Dr.K.Gopi
Department of ECE

Introduction to Embedded Systems

Syllabus:

Embedded System-Definition, History, Classification, application areas and purpose of embedded systems, Block diagram of embedded system, Quality attributes of an Embedded systems, Application-specific and Domain-Specific examples of an embedded system.

What is Embedded System?

Definition:

An Electronic/Electro mechanical system which is designed to perform a specific function and is a combination of both hardware and firmware (Software).

E.g. Electronic Toys, Mobile Handsets, Washing Machines, Air Conditioners, Automotive Control Units, Set Top Box, DVD Player etc.

(OR)

An Embedded system is a Combination of Hardware and Software designed for specific function.

Differences between Embedded systems and General Purpose computing systems:

Embedded Systems Vs General Computing Systems:

General Purpose Computing System	Embedded System
A system which is a combination of generic hardware and General Purpose Operating System for executing a variety of applications	A system which is a combination of special purpose hardware and embedded OS for executing a specific set of applications
Contain a General Purpose Operating System (GPOS)	May or may not contain an operating system for functioning
Applications are alterable (programmable) by user (It is possible for the end user to re-install the Operating System, and add or remove user applications)	The firmware of the embedded system is pre-programmed and it is non-alterable by end-user
Performance is the key deciding factor on the selection of the system. Always „Faster is Better"	Application specific requirements (like performance, power requirements, memory usage etc) are the key deciding factors
Less/not at all tailored towards reduced operating power requirements, options for different levels of power management.	Highly tailored to take advantage of the power saving modes supported by hardware and Operating System
Response requirements are not time critical	For certain category of embedded systems like mission critical systems, the response time requirement is highly critical
Need not be deterministic in execution behavior	Execution behavior is deterministic for certain type of embedded systems like „Hard Real Time" systems

History of Embedded Systems:

First Recognized Modern Embedded System: Apollo Guidance Computer (AGC) developed by Charles Stark Draper at the MIT Instrumentation Laboratory.

- It has two modules
- 1.Command module(CM) 2.Lunar Excursion module(LEM)
- RAM size 256 , 1K ,2K words
- ROM size 4K,10K,36K words
- Clock frequency is 1.024MHz
- 5000 ,3-input RTL NOR gates are used
- User interface is DSKY(display/Keyboard)

The command module was designed to encircle the moon while the Lunar Module and its crew were designed to go down to the moon surface and land there safely. **The Apollo Guidance computer was developed for the Lunar Expedition (Journey taken by the people to do the research on Moon)**

Classification of Embedded systems

It is possible to have Multitude classifications for embedded systems, based on the different criteria.

The classifications are based on

1. Based on Generation
2. Complexity and performance requirements
3. Based on Deterministic behavior
4. Based on triggering.

1. Based on Generation:

First Generation: The early embedded systems built around 8-bit microprocessors like 8085 and Z80 and 4-bit microcontrollers

EX. stepper motor control units, Digital Telephone Keypads etc.

Second Generation: Embedded Systems built around 16-bit microprocessors and 8 or 16-bit microcontrollers, following the first generation embedded systems

EX. Data Acquisition Systems etc.

Third Generation: Embedded Systems built around high performance 16/32 bit Microprocessors/controllers, Application Specific Instruction set processors like Digital Signal Processors (DSPs), and Application Specific Integrated Circuits (ASICs).The instruction set is complex and powerful.

EX. Robotics, industrial process control, networking etc.

2. Classification based on Complexity & Performance :

➢ **Small Scale systems:** The embedded systems built around low performance and low cost 8 or 16 bit microprocessors/ microcontrollers. It is suitable for simple applications and where performance is not time critical. It may or may not contain OS.

➢ **Medium Scale systems:** Embedded Systems built around medium performance, low cost 16 or 32 bit microprocessors / microcontrollers or DSPs. These are slightly complex in hardware and firmware. It may contain GPOS/RTOS.

➢ **Large Scale/Complex systems:** Embedded Systems built around high performance 32 or 64 bit RISC processors/controllers, or multi-core processors. It requires complex hardware and software. These system may contain multiple processors/controllers and co-units/hardware accelerators for offloading the processing requirements from the main processor. It contains RTOS for scheduling, prioritization and management

3. Classification Based on deterministic behavior:

It is applicable for Real Time systems. The application/task execution behavior for an embedded system can be either deterministic or non-deterministic These are classified in to two types

1. **Soft Real time Systems:** Missing a deadline may not be critical and can be tolerated to a certain degree

2. **Hard Real time systems:** Missing a program/task execution time deadline can have catastrophic consequences (financial, human loss of life, etc.)

4. Classification Based on Triggering: These are classified into two types

1. **Event Triggered:** Activities within the system (e.g., task run-times) are dynamic and depend upon occurrence of different events.

2. **Time triggered:** Activities within the system follow a statically computed schedule (i.e., they are allocated time slots during which they can take place) and thus by nature are predictable.

Application areas of Embedded systems

Major Application Areas of Embedded Systems:

- **Consumer Electronics:** Camcorders, Cameras etc.
- **Household Appliances:** Television, DVD players, washing machine, Fridge, Microwave Oven etc.
- **Home Automation and Security Systems:** Air conditioners, sprinklers, Intruder detection alarms, Closed Circuit Television Cameras, Fire alarms etc.
- **Automotive Industry:** Anti-lock breaking systems (ABS), Engine Control, Ignition Systems, Automatic Navigation Systems etc.
- **Telecom**: Cellular Telephones, Telephone switches, Handset Multimedia Applications etc.
- **Computer Peripherals:** Printers, Scanners, Fax machines etc.
- **Computer Networking Systems:** Network Routers, Switches, Hubs, Firewalls etc.
- **Health Care:** Different Kinds of Scanners, EEG, ECG Machines etc.
- **Measurement & Instrumentation:** Digital multi meters, Digital CROs, Logic Analyzers PLC systems etc.
- **Banking & Retail:** Automatic Teller Machines (ATM) and Currency counters, Point of Sales (POS)
- **Card Readers:** Barcode, Smart Card Readers, Hand held Devices etc.

Purpose of Embedded Systems

Each Embedded Systems is designed to serve the purpose of any one or a combination of the following tasks.

1. Data Collection/Storage/Representation
2. Data Communication
3. Data (Signal) Processing
4. Data Monitoring
5. Data Control
6. Application Specific User Interface

1. Data Collection/Storage/Representation:-

- Performs acquisition of data from the external world.
- The collected data can be either analog or digital
- Data collection is usually done for storage, analysis, manipulation and transmission
- The collected data may be stored directly in the system or may be transmitted to some other systems or it may be processed by the system or it may be deleted instantly after giving a meaningful representation

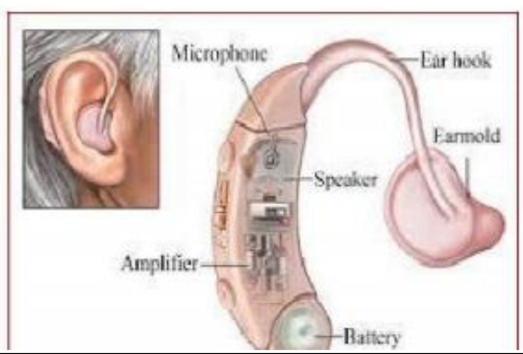
2. Data Communication:-

- Embedded Data communication systems are deployed in applications ranging from complex satellite communication systems to simple home networking systems
- Embedded Data communication systems are dedicated for data communication
- The data communication can happen through a wired interface (like Ethernet, RS-232C/USB/IEEE1394 etc) or wireless interface (like Wi-Fi, GSM,/GPRS, Bluetooth, ZigBee etc)

- Network hubs, Routers, switches, Modems etc are typical examples for dedicated data transmission embedded systems

3. Data (Signal) Processing:-

- Embedded systems with Signal processing functionalities are employed in applications demanding signal processing like Speech coding, synthesis, audio video codec, transmission applications etc
- Computational intensive systems
- Employs Digital Signal Processors (DSPs)



4. Monitoring:-

- Embedded systems coming under this category are specifically designed for monitoring purpose
- They are used for determining the state of some variables using input sensors
- They cannot impose control over variables.
- Electro Cardiogram (ECG) machine for monitoring the heart beat of a patient is a typical example for this
- The sensors used in ECG are the different Electrodes connected to the patient"s body
- Measuring instruments like Digital CRO, Digital Multi meter, Logic Analyzer etc used in Control & Instrumentation applications are also examples of embedded systems for monitoring purpose

5. Control:-

- Embedded systems with control functionalities are used for imposing control over some variables according to the changes in input variables
- Embedded system with control functionality contains both sensors and actuators
- Sensors are connected to the input port for capturing the changes in environmental variable or measuring variable
- The actuators connected to the output port are controlled according to the changes in input variable to put an impact on the controlling variable to bring the controlled variable to the specified range

Air conditioner for controlling room temperature is a typical example for embedded system with „Control" functionality

Air conditioner contains a room temperature sensing element (sensor) which may be a thermistor and a handheld unit for setting up (feeding) the desired temperature

6. **Application Specific User Interface:-**

Embedded systems which are designed for a specific application Contains Application Specific User interface (rather than general standard UI) like key board, Display units etc Aimed at a specific target group of users Mobile handsets, Control units in industrial applications etc are examples.

Dr.K.Gopi
Department of ECE

Characteristics of Embedded systems

Embedded systems possess certain specific characteristics and these are unique to each Embedded system.

1. Application and domain specific

2. Reactive and Real Time

3. Operates in harsh environments

4. Distributed

5. Small Size and weight

6. Power concerns

1. **Application and Domain Specific:-**

- Each E.S has certain functions to perform and they are developed in such a manner to do the intended functions only.
- They cannot be used for any other purpose.
- Ex – The embedded control units of the microwave oven cannot be replaced with AC"S embedded control unit because the embedded control units of microwave oven and AC are specifically designed to perform certain specific tasks.

2. Reactive and Real Time:-

- Embedded systems are in constant interaction with the real world through sensors and user-defined input devices which are connected to the input port of the system.
- Any changes in the real world are captured by the sensors or input devices in real time and the control algorithm running inside the unit reacts in a designed manner to bring the controlled output variables to the desired level.
- E.S produce changes in output in response to the changes in the input, so they are referred as reactive systems.
- Real Time system operation means the timing behavior of the system should be deterministic ie the system should respond to requests in a known amount of time.
- Example – E.S which are mission critical like flight control systems, Antilock Brake Systems (ABS) etc are Real Time systems.

3. **Operates in Harsh Environment:–**

- The design of E.S should take care of the operating conditions of the area where the system is going to implement.

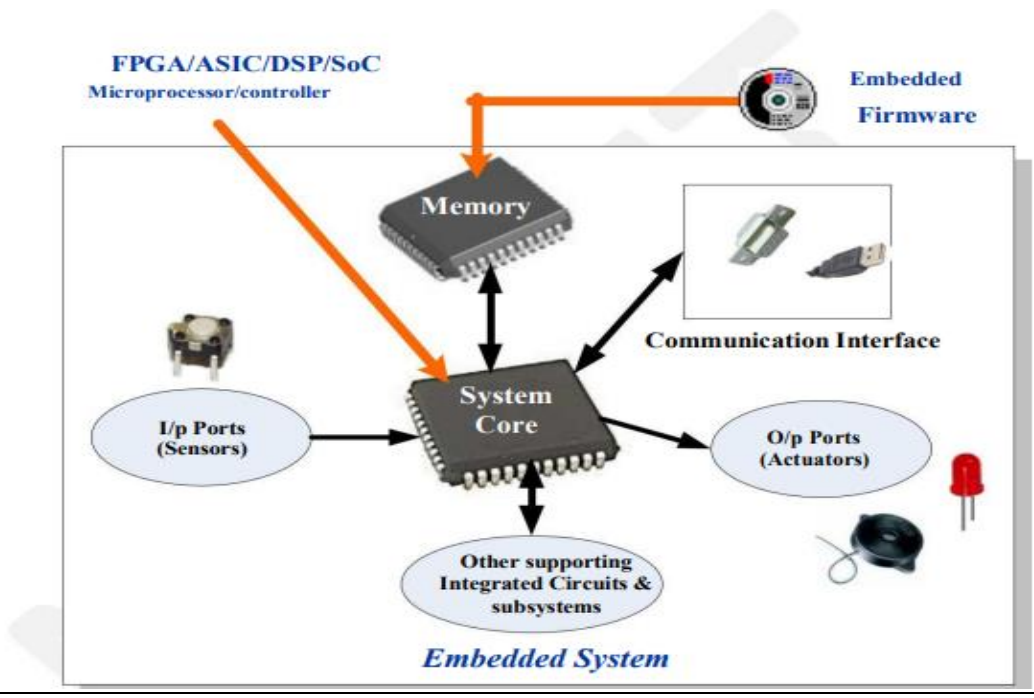
- Ex – If the system needs to be deployed in a high temperature zone, then all the components used in the system should be of high temperature grade.
- Also proper shock absorption techniques should be provided to systems which are going to be commissioned in places subject to high shock.

4. **Distributed: –**
   - It means that embedded systems may be a part of a larger system.
   - Many numbers of such distributed embedded systems form a single large embedded control unit.
   - Ex – Automatic vending machine. It contains a card reader, a vending unit etc. Each of them are independent embedded units but they work together to perform the overall vending function.
5. **Small Size and Weight:-**
   - Product aesthetics (size, weight, shape, style, etc) is an important factor in choosing a product.
   - It is convenient to handle a compact device than a bulky product.

BLOCK DIAGRAM OF EMBEDDED SYSTEMS

ELEMENTS OF EMBEDDED SYSTEMS:

An embedded system is a combination of 3 things, Hardware Software Mechanical Components and it is supposed to do one specific task only. A typical embedded system contains a single chip controller which acts as the master brain of the system. Diagrammatically an embedded system can be represented as follows:



Embedded systems are basically designed to regulate a physical variable (such Microwave Oven) or to manipulate the state of some devices by sending some signals to the actuators or devices connected to the output port system (such as temperature in Air Conditioner), in response to the input signal provided by the end users or sensors which are connected to the input ports. Hence the embedded systems can be viewed as a reactive system.

The control is achieved by processing the information coming from the sensors and user interfaces and controlling some actuators that regulate the physical variable. Keyboards, push button, switches, etc. are Examples of common user interface input devices and LEDs, LCDs, Piezoelectric buzzers, etc examples for common user interface output devices for a typical embedded system.

The memory of the system is responsible for holding the code (control algorithm and other important configuration details). There are two types of memories are used in any embedded system. Fixed memory (ROM) is used for storing code or program. The user cannot change the firmware in this type of memory. The most common types of memories

The Core of the Embedded Systems: The core of the embedded system falls into any one of the following categories.

- **Microprocessors and Microcontrollers**
- **Digital Signal Processors**
- **Programmable Logic Devices (PLDs)**
- **Application Specific Integrated Circuits (ASICs)**
- **Commercial off the shelf Components (COTS)**

Microprocessor and Microcontrollers:

- Almost 80% of the embedded systems are processor/ controller based. The processor may be microprocessor or a microcontroller or digital signal processor, depending on the domain and application.
- Microprocessor: A silicon chip representing a Central Processing Unit (CPU), which is capable of performing arithmetic as well as logical operations according to a pre-defined set of Instructions, which is specific to the manufacturer In general the CPU contains the Arithmetic and Logic Unit (ALU), Control Unit and Working registers Microprocessor is a dependant unit and it requires the combination of other hardware like Memory, Timer Unit, and Interrupt Controller etc for proper functioning.

Microcontroller:

- A highly integrated silicon chip containing a CPU, scratch pad RAM, Special and General purpose Register Arrays, On Chip ROM/FLASH memory for program storage, Timer and Interrupt control units and dedicated I/O ports .

- Microcontrollers can be considered as a super set of Microprocessors ϖ Microcontroller can be general purpose (like Intel 8051, designed for generic applications and domains) or application specific (Like Automotive AVR from Atmel Corporation).

Microprocessor Vs Microcontroller:

Microprocessor	Microcontroller
A silicon chip representing a Central Processing Unit (CPU), which is capable of performing arithmetic as well as logical operations according to a pre-defined set of Instructions	A microcontroller is a highly integrated chip that contains a CPU, scratch pad RAM, Special and General purpose Register Arrays, On Chip ROM/FLASH memory for program storage, Timer and Interrupt control units and dedicated I/O ports
It is a dependent unit. It requires the combination of other chips like Timers, Program and data memory chips, Interrupt controllers etc for functioning	It is a self contained unit and it doesn't require external Interrupt Controller, Timer, UART etc for its functioning
Most of the time general purpose in design and operation	Mostly application oriented or domain specific
Doesn"t contain a built in I/O port. The I/O Port functionality needs to be implemented with the help of external Programmable Peripheral Interface Chips like 8255	Most of the processors contain multiple built-in I/O ports which can be operated as a single 8 or 16 or 32 bit Port or as individual port pins
Targeted for high end market where performance is important	Targeted for embedded market where performance is not so critical (At present this demarcation is invalid)
Limited power saving options compared to microcontrollers	Includes lot of power saving features

<u>Digital Signal Processors (DSPs):</u>

- Powerful special purpose 8/16/32 bit microprocessors designed specifically to meet the computational demands and power constraints of today's embedded audio, video, and communications applications Digital Signal Processors are 2 to 3 times faster than the general purpose microprocessors in signal processing applications
- DSPs implement algorithms in hardware which speeds up the execution whereas general purpose processors implement the algorithm in firmware and the speed of execution depends primarily on the clock for the processors
- DSP can be viewed as a microchip designed for performing high speed computational operations for „addition‟, „subtraction‟, „multiplication‟ and „division‟
- A typical Digital Signal Processor incorporates the following key units ϖ Program Memory Data Memoryϖ Computational Engineϖ I/O Unitϖ Audio video signal

processing, telecommunication and multimedia applications are typical examples where DSP is employed

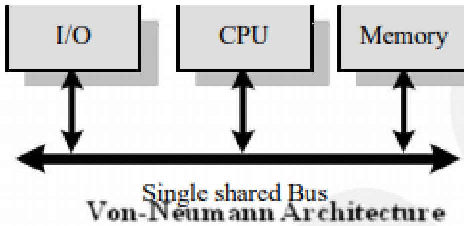
RISC V/s CISC Processors/Controllers:

RISC	CISC
Lesser no. of instructions	Greater no. of Instructions
Instruction Pipelining and increased execution speed	Generally no instruction pipelining feature
Orthogonal Instruction Set (Allows each instruction to operate on any register and use any addressing mode)	Non Orthogonal Instruction Set (All instructions are not allowed to operate on any register and use any addressing mode. It is instruction specific)
Operations are performed on registers only, the only memory operations are load and store	Operations are performed on registers or memory depending on the instruction
Large number of registers are available	Limited no. of general purpose registers
Programmer needs to write more code to execute a task since the instructions are simpler ones	. A programmer can achieve the desired functionality with a single instruction which in turn provides the effect of using more simpler single instructions in RISC
Single, Fixed length Instructions	Variable length Instructions
Less Silicon usage and pin count	More silicon usage since more additional decoder logic is required to implement the complex instruction decoding.
With Harvard Architecture	Can be Harvard or Von-Neumann Architecture

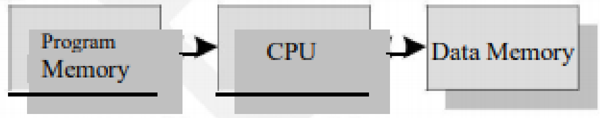
Harvard V/s Von-Neumann Processor/Controller Architecture

The terms Harvard and Von-Neumann refers to the processor architecture design. Microprocessors/controllers based on the common bus for fetching both instructions stored in a common main memory Von-Neumann architecture shares a single and data. Program instructions and data are Microprocessors/controllers based on the Harvard architecture will have separate data bus and instruction bus. This allows the data transfer and program fetching to occur simultaneously on both buses With Harvard architecture the data memory can be read and written while the program memory is being accessed.

Dr.K.Gopi
Department of ECE



Von-Neumann Architecture



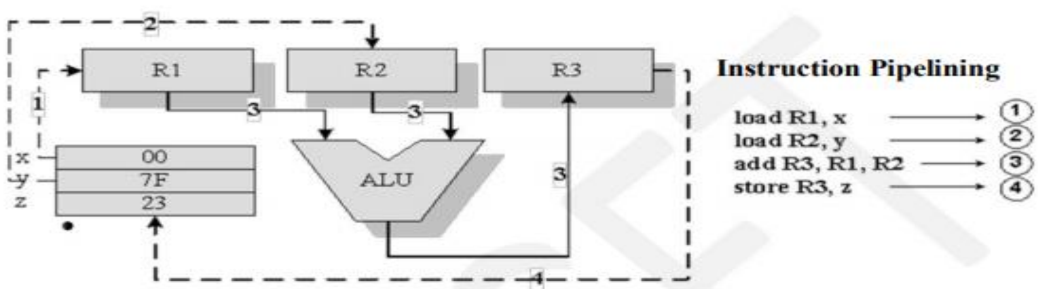
Harvard Architecture

Harvard V/s Von-Neumann Processor/Controller Architecture:

Harvard Architecture	Von-Neumann Architecture
Separate buses for Instruction and Data fetching	Single shared bus for Instruction and Data fetching
Easier to Pipeline, so high performance can be achieved	Low performance Compared to Harvard Architecture
Comparatively high cost	Cheaper
No memory alignment problems	Allows self modifying codes†
Since data memory and program memory are stored physically in different locations, no chances for accidental corruption of program memory	Since data memory and program memory are stored physically in same chip, chances for accidental corruption of program memory

Load Store Operation & Instruction Pipelining:

The RISC processor instruction set is orthogonal and it operates on registers. The memory access related operations are performed by the special instructions *load* and *store*. If the operand is specified as memory location, the content of it is loaded to a register using the *load* instruction. The instruction *store* stores data from a specified register to a specified memory location



Load Store Operation

- The conventional instruction execution by the processor follows the fetch-decode-execute sequence
- The „fetch" part fetches the instruction from program memory or code memory and the decode part decodes the instruction to generate the necessary control signals

Application Specific Integrated Circuit (ASIC):

A microchip designed to perform a specific or unique application. It is used as replacement to conventional general purpose logic chips. ASIC integrates several functions into a single chip and thereby reduces the system development cost Most of the ASICs are proprietary products.

As a single chip, ASIC consumes very small area in the total system and thereby helps in the design of smaller systems with high capabilities/functionalities. ASICs can be pre-fabricated for a special application or it can be custom fabricated by using the components from a re-usable „building block" library of components for a particular customer application.

Commercial off the Shelf Component (COTS): A Commercial off-the-shelf (COTS) product is one which is used „as-is" COTS products are designed in such a way to provide easy integration and interoperability with existing system component.

Typical examples for the COTS hardware unit are Remote Controlled Toy Car control unit including the RF Circuitry part, High performance, high frequency microwave electronics (2 to 200 GHz), High bandwidth analog-to-digital converters, Devices and components for operation at very high temperatures, Electro-optic IR imaging arrays, UV/IR Detectors etc.

The major advantage of using COTS is that they are readily available in the market, cheap and a developer can cut down his/her development time to a great extend. There is no need to design the module yourself and write the firmware. Everything will be readily supplied by the COTs manufacturer.

QUALITY ATTRIBUTES OF EMBEDDED SYSTEMS

Quality attributes are the non-functional requirements that need to be documented properly in any system design.

Quality attributes can be classified as

I. Operational quality attributes
II. Non-operational quality attributes.

I. **Operational Quality Attributes**: The operational quality attributes represent the relevant quality attributes related to the embedded system when it is in the operational mode or online mode. Operational Quality Attributes are:

1. **Response :-** It is the measure of quickness of the system. It tells how fast the system is tracking the changes in input variables. Most of the E.S demands fast response which should be almost realtime. Ex – Flight control application

2. **Throughput :-** It deals with the efficiency of a system. It can be defined as the rate of production or operation of a defined process over a stated period of time. The rates can be expressed in terms of products, batches produced or any other meaningful measurements.

Ex – In case of card reader throughput means how many transactions the reader can perform in a minute or in an hour or in a day. Throughput is generally measured in terms of "Benchmark". A Benchmark is a reference point by which something can be measured

3. **Reliability :-**

It is a measure of how much we can rely upon the proper functioning of the system.

• Mean Time Between Failure (MTBF) and Mean Time To Repair (MTTR) are the terms used in determining system reliability.

• MTBF gives the frequency of failures in hours/weeks/months.

• MTTR specifies how long the system is allowed to be out of order following a failure.

For embedded system with critical application need, it should be of the order of minutes.

4. **Maintainability:-**

• It deals with support and maintenance to the end user or client in case of technical issues and product failure or on the basis of a routine system checkup.

• Reliability and maintainability are complementary to each other.

• A more reliable system means a system with less corrective maintainability requirements and vice versa. • Maintainability can be broadly classified into two categories

1. Scheduled or Periodic maintenance (Preventive maintenance)

2. Corrective maintenance to unexpected failures.

5. **Security:-**

Confidentiality, Integrity and availability are the three major measures of information security.

• Confidentiality deals with protection of data and application from unauthorized disclosure.

• Integrity deals with the protection of data and application from unauthorized modification.

• Availability deals with protection of data and application from unauthorized users.

6. **Safety :-** Safety deals with the possible damages that can happen to the operator, public and the environment due to the breakdown of an Embedded System. The breakdown of an embedded system may occur due to a hardware failure or a firmware failure. Safety analysis is a must in product engineering to evaluate the anticipated damages and determine the best course of action to bring down the consequences of damage to an acceptable level.

II. Non-Operational Quality Attributes: The quality attributes that needs to be addressed for the product not on the basis of operational aspects are grouped under this category.

1. **Testability and Debug-ability:-**
   • Testability deals with how easily one can test the design, application and by which means it can be done. For an E.S testability is applicable to both the embedded hardware and firmware.

   • Embedded hardware testing ensures that the peripherals and total hardware functions in the desired manner, whereas firmware testing ensures that the firmware is functioning in the expected way.

   Debug-ability is a means of debugging the product from unexpected behavior in the system

• Debug-ability is two level process

1. Hardware level
2. software level

- Hardware level: It is used for finding the issues created by hardware problems.
- Software level: It is employed for finding the errors created by the flaws in the software.

2. **Evolvability :-** It is a term which is closely related to Biology. It is referred as the non-heritable variation. For an embedded system evolvability refers to the ease with which the embedded product can be modified to take advantage of new firmware or hardware technologies.

3. **Portability:-** It is the measure of system independence. An embedded product is said to be portable if the product is capable of functioning in various environments, target processors and

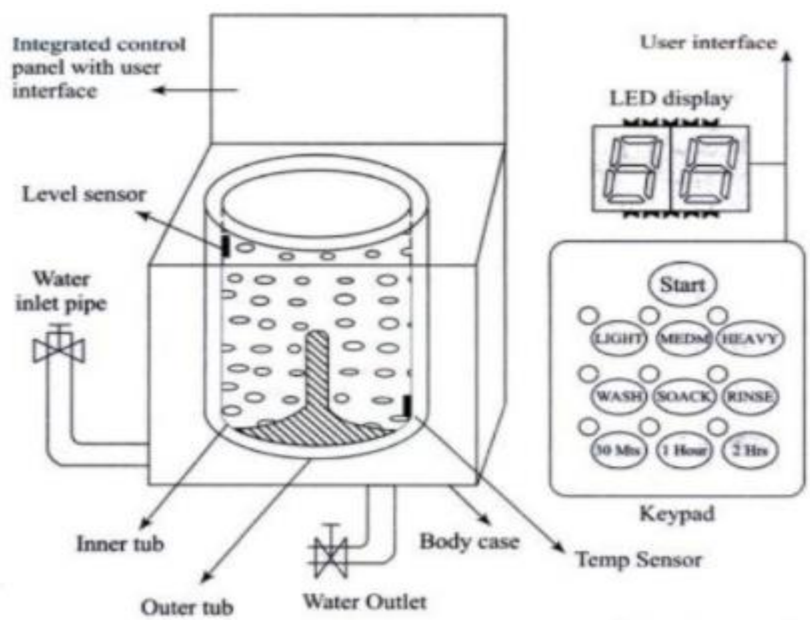
embedded operating systems. „Porting‟ represents the migration of embedded firmware written for one target processor to a different target processor.

4. **Time-to-Prototype and Market:-** It is the time elapsed between the conceptualization of a product and the time at which the product is ready for selling. The commercial embedded product market is highly competitive and time to market the product is critical factor in the success of commercial embedded product. There may be multiple players in embedded industry who develop products of the same category (like mobile phone).

5. **Per Unit Cost and Revenue:-** Cost is a factor which is closely monitored by both end user and product manufacturer. Cost is highly sensitive factor for commercial products •Any failure to position the cost of a commercial product at a nominal rate may lead to the failure of the product in the market. Proper market study and cost benefit analysis should be carried out before taking a decision on the per-unit cost of the embedded product.

Application specific Embedded system

Washing machine is a typical example of embedded system which provides extensive support in home automation applications. The embedded system contains sensors, actuators, control unit and application specific user interfaces like Keyboards, display units, etc. you can see all the components in the washing machine where some of them are visible and some them are invisible to you.



The actuator part of washing machine consists of motorized agitator, water drawing pump and inlet value to control the flow of water in to the unit. The sensor part consists of the water temperature sensor, level sensor etc.

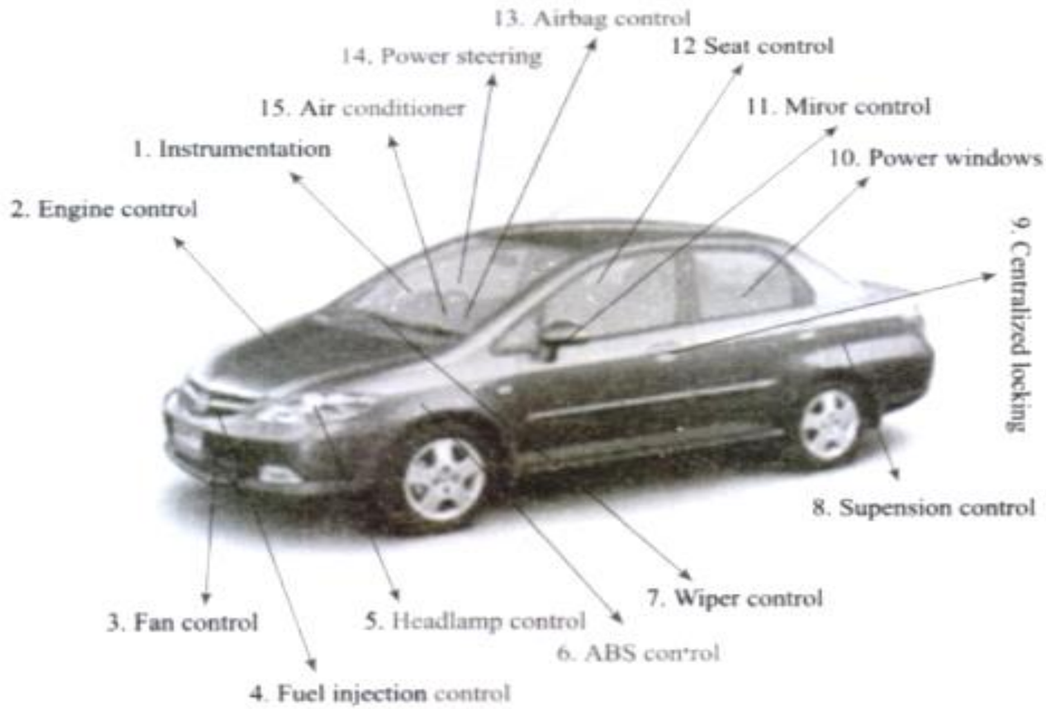
The control part contains a microprocessor/ controller based with interfaces to the sensors and actuators. The sensor data is fed back to the control unit and the control unit generates the necessary actuator outputs. The control unit also provides connectivity to user interfaces like keypad for setting the washing time, selecting the type of material to be washed like light, medium, heavy etc. The led and display unit act as the user output devices.

Input interface includes the keyboard which consists of wash type selector namely Wash, Spin and Rinse, Cloth type selector namely Light Medium and heavy etc.

Domain Specific Embedded system

The major applications domain of embedded systems is industrial, consumer automotive and telecom etc. out of which the telecom and automotive holds the big market. Automotive embedded systems are the one where electronics take control over the mechanical systems. The presence of automotive embedded system in a vehicle varies from simple mirror and wiper controls to complex air bag controller and antilock brake systems (ABS). Automotive embedded systems are normally built around microcontrollers or DSPs or hybrid of the two and are generally known as Electronic Control Units (ECUs). The number of embedded controllers in ordinary vehicles varies from 20 to 40. Whereas luxury vehicles like Mercedes and MBW 7 may contain 75 to 100 numbers of embedded controllers.

The various types of electronic control units used in the automotive embedded industry can be broadly classified in to High speed control units and low speed embedded control units.



High speed control units: These units deployed in critical control units requiring fast response. They include fuel injection systems, antilock brake system, engine control system, steering control, transmission control units and central control unit.

Low speed control units: Low speed control unit are deployed in applications where response time is not so critical. They generally are built around low cost, microprocessor/microcontrollers and digital signal processors. Audio controllers, passenger and driver door locks, wiper control, mirror control and head lamp control unit etc, are the examples of low speed control units.