2.1 Characteristics of an Embedded System

- **Unlike** general purpose computing systems, embedded systems **possess certain specific characteristics** and these characteristics are unique to each embedded system.

- Some of **the important characteristics** of an embedded system are:

  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

- An embedded system **is designed** for **a specific purpose only**.
- It **will not do** any other task.
- Ex. Air conditioner's embedded control unit, it cannot replace microwave oven…
- Ex. A washing machine can only wash, it cannot cook..
- Because the embedded control units of microwave oven and air conditioner **are specifically designed** to perform certain specific tasks.
- Certain embedded systems are **specific to a domain**: ex. **A hearing aid** is an application that belongs to **the domain of signal processing** and **telecom** with another control unit designed to serve another domain like consumer electronics.

2. Reactive and Real Time

- Certain embedded systems **are designed to react to the events** that occur in the nearby environment. These events also **occur real-time**.
- Ex. Flight control systems, Antilock Brake Systems (ABS), etc. are examples of **Real Time systems**
- Ex. An air conditioner adjusts its mechanical parts **as soon as** it **gets a signal from its sensors** **to increase or decrease the temperature** when the user operates it using a remote control.
- An embedded system uses **Sensors** **to take inputs** and has **actuators** **to bring out the required functionality**.

3. Operation in Harsh Environment

- Certain embedded systems **are designed to operate in harsh environments** like a dusty one or a high temperature zone or an area subject to vibrations and shock or very high temperature of the deserts or very low temperature of the mountains or extreme rains.

- These embedded systems have to be **capable of sustaining the environmental conditions** it is designed to operate in.

4. Distributed

- The term distributed **means** that embedded systems **may be a part of a larger system.**
- These components **are independent** of each other **but** have **to work together for the larger system** to function properly.
- Ex. **An automatic vending machine** is a typical example for this. The vending machine contains **a card reader** (for pre-paid vending systems), **a vending unit, etc**. Each of them are **independent** embedded units but they **work together to perform the overall vending function.**
- Ex. **Automatic Teller Machine (ATM)** contains **a card reader embedded unit**, responsible for reading and validating the user's ATM card, **transaction unit** for performing transactions, **a currency counter** for dispatching/vending currency to the authorized person and **a printer unit** for printing the transaction details.
- This **can visualize** these as **independent embedded systems**. **But** they **work together to achieve a common goal**.

5. Small Size and Weight

- An embedded system that is **compact in size** and has **light weight** will be desirable or more popular than one that is bulky and heavy.
- Ex. Currently available **cell phones**. The cell phones that have the **maximum features** are popular but also their size and weight is an important characteristic.

6. Power Concerns

- It is desirable that the **power utilization and heat dissipation** of any embedded system be **low**.
- If **more heat** is dissipated then additional units **like heat sinks or cooling fans** need to be added to the circuit.
- Ex. **The production of high amount of heat** **demands** **cooling requirements like cooling fans** which in turn occupies additional space and make the system bulky. Nowadays ultra low power components are available in the market.
- Select the design according to the **low power components** like **low dropout regulators**, and **controllers/processors** **with power saving modes.**
- Also **power management** is a **critical** constraint in battery operated application.
- **The more the power consumption the less is the battery life.**

2.2 Quality Attributes of Embedded Systems

- Quality attributes are the **non-functional requirements** that need to be documented properly in any system design.
- If the quality attributes are more concrete and measurable, it will give a **positive impact** on the system development process and the end product.
- The various quality attributes that needs to be addressed in any embedded system development are broadly **classified into two**, namely

  i. Operational Quality Attributes

  ii. Non-Operational Quality Attributes

1. 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. The important quality attributes coming under **this category are listed** below:

    i. Response

    ii. Throughput

    iii. Reliability

    iv. Maintainability

    v. Security

    vi. Safety

i. Response

- **Response** is a **measure of quickness** of the system.
- It gives you an idea about **how fast your system** is **tracking the input variables.**
- Most of the embedded system demand **fast response** which should be **real-time**.
- Ex. An embedded system deployed in **flight control application** **should respond in a Real Time manner**.
- Any **response delay** in the system **will create potential damages** to the **safety of the flight as well as the passengers**.
- It is not necessary that all embedded systems should be Real Time in response.
- For example, the response time requirement for **an electronic toy** is not at all time-critical.

ii. Throughput

- **Throughput** **<u>deals with</u>** **the efficiency** of system.
- It can be **<u>defined</u>** as **rate of production or process** of a defined process over a stated period of time.
- The rates **<u>can be expressed</u>** in terms of **units of products**, **batches produced**, or **any other meaningful measurements**.
- In case of **card reader** like the ones used **in buses**, throughput means **how much transactions** the Reader can perform **in a minute or hour or day.**
- Throughput is generally measured in terms of **‘Benchmark’.** A ‘Benchmark’ is a reference point by which something can be measured.
- Benchmark can be a set of **performance criteria** that a product is expected to meet or **a standard product** that **can be used for comparing other products** of the same product line.

iii. Reliability

- Reliability is **a measure of how much percentage** you rely upon the proper functioning of the system or what is the **% susceptibility** of the system to failure.
- **Mean Time Between Failures (MTBF)** and **Mean Time To Repair (MTTR)** are the terms used in defining 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 an embedded system with critical application need, it **should be of the order of minutes.**

iv. Maintainability

- **Maintainability** deals with **support and maintenance** to the end user or client in case of **technical issues** and **product failures** or on the basis of **a routine system checkup**.
- Reliability and maintainability are considered as **two complementary disciplines**. A **more reliable system** means a system with **less corrective maintainability requirements** and vice versa.
- Maintainability can be classified into **two types**:

1. **Scheduled or Periodic Maintenance (Preventive Maintenance)**

An inkjet printer uses ink cartridges, which are consumable components and as per the printer manufacturer the end use **should replace** the cartridge **after** each ‘n’ number of printouts to get quality prints.

2. **Maintenance to Unexpected Failures (Corrective Maintenance)**

If the paper feeding part of the printer fails the printer fails to print and it requires **immediate repairs** to rectify this problem.

Hence it is obvious that maintainability is simply **an indication of the availability** of the product for use. In any embedded system design, **the ideal value for availability** is expressed as

$$\boxed{A_i = MTBF/(MTBF + MTTR)}$$

Where A_i=Availability in the ideal condition, MTBF=Mean Time Between Failures, and MTTR= Mean Time To Repair

v. Security

- **'Confidentially', 'Integrity', and 'Availability'** are three major measures of information security.
- **'Confidentially'** deals with the 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.**
- Certain embedded systems have to make sure they conform to **the security measures.**
- Ex. An electronic safety Deposit Locker can be used only with a pin number like a password.

vi. Safety

- **Safety** deals with **the possible damages** that can happen to **the operators**, public and the environment **<u>due to</u>** **the breakdown** of an embedded system or **<u>due to</u>** **the emission of radioactive or hazardous materials** from the embedded products.

- The breakdown of an embedded system may occur **<u>due to</u>** **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 the damages to an acceptable level**.

2. Non Operational Attributes

- The quality attributes that needs to be addressed for the product **'not' on the basic of operational aspects** are grouped under this category. The important quality attributes coming under this category are listed below:

    i. Testability & Debug-ability

    ii. Evolvability

    iii. Portability

    iv. Time to prototype and market

    v. Per unit and total cost

i. Testability & Debug-ability

- **Testability** deals with **how easily one can test** his/her design, application and by which means he/she can test it.
- For an embedded product, **testability is applicable** to **both** the embedded **hardware and firmware**.
- **Debug-ability** is a means of **debugging the product** as such for figuring out the probable sources that create **unexpected behavior** in the total system.
- Debug-ability has **two aspects** in the embedded system development context, namely, **hardware level debugging** and **firmware level debugging**.
- Hardware debugging is used for figuring out the issues created by hardware problems whereas firmware debugging is employed to figure out the probable errors that appear as a result of flaws in the firmware.

ii. Evolvability

- **Evolvability** is a term which is closely related to Biology.
- Evolvability is referred as the **non-heritable variation**.
- For an embedded system, the quality attribute 'Evolvability' refers to the ease with which the embedded product (including firmware and hardware) **can be modified** to take advantage of **new firmware or hardware technologies.**

iii. Portability

- **Portability** is a measure of **'system independence'.**
- An embedded product can be called portable **if** it is **capable of functioning** in various environments, target processors/controllers and embedded operating systems.
- A standard embedded product should always be **flexible and portable**.

iv. Time-to-Prototype and Market

- Time-to-market is **the time elapsed** between the conceptualization of a product and the time at which **the product is ready for selling** (for commercial product) or use (for non-commercial products).
- **The commercial embedded product marke**t is **highly competitive** and time to market the product is **a critical factor** in the success of a commercial embedded product.
- **Product prototyping** helps a lot in **reducing** time-to-market.

v. Per Unit Cost and Revenue

- **Cost** is a factor which is **closely monitored** by **both** **end user** (those who buy the product) and **product manufacturer** (those who build the product).
- Cost is **a highly sensitive factor** for commercial products.
- **Proper market study and cost benefit analysis** should be carried out **before taking decision** on the per unit cost of the embedded product.
- When the product is **introduced** in the market, for **the initial period** the sales and **revenue** will be **low**.
- There **won't be much competition** when the product sales and revenue increase.
- During the **maturing phase**, the **growth** will be **steady** and **revenue** reaches **highest point** and at **retirement time** there will be **a drop in sales volume.**