MODULE 2: Characteristics and Quality attributes of Embedded systems:

Structure

MODULE-2

2.1 Characteristics and quality attributes of embedded systems: Characteristics,

2.2 Operational and nonoperational quality attributes,

2.3 Application specific embedded system - washing machine,

2.4 Application specific embedded system domain specific – automotive

Objectives

Learning Objectives

1. Understand Characteristics of Embedded Systems – Real-time, application-specific, power-efficient.
2. Analyze Quality Attributes – Differentiate operational (response, reliability) and non-operational (testability, cost).
3. Explore Application-Specific Systems – Study washing machine automation.
4. Examine Domain-Specific Systems – Learn automotive embedded systems (ECUs, ABS, infotainment).

2.1 Characteristics and quality attributes of embedded systems: Characteristics,

2.1 Characteristics of Embedded Systems

1. Application & Domain Specific
   - Embedded systems are designed for a particular function or domain and cannot be repurposed.
   - Example: A washing machine control unit cannot be used for an air conditioner.
2. Reactive & Real-Time
   - Embedded systems interact with the real world using sensors and must react instantly to changes.
   - Real-Time Systems ensure deterministic responses, critical in applications like flight control and ABS (Anti-lock Braking Systems) in vehicles.
3. Operates in Harsh Environments
   - Some embedded systems function in extreme conditions (e.g., high temperatures, vibrations, dust).
   - Example: Industrial automation systems deployed in high-temperature zones.
4. Distributed Systems
   - Multiple embedded systems work together in a network to achieve a common function.
   - Example: ATM machines consist of independent embedded units (card reader, cash dispenser, printer) working together.
5. Small Size & Lightweight
   - Compact and aesthetically pleasing designs are important for usability and portability.
   - Example: Mobile phones, smartwatches, and medical devices.
6. Power Efficiency
   - Optimized for low power consumption to enhance battery life and reduce heat dissipation.
   - Example: Ultra-low power controllers and energy-efficient components in IoT devices.

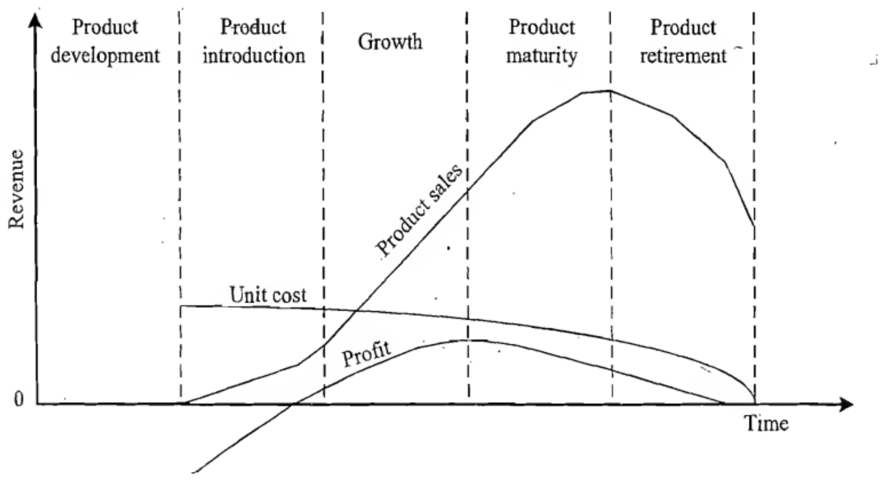
2.2 Quality Attributes of Embedded Systems

Operational Quality Attributes

1. Response
    - Defines how quickly an embedded system reacts to input changes.
    - Example: Flight control systems must respond instantly to pilot inputs.
2. Throughput
    - Measures system efficiency in terms of completed tasks per unit time.
    - Example: Card readers measure throughput as transactions per second.
3. Reliability
    - Indicates system stability and performance over long durations.
    - Measured using MTBF (Mean Time Between Failures) and MTTR (Mean Time to Repair).
    - Example: Medical monitoring systems must function reliably 24/7.
4. Maintainability
    - Deals with ease of repairing and updating the system.
    - Includes scheduled maintenance (e.g., printer ink replacement) and corrective maintenance (e.g., ATM repair after failure).
5. Security
    - Ensures data protection and prevents unauthorized access.
    - Example: Password-protected PDAs, encryption in IoT devices.
6. Safety
    - Prevents damage to users or the environment due to failures.
    - Example: Airbag systems, industrial robots with emergency stop mechanisms.

2.2.1 Non-Operational Quality Attributes

1. Testability & Debug-ability
   - Determines how easily the system can be tested for faults.
   - Includes hardware testing (e.g., checking sensors) and firmware debugging.
2. Evolvability
   - The ability of an embedded system to integrate new features or hardware over time.
   - Example: Firmware updates in smartphones.
3. Portability
   - The capability of running on multiple platforms with minimal modifications.
   - Example: C-based firmware can be recompiled for different processors.
4. Time-to-Market
   - The speed from product idea to launch is crucial in competitive markets.
   - Example: Fast prototyping using off-the-shelf components.
5. Cost & Revenue
   - Unit cost must be optimized to ensure profitability while maintaining quality.
   - Example: Smartphones are expensive at launch but become cheaper over time.

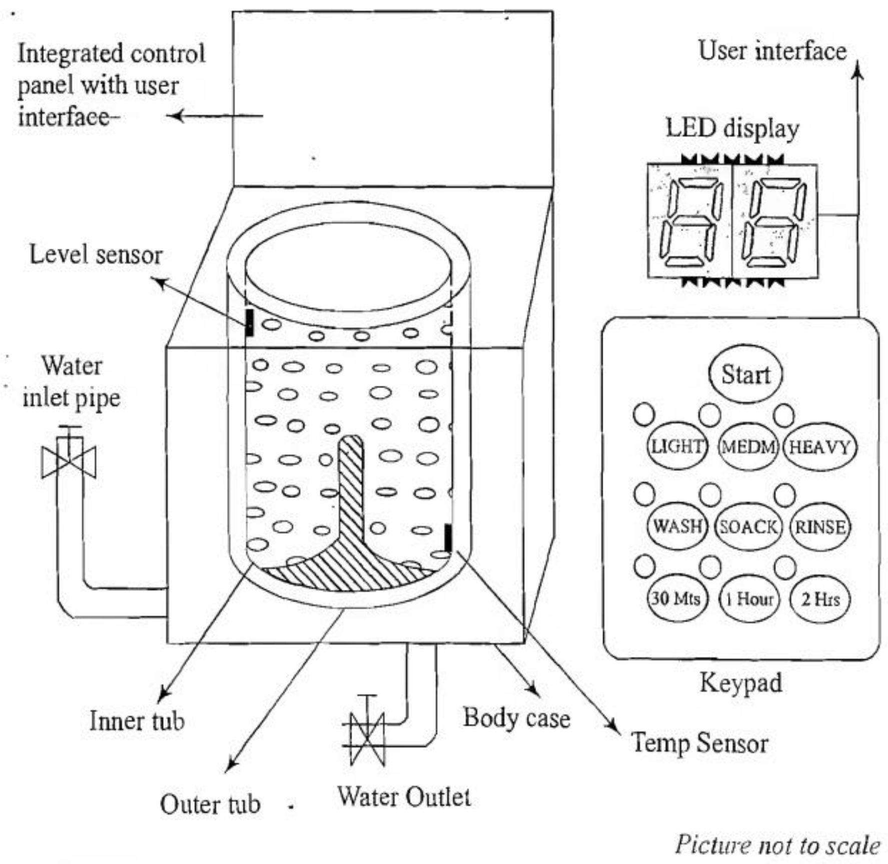


The product lifecycle follows a pattern where revenue grows from the introduction stage to maturity, then declines in the retirement phase.

- Unit cost is highest at launch (e.g., new smartphones are expensive initially but drop in price over time).

- Profit starts negative due to development costs, increases with sales, stabilizes, and then declines as sales drop.
- Returns exceed investment only after reaching a profitable stage.

2.3 application specific embedded system - washing machine,



Washing machine - Functional block diagram

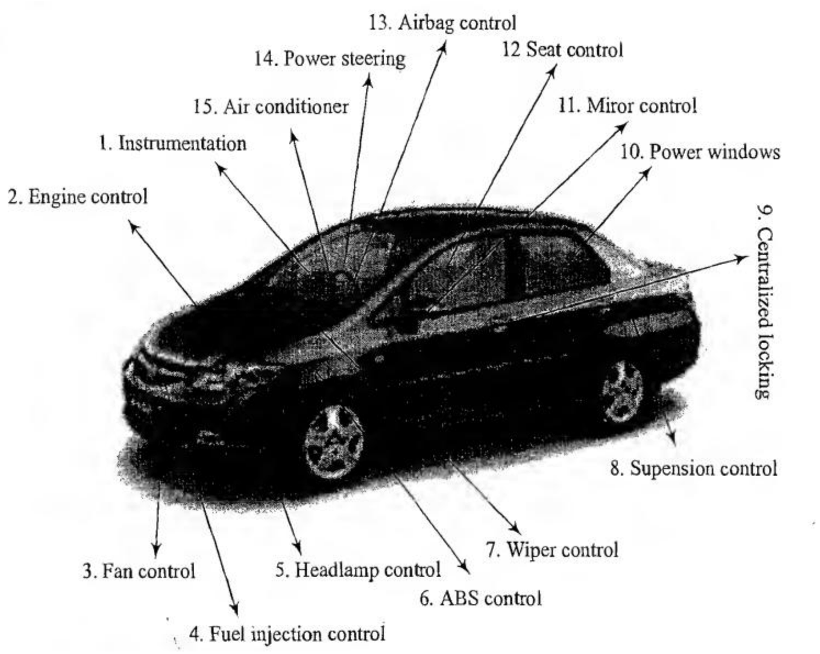
Washing Machine – Application-Specific Embedded System

1. Embedded System Role:
   - Uses sensors, actuators, control units, and user interfaces to automate washing.
   - Enhances home automation with intelligent processing.
2. Main Components:
   - Actuators: Motorized agitator, tumble tub, water pump, inlet valve.
   - Sensors: Water temperature sensor, water level sensor.

- Control Unit: Microcontroller processes sensor data and manages actuators.

3. Types of Washing Machines:
   - Top Loading: Agitator moves clothes up and down.
   - Front Loading: Clothes tumble repeatedly in water.
4. Washing Phases:
   - Wash: Agitator/tub moves clothes through water and detergent.
   - Spin: High-speed tub rotation removes excess water using centrifugal force.
   - Rinse: Removes detergent by adding fresh water and spinning.
5. User Interface & Controls:
   - Inputs: Wash type selector (Wash, Spin, Rinse), cloth type selector (Light, Medium, Heavy Duty), timer settings.
   - Outputs: LED/LCD display, status indicators.
6. Working Principle:
   - Sensors detect water levels & temperature → Microcontroller processes data → Actuators perform washing functions.
7. Manufacturer Variations:
   - Basic principles remain the same, but interface & control panel designs vary.

2.4 Application specific embedded system domain specific – automotive



Automotive Embedded Systems – Key Points

1. Overview

- Embedded systems control mechanical functions in vehicles.
- Found in simple (wipers, mirrors) to complex (ABS, airbags) systems.
- Built on microcontrollers, DSPs, or hybrid ECUs (Electronic Control Units).
- 20-40 ECUs in standard cars, 75-100 in luxury vehicles.

2. Types of ECUs (Electronic Control Units)

- High-Speed ECUs (HECUs): Require fast response for engine control, ABS, fuel injection, electronic throttle, transmission control.
- Low-Speed ECUs (LECUs): Used for non-critical functions like audio controls, door locks, mirrors, wipers, seat adjustments.

3. Automotive Communication Buses

- CAN (Controller Area Network): High-speed (1 Mbps), used for safety & powertrain systems (ABS, airbags, engine control).
- LIN (Local Interconnect Network): Low-speed (20 Kbps), used for sensor-actuator interfacing (mirrors, fans, seats, windows).
- MOST (Media-Oriented System Transport): Fiber-optic bus for automotive multimedia systems (audio, video, infotainment).

4. Automotive Embedded Market Key Players

(a) Silicon Providers

- Analog Devices: Signal processing chips for GPS, driver assistance systems.
- Xilinx: FPGAs & CPLDs for collision avoidance, voice recognition.
- Atmel: Flash microcontrollers for car infotainment, safety, and CAN/LIN networking.
- Maxim/Dallas, NXP, Renesas, Texas Instruments, Infineon, NEC – Leading chip providers for automotive control systems.

(b) Tools & Platform Providers

- Enea OSE: Real-time OS for multi-core, fault-tolerant systems.
- MathWorks MATLAB/SIMULINK: Used for simulation & modeling.
- Keil Microvision, Lauterbach, ARTiSAN, Microsoft Windows CE – Provide debugging & development tools for automotive applications.

(c) Solution Providers (OEMs)

- Bosch Automotive: Full automotive solutions (engine control, safety, infotainment).
- DENSO: Provides engine management, hybrid vehicle solutions.
- Delphi, Infosys: Supply hardware & software automotive solutions.

Outcomes

At the end of the module, students will be able to:
CO-2: Acquire knowledge about basic concepts of circuit emulators, debugging and RTOS [L2]

TEXT BOOKS:

Shibu K V, “Introduction to Embedded Systems”, Second Edition, McGraw Hill Education

Reference Books/ Link

NPTL Lectures: https://nptel.ac.in/courses/108102045
Embedded Systems, IIT Delhi, Prof. Santanu Chaudhary