SNS COLLEGE OF TECHNOLOGY

An Autonomous Institution

Coimbatore-35

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A++' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

19ITT204 - MICROCONTROLLER AND EMBEDDED SYSTEMS

II YEAR/ IV SEMESTER

UNIT III EMBEDDED SYSTEM CONCEPTS AND PROCESSORS

TOPIC – Characteristics and Quality Attributes of Embedded Systems

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**

OUTLINE

Characteristics of an Embedded system

Application and Domain Specific

➢If you closely observe any embedded system, you will find that each embedded system is having 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. It is the major criterion which distinguishes an embedded system from a general purpose system. For example, **you cannot replace the embedded control unit of your microwave oven with your air conditioner's embedded control unit**, because the embedded control units of microwave oven and airconditioner are specifically designed to perform certain specific tasks.

➢Also you cannot replace an embedded control unit developed for a particular domain say telecom with another control unit designed to serve another domain like consumer electronics.

Characteristics of an Embedded system

Reactive and Real Time

- As mentioned earlier, 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 happening in the Real world (which is called an Event)** 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.
- The event may be a periodic one or an unpredicted one. If the event is an unpredicted one then such systems should be designed in such a way that it should be scheduled to capture the events without missing them.
- Embedded systems produce changes in output in response to the changes in the input. So they are generally referred as **Reactive Systems.**

Characteristics of an Embedded system

- **Real Time System** operation means the timing behaviour of the system should be deterministic; meaning the system should respond to requests or tasks in a known amount of time.
- A Real Time system should **not miss any deadlines for tasks or operations.**
- It is not necessary that all embedded systems should be Real Time in operations. Embedded applications or **systems which are mission critical**, like flight control systems, Antilock Brake Systems (ABS), etc. are examples of Real Time systems.
- The design of an embedded Real time system should take the **worst case scenario into consideration**

Characteristics of an Embedded system

Operates in Harsh Environment

- It is not necessary that all embedded systems should be deployed in **controlled environments**. The environment in which the embedded system deployed may be a dusty one or a high temperature zone or an area subject to vibrations and shock.
- Systems placed in such areas should be capable to withstand all these **adverse operating conditions**. The design should take care of the operating conditions of the area where the system is going to implement.
- For example, 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**. Here we cannot go for a compromise in cost.
- Also proper **shock absorption techniques** should be provided to systems which are going to be commissioned **in places subject to high shock.**
- **Power supply fluctuations, corrosion and component aging, etc**. are the other factors that need to be taken into consideration for embedded systems to work in harsh environments.

Characteristics of an Embedded system

Distributed

➢The term distributed means that **embedded systems may be a part of larger systems**. Many numbers of such distributed embedded systems form a single large embedded control unit.

➢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.

➢Another example is the **Automatic Teller Machine (ATM)**. An 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. We can visualize these as independent embedded systems. But they work together to achieve a common goal.

➢Another typical example of a distributed embedded system is the **Supervisory Control And Data Acquisition (SCADA) system** used in Control & Instrumentation applications, which contains physically distributed individual embedded control units connected to a supervisory module

Characteristics of an Embedded system

Small Size and Weight

- **Product aesthetics** is an important factor in choosing a product. For example, when you plan to buy a new mobile phone, you may make a comparative study on the pros and corns of the products available in the market.

- Definitely the product aesthetics (size, weight, shape, style, etc.) will be one of the deciding factors to choose a product. People believe in the phrase "**Small is beautiful**".

- Moreover it is **convenient to handle** a compact device than a bulky product. In embedded domain also compactness is a significant deciding factor. **Most of the application demands small sized and low weight products.**

Characteristics of an Embedded system

Power Concerns

- Power management is another important factor that needs to be considered in designing embedded systems.
- Embedded systems should be designed in such a way as to **minimise the heat dissipation** by the system. 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**.

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 **'Operational Quality Attributes' and 'Non-Operational Quality Attributes'**.

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:

1. **Response**
2. **Throughput**
3. **Reliability**
4. **Maintainability**
5. **Security**
6. **Safety**

1. Response

- Response is a **measure of quickness of the system**. It gives you an idea about how fast your system is tracking the changes in input variables. Most of the embedded systems demand fast response which should be almost Real Time.

- For example, 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**. There is no specific deadline that this system should respond within this particular timeline.

2. Throughput

- Throughput deals with the efficiency of a system.

- In general 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 units of products, batches produced, or any other meaningful measurements.

- In the case of a 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.

- 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.

3. Reliability

➢Reliability is a **measure of how much you can rely upon the proper functioning of the system** or **what is the % susceptibility of the system to failures**.

➢**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 specifics 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.

4. 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.

➢As the reliability of the system increases, the chances of failure and non-functioning also reduces, thereby the need for maintainability is also reduced.

➢Maintainability is closely related to the **system availability**. Maintainability can be broadly classified into two categories, namely,
'Scheduled or Periodic Maintenance (preventive maintenance) and Maintenance to unexpected failures (corrective maintenance)'.

➢Some embedded products may use consumable components or may contain components which are subject to wear and tear and they should be replaced on a periodic basis. The period may be based on the total hours of the system usage or the total output the system delivered.

➢A printer is a typical example for illustrating the two types of maintainability. An inkjet printer uses ink cartridges, which are consumable components and as per the printer manufacturer the end user should replace the cartridge after each 'n' number of printouts to get quality prints. This is an **example for Scheduled or Periodic maintenance'**.

5. Security

- **Confidentiality, 'Integrity', and 'Availability** (The term 'Availability' mentioned here is not related to the term 'Availability' mentioned under the Maintainability' section) are the three major measures of information security..
- **Confidentiality** deals with the protection of data and application from unauthorised disclosure.
- **Integrity** deals with the protection of data and application from unauthorised modification.
- **Availability** deals with protection of data and application from unauthorized users.
- A very good example of the Security' aspect in an embedded product is a Personal Digital Assistant (PDA). The PDA can be either a shared resource (e.g. PDAs used in LAB setups) or an individual one.
- If it is a shared one there should be some mechanism in the form of a user name and password to access into a particular person's profile-This is an example of **'Availability'.**
- Also all data and applications present in the PDA need not be accessible to all users. Some of them are specifically accessible to administrators only. For achieving this, Administrator and user levels of security should be implemented -An example of **Confidentiality.**
- Some data present in the PDA may be visible to all users but there may not be necessary permissions to alter the data by the users. That is Read Only access is allocated to all users-An example of **Integrity.**

6. Safety

- 'Safety and Security are two confusing terms. Sometimes you may feel both of them as a single attribute. But they represent two unique aspects in quality attributes.

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

- 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 the damages to an acceptable level.

- As stated before, some of the safety threats are **sudden** (like product breakdown) and some of them are **gradual** (like hazardous emissions from the product).

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

1. **Testability & Debug-ability**
2. **Evolvability**
3. **Portability**
4. **Time to prototype and market**
5. **Per unit and total cost.**

1. 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.**
- Embedded hardware testing ensures that the **peripherals and the total hardware functions in the desired manner,** whereas firmware testing ensures that the **fimmware is functioning in the expected way.**

➢**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.

2. 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.**

3. Portability

- **Portability** is a measure of **system independence**. An embedded product is said to be portable if the product is capable of functioning as such in various environments, target processors /controllers and embedded operating systems..
- The ease with which an embedded product can be ported on to a new platform is a direct measure of the re-work required.
- A standard embedded product should always be flexible and portable. In embedded products, the term **'porting'** represents the **migration of the embedded firmware written for one target processor (e.g. Intel x86) to a different target processor (say Hitachi SH3 processor).**
- If the **firmware is written in a high level language like 'C'** with little target processor-specific functions (operating system extensions or compiler specific utilities), it is very easy to port the firmware for the new processor by replacing those 'target processor-specific functions with the ones for the new target processor and re-compiling the program for the new target processor specific settings.
- Re-compiling the program for the new target processor generates the new target processor-specific machine codes.

4. 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 market is highly competitive and time to market the product is a critical factor in the success of a commercial embedded product..

➢There may be multiple players in the embedded industry who develop products of the same category (like mobile phone, portable media players, etc.). If you come up with a new design and if it takes long time to develop and market it, the competitor product may take advantage of it with their product.

➢Also, embedded technology is one where rapid technology change is happening. If you start your design by making use of a new technology and if it takes long time to develop and market the product, by the time you market the product, the technology might have superseded with a new technology.

➢**Product prototyping** helps a lot in reducing time-to-market. Whenever you have a product idea, you may not be certain about the feasibility of the idea. Prototyping is an informal kind of rapid product development in which the important features of the product under consideration are developed technology.

5. 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. 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.
- From a designer/product development company perspective the ultimate aim of a product is to generate marginal profit.
- So the budget and total system cost should be properly balanced to provide a marginal profit.

THANK YOU