Characteristics of Embedded Systems

The characteristics of embedded system are different from those of a general purpose computer and so are its Quality metrics. Some of the characteristics of an embedded system that make it different from a general purpose computer:

1. Application and Domain specific

An embedded system is designed for a specific purpose only. It will not do any other task.
Example: A washing machine can only wash, it cannot cook.

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.
Example: 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.

3. Operation in harsh environment

Certain embedded systems are designed to operate in harsh environments like very high temperature of the deserts or very low temperature of the mountains or extreme rains.

4. Distributed

Certain embedded systems are part of a larger system and thus form components of a distributed system.These components are independent of each other but have to work together for the larger system to function properly.
Example: A car has many embedded systems controlled to its dash board. Each one is an independent embedded system yet the entire car can be said to function properly only if all the systems work together.

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.

Example: The heat seeking system used in some missiles needs to be small and lightweight so as to fit in well and allow high speed maneuvers.

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.If more power is required then a battery of higher power or more batteries need to be accommodated in the embedded system.

Design/Quality Metrics and Constraints

There are two types of quality attributes:
Operational Quality Attributes: These are attributes related to operation or functioning of an embedded system. The way an embedded system operates affects its overall quality.

1. Response

Response is a measure of quickness of the system. Gives an idea about how fast a system is tracking the input variables.Most of the embedded system demand fast response which should be real-time.A real-time system must react to stimuli from the controlled object (or the operator) within the time interval dictated by the environment.For real-time systems, right answers arriving too late are wrong. A real-time constraint is called hard, if not meeting that constraint could result in a catastrophe. All other time-constraints are called soft.

2. Throughput

Deals with the efficiency of system.Defined as rate of production or process of a defined process over a stated period of time.In case of card reader like the ones used in buses, throughput means how much transaction the reader can perform in a minute or hour or day.

3. Reliability

Measures how much percentage we can rely upon the proper functioning of the system. Mean Time between failures and Mean Time to Repair are terms used in defining system reliability. Mean Time between failures can be defined as the average time the system is

functioning before a failure occurs. Mean time to repair can be defined as the average time the system has spent in repairs.

4. Maintainability:

Deals with support and maintenance to the end user or a client in case of technical issues and product failures or on the basis of a routine system checkup.

It can be classified into two types:

a) Scheduled or Periodic Maintenance: This is the maintenance that is required regularly after a periodic time interval e.g. refilling of printer cartridges.

b) Maintenance to unexpected failure: Involves the maintenance due to a sudden breakdown in the functioning of the system e.g. printer not taking paper in spite of a full paper stack.

5. Security

Confidentiality, Integrity and Availability are three corner stones of information security.

Confidentiality deals with protection data from unauthorized disclosure.

Integrity gives protection from unauthorized modification.

Availability gives protection from unauthorized user

Certain Embedded systems have to make sure they conform to the security measures e.g. an electronic safety deposit Locker can be used only with a pin number like a password.

6. Safety

Deals with the possible damage that can happen to the operating person and environment due to the breakdown of an embedded system or due to the emission of hazardous materials from the embedded products.

A safety analysis is a must in product engineering to evaluate the anticipated damage and determine the best course of action to bring down the consequence of damages to an acceptable level.

Non-Operational Quality Attributes

These are attributes not related to operation or functioning of an embedded system. These are the attributes that are associated with the embedded system before it can be put in operation.

1. Testability and Debug-ability

Deals with how easily one can test his/her design, application and by which mean he/she can test it.

Hardware: testing the peripherals and total hardware function in designed manner.

Firmware testing.

Debug-ability: means of debugging the product as such for figuring out the probable sources that create unexpected behavior in the total system.

2. Evolvability

Refers to the ease with which an embedded product can be modified to take advantage of new firmware or hardware technology.

3. Portability

Measure of system Independence.

An embedded product can be called portable if it is capable of performing its operation as it is intended to do in various environments irrespective of different processor and or controller and embedded operating systems.

4. Time to prototype and market

Time elapsed.

Product prototyping helps in reducing time to market. In order to shorten the time to prototype, make use of all possible option like use of reuse, off the self component etc.

5. Per unit and total cost

Proper market study and cost benefit analysis should be carried out before taking decision on the per-unit cost of the embedded product.

Design challenge – Optimizing the design metrics

The above metrics typically compete with one another: improving one often leads to degradation in another. For example, if we reduce an implementation's size, its performance may suffer. To best meet this optimization challenge, the designer must be comfortable with a variety

of hardware and software implementation technologies, and must be able to migrate from one technology to another, in order to find the best implementation for a given application and constraints. Thus, a designer cannot simply be a hardware expert or a software expert, as is commonly the case today; the designer must be an expert in both areas.