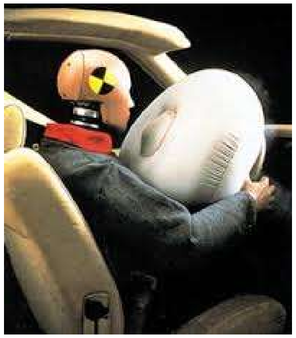
2. Embedded System Design Issues

3. Real-time issues

– Determine whether the system is hard/soft/non real-time
– Determine the time constraint (delay)

Example

- Door entry alarm
  – Non/soft real-time system: delay < 1-2s
- Video recorder
  – Soft real-time system: delay < 1ms
- Car airbag system
  – ?
- Weather temperature monitoring
  – ?




2. Embedded System Design Issues

4. Concurrent issues

– System and environment run concurrently

– multi-functions

– interface with other systems

– May need a scheduler to manage concurrent tasks

Examples: Weather temperature monitoring

Multi-functions:

– Read temperature values from the sensor

– Write data to memory

– Display data on LCD

2. Embedded System Design Issues

5. Reactive issues

- **Continuous / discontinuous interaction**
  - Power on demand
    - Turn ON when using
    - **Ex:** MP3 player, Tivi system
  - Always ON, once started run forever
    - Continuous interaction with their environment
    - Termination is a bad behavior => watchdog timer
    - **Ex:** Camera surveillance system, data acquisition system
- **Response to external periodic/non-periodic events**
  - Events are periodic: the system needs a scheduler to capture the events
  - Events are non-periodic: the system needs to estimate miss event cases

2. Embedded System Design Issues

- Group discussion
  - Discuss about design issues of your own class project

4. Project plan

- Build a team

- Build a plan

Build a team

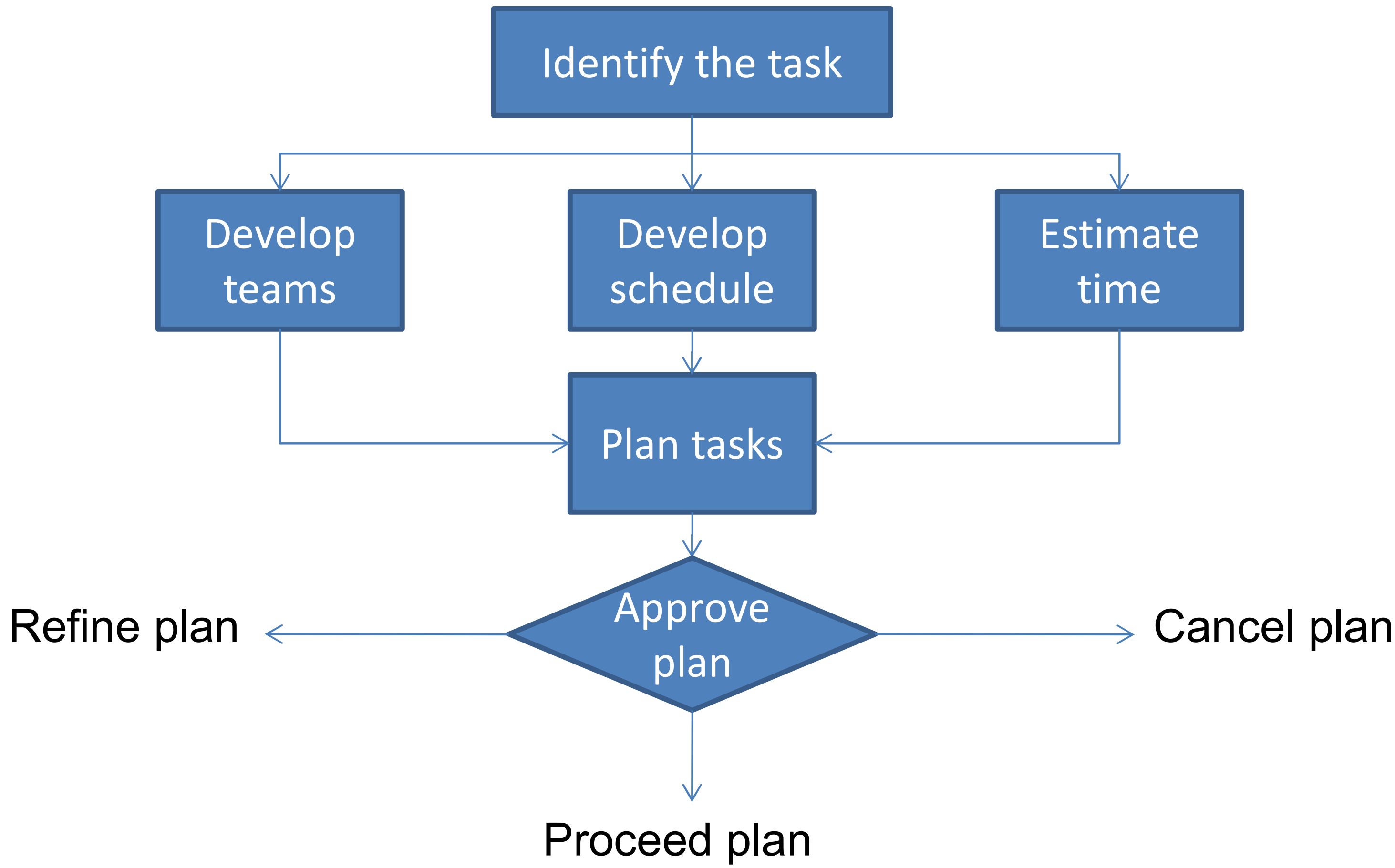
- Important points about teams
  - Teams bring together complementary skills and experiences
  - Teams establish communication to support real-time problem solving
  - Teams develop decisions by consensus rather than by authority

Team Contract

TEAM CONTRACT		
Team name: BK1		Date: 26 Aug. 2014
Team member	**Roles**	**Signature**
Nguyễn Văn A	Leader, system engineer	
Trần Văn B	Hardware design	
Nguyễn Thị C	Software design	
Tasks		**Responsible member**
1. Develop system architecture		Nguyễn Văn A
2. Design hardware		Trần Văn B
3. Develop software		Nguyễn Thị C
4. Integrate and test		All
Team meeting		9AM, Wednesday, weekly
Team rules	1. Participate in all team meetings 2. Listen carefully to all comments at meetings 3. Complete all assigned tasks before deadlines 4. Focus on results rather than excuses after.	

Build a plan

- Project planning activities



Project plan example (1)

Project planning	
Team name	BK_DEE
Product name	Home security system
Main features	-Fire alarm -Door alarm -Send warning to home owner
Estimated Time	3 months (8 hours / a day) Start: 20 Aug. 2014 End: 20 Nov. 2014
Estimated Cost	Components : 300,000 VNĐ Tools : 100,000 VNĐ Materials : 100,000 VNĐ Total : 500,000 VNĐ
Team members	Student 1: leader Student 2: hardware design Student 3: software design
Schedule	

Project plan example (2)

Project planning			
Schedule	Month 1	Month2	Month 3
1. Design system architecture			
2. Design hardware part			
2.1. Design central control board			
2.2. Design interface			
2.3. Implement hardware board			
3. Develop software part			
3.1. Develop control algorithm			
3.2. Develop driver, user interface			
3.3. Implement software program			
4. Integrate and test			
4.1. Simulate operations			
4.2. Verify system			

Group discussion

1. Consider the project car’s door mechanism

   Write system specification for this project

Group discussion

2. Consider the project car door mechanism

   Write team contract and plan for the project