Embedded systems

unit-1:- Introduction to embedded system: Definition of embedded system, embedded system vs general computing system, history of embedded systems, classification of embedded system, major applications area of embedded system, purpose of embedded system, processor and os trends in embedded system.

Embedded hardware units and devices in a system, embedded software in a system and an overview of programming languages, skills required for an embedded system designer, examples of the embedded system.

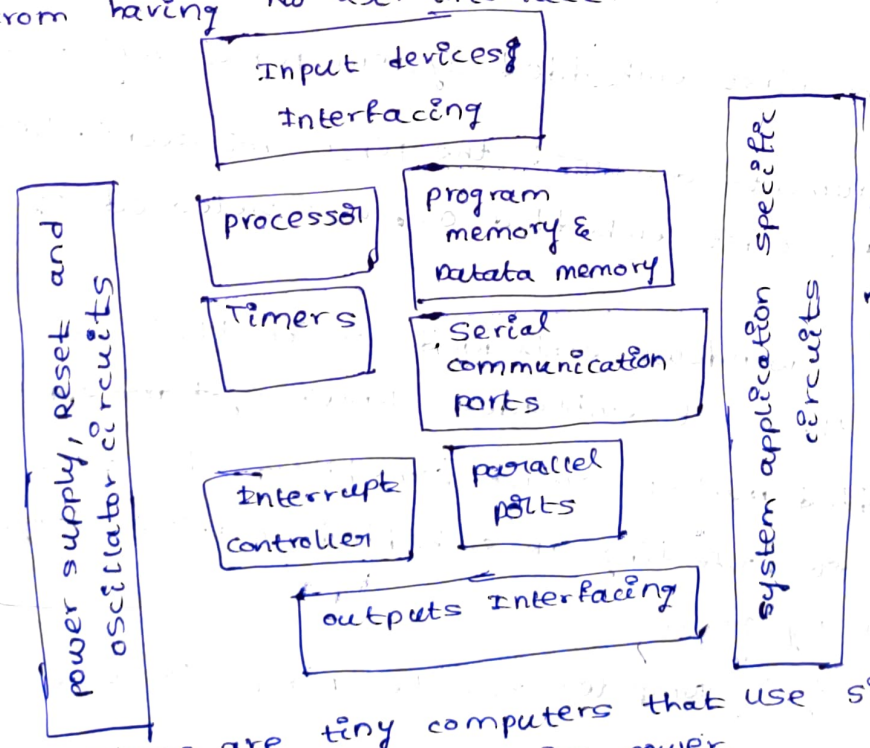
Definition of embedded system:-

An embedded system is a combination of computer hardware and software designed for a specific specific functions. embedded system may also function within a large systems. The systems can be programmable (or) fixed functionality.

* It is mostly designed for a specific functions with a larger system.

Ex:- Fire alarm is a common example of an embedded system which can sense only smoke

* while embedded systems are computing systems, they can range from having No user interface.



* some embedded system use a single microprocessor to acheire their aim, while others utilise multiple core processors

* embedded systems are tiny computers that use similar small amounts of memory and processing power.

Embedded system vs General computing system:-

General purpose computing system	Embedded system
1) multipurpose	1) single functioned
2) variety of applications	2) specific set of applications
3) faster and better	3) fixed Runtime Required
4) applications are altered by user	4) applications are not altered by user
5) contains a general purpose OS (GPOS)	5) May or may not contain an operating system for functioning
6) size of system is big	6) size of system is small
7) power consume is more	7) power consume is less
8) cost of system is costly.	8) cost of system is very less
9) memory is high	9) memory is low.
~~10) It contains~~	

History of embedded system:-

* In 1960, embedded system was first used for developing Apollo guidance system by charles stark developer draper at MIT
* In 1968, the first embedded system for a vehicle was released.
* Texas Instruments developed the first microcontroller in 1971
* In 1987, the first embedded OS, was released by wind River.
* Microsoft's windows embedded CE in 1996.
* In 1990s, the first embedded linux system appeared
* The embedded market reach $140 billion in 2013
* Analysts are projecting an embedded market larger than $40 bilion by 2030.
* The apollo flight computer was the first to use Integrated circuits.

* by the mid-1980s, widespread use of embedded systems became feasible with microcontroller.
* the first mass-produced embedded system was Autonetics D-17 guidance computer.

classification of embedded systems:-

embedded systems can be classified into different types based on different criteria;

(a) classification of based on generation
(b) classification based on complexity and performance
(c) classification based on requirements.
(d) classification based on Triggering
(e) classification based on deterministic behaviour

(a) classification based on generation:-

1) First generation (1G):-

* Bit around 8-bit microprocessors like Intel 8085, zilog z80
* simple in hardware circuit

ex:- stepper meter control units,
digital telephone keypads.

2) second generation (2G):-

* Bit around 16-bit microprocessors and 8-bit/16-bit microcon-
* Instruction set for processors/controllers are power than f generation
* Ex:- Intel 8086 processor, 8051 microcontroller.

Ex:- Data acquisition system (DAS)
supervisory control & data acquisition system (SCADA)

3) Third generation (3G):-

* Bit around 32-bit microprocessors & 16-bit/32-bit microcont
* Instruction set of processors became more complex and powe than 2G.
* concept of instruction pipelining is also evolved.
* Ex:- pentium, ARM, TMS320.

Ex:- Robotics, media, Networking.

4) Fourth generation (4G):-

* bit around 64-bit microprocessors & 32-bit microcontrollers
* high performance, tight Integration, very powerful.
* high performance (RTOS)

Eg:- smart phones, mobile Internet devices, high speed Network routers.

(b) classification based on complexity & performance :-

1) small-scale embedded system :-

* Embedded systems which are simple in application needs
* The performance requirements are Not time-critical.
* bit around low performance & low cost 8-bit or 16-bit

Ex :- stepper motor control
washing machine

2) medium-scale :-

* They are slightly complex in hardware & firmware requirements
* built around medium performance 16 or 32-microcontrollers.
* usually contain embedded operating system for functioning.

ex :- video games,
ATm Machines.

3) large-scale :-

* highly complex hardware and firmware
* bit around 32 or 64 RISC -bit
* they may contain multiple processors/controllers
* contains high performance Real time operating system
* performance requirements are Not time-critical.

Ex :- control systems
Flight control systems
Nuclear plant safety system

(c) classification based on Functional requirements :-

1) Stand-alone :-

* all the name implies, stand-alone system work in stand-alone mode.
* they take inputs, process them and produce outputs.
* The Inputs can be electrical signals from Transducers/ comm-ands &
* The outputs can be electrical signals to drive another system, an LED display/LCD display.

Eg :- digital camera
Air conditioner.

2) Real-time systems:-

* Real-time embedded systems is strictly specified time specific which means these embedded systems provides output in a particular time interval.
* embedded systems in which some specific task has to be done in a specified speriod are called Real-time systems.
* they have to work against deadlines.
* A Real time system should Not miss any deadline
* the response time requirement is highly critical for real time systems.

(e) classification based on deterministic behaviour:-

Soft Real time embedded systems:-

* In these types of embedded systems time/dead line is Not so strictly Followed. If deadline of the task is passed still result is accepted.

Hard Real-time:-

* In these types of embedded systems time/deadline of task is strictly Followed. task must be completed in time otherwise, result may Not be accepted.

Ex:- Traffic control system.
Medical usage in health sector.

3) ~~stand~~ Networked:-

Network embedded systems are connected to a Network which may be wire or wireless to provide output to the attached devices.

Ex:- ATM Machine
card swipe Machine.

4) Mobile:-

* mobile embedded systems are small & easy to use & require less resources. they are the most preferred embedded systems

Ex:- mp3 player
mobile phones
digital camera.

(d) classification based on Triggering :-

The embedded systems are classified into two types based on the Triggering of the systems

1) time-Triggered
2) event-Triggered

* If the system is activated or Triggered based on the pre-defined task or present time, then such a system is said to be the time Triggered embedded system

Ex:- automatic street light control system based on RTC i/p

* if the system is activated or Triggered based on some activity like change in temperature, change in pressure is said to be an event triggered embedded system

Ex:- automatic street light control system based on light intensity.

purpose of embedded systems:- (pdf).

The embedded systems are used in various domains like consumer electronics, home automation, telecommunications, computer Networking system, automative Industry, healthcare, Instrumentation, retail and banking applications etc.

* According to the application, they may have different Functionalities.
* Each embedded system is designed to serve the purpose of any one or combination of the following tasks.

* Data collection
* Data communication
* Data processing
* Monitoring.
* control
* application specific user Interface.

Data collection:-

* Embedded systems designed for the purpose of data collection performs of data from the external world
* Data collection is usually done for storage, analysis, manipulation & transmission
* the data refers all kinds of Information such as text, voice, Image, video, electrical signals. etc.
* The ~~data~~ collected data may be
  * stored directly in the system or
  * Transmitted to some other system or

proceed by the system or
displayed for giving a meaningful representation

Ex:- a digital camera

Data communication:-

* embedded data communication systems are deployed in applications ranging from complex satelite communication system to simple home networking systems.
* the data collecting embedded terminal itself can incorporate data communications units like wireless modules or wire-line modules.
* the Transmission is achieved by a wire-line or wire-less medium.
* certain embedded system acts as dedicated transmission units blw sending & receiving terminals.

Ex:- wireless Network router.

Data (Signal) Processing:

- The data (voice, image, video, electrical signals and other measurable quantities) collected by embedded systems may be used for various kinds of data processing.
- Embedded systems with signal processing functionalities are employed in applications demanding signal processing like speech coding, synthesis, audio video codec, transmission applications, etc.

Ex: A digital hearing aid is a typical example of an embedded system employing data processing.

Monitoring:

- These systems are specifically designed for monitoring purpose.
- Almost all embedded products coming under the medical domain are with monitoring functions only.
- They are used for determining the state of some variables using input sensors.

Ex: ECG machine for monitoring the heartbeat of a patient
Digital oscilloscope, Digital multi-meters, Logic analysers

Control

- Embedded systems are basically designed to regulate a physical variable (or) to manipulate the state of some devices by sending some signals to the actuators or devices connected to the output ports of the system, in response to the input signals provided by the end users or sensors which are connected to the input ports.

- An Embedded system with control functionality contains both sensors and actuators.
- **Sensors** are used to sense/detect the changes in the input variables and convert into electrical signals for any measurements. The sensors are connected to the input port.
- **Actuators** are used to converts electrical signals into corresponding physical action.
- The actuators are connected at the output port.

Ex: Air conditioner system - contains temperature sensing element (sensor) which may be a thermistor and a handheld unit for setting up the desired temperature.
The air compressor unit acts as the actuator.

Application Specific User Interface

These are embedded systems with application-specific user interfaces like buttons, switches, keypad, lights, bells, display units, etc.
Mobile phone is an example for this.
In mobile phone the user interface is provided through the keypad, graphic LCD module, system speaker, vibration alert, etc.

* by the mid-1980s, widespread use of embedded systems became feasible with microcontroller.
* the first mass-produced embedded system was Autonetics D-17 guidance computer.

Major applications of embedded systems:-

1) Robotic science:-
   - Ground vehicles
   - Drones
   - underwater vehicles
   - Industrial Robots

2) Automative:-
   - Engine control
   - Ignition system
   - Brake system

3) Networking:-
   - Router
   - gateways
   - electronics Instruments

4) Home devices:-
   - Refrigrator
   - TVS
   - Digital alarm
   - Air conditioner
   - cameras

5) Automobiles:-
   - Fuel Injection
   - Door locks
   - Air bags
   - windows
   - lighting system

6) Industrial control:-
   - Robotics
   - control system
   - Nuclear Reactors

7) entertainment systems:-
   1) video games
   - music system

8) health care:-
   - EEG Machines
   - ECG "
   - patient monitoring system

9) consumer electronics:-
   - camera, camcorders.

10) signal & image processing:-
   - speech processing
   - video processing.

(9) Processor and OS Trends in Embedded system

* An embedded system is a combination of computer hardware & software

↓ processor ↓ memory } single chip

* An embedded system is a processor with programs stored in RAM.

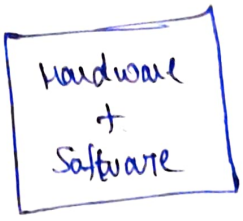
* watches, mechanical oven
car - Embedded system

Various recent trends in Embedded system are

(1) SOC (System on chip) *
(2) Wireless Technology
(3) multi-core processor *
(4) multi-language support
(5) User Interface
(6) use of opensource technology
(7) Inter-operatability
(8) Automation
(9) Security
(10) power consumption

1) Soc (system on a chip)

* A system on a chip is an integrated circuit that integrates all components of the system into a single chip



mobile electronic market

* low power consumption

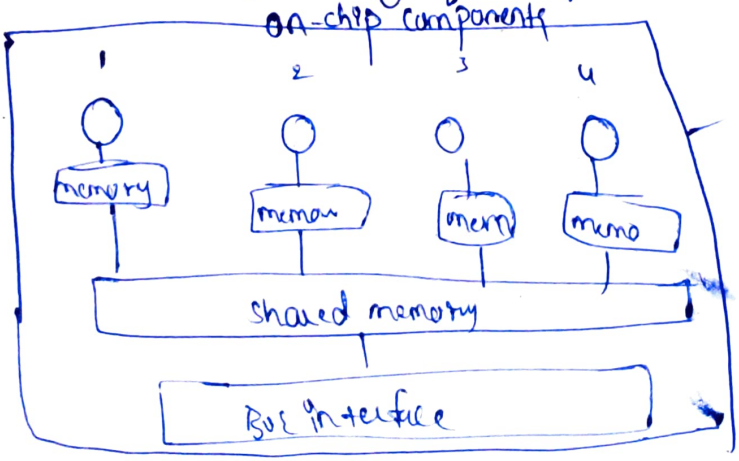
2) wireless technology

* wireless → no use of wires or cables
* cellular communications,
  networking blw computers with wireless adapters and wireless computers accessories

* over air via electromagnetic waves (radio freq, infrared, satellite etc)

3) multi-core processor

* A multi-core processor is an integrated circuit to which two or more processors have been attached for enhanced performance, reduced power consumption and more efficient simultaneous processing of multiple tasks



(4) multi-language support :- Embedded system provide facility of multiple languages
Example - ATM (Hindi, English, Punjabi)

(5) user Interface - visual part of computer application or operating system through which the user interacts with a computer or a software
* It determines how commands are given to the computer or the program and how information is displayed on the screen.

(6) use of open source technology :- refers to a program in which the source code is available to the general public for use and/or modification from its original design i.e open
* performance improve

(7) Automation :- use of various control systems for operating equipment like machinery, processes in factories, switching in telephone networks and other Application
* saves labor, save energy and materials quality, accuracy and precision

(8) Security :- secure Design, threat model Analysis, Software and code Testing, crypto implementation.

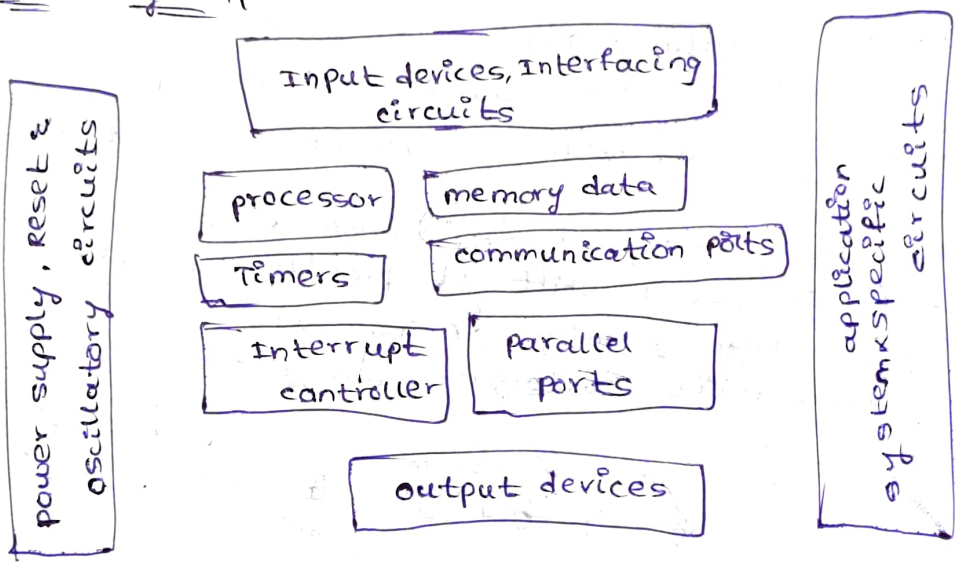
(9) power consumption :- Through power consumption it is possible to test and
↓ ↓
Hardware Software
then the system for power optimization.

① Explain the hardware components of an embeded system in detail

A) The hardware components of embedded system are

1) power source
2) oscillator circuit
3) Input / output ports
4) Interrupt controller.

block diagram:-



1) Power supply :-

1) most system have its own power source
2) certain systems without power source will be connected to external supply
3) voltage Range - 5V ± 0.25V, 3.3V ± 0.3V
4) 5V system processor and units are used in most high performance system.

2) Oscillator circuits:-

1) oscillator ensures the harmonic & synchronous operation of all circuits within the microcontroller
2) controls the time for execution an instruction, system timers, cpu machines cycles.
3) processor clock-out signal provides the clock for synchronizing all system units with the processor.
4) though the speed of execution depends upon the clock frequency, we cannot increase the clock frequency after

a maximum threshold limit otherwise the system becomes stable

3) ports :-

1) Input port - gets I/P from physical device.
2) Inputs from Touch screen, Keypad, sensor, modem
3) output port - system sends output bytes to the real world
4) LED, LCD, display pannel, alarms
5) serial & parallel ports
6) communication ~~ports~~ bus.

4) Interrupt controller :-

1) An interrupt is a condition that causes the microprocessor to temporarily work on a different task, and then later return to its previous task.
2) Interrupts can be external (or) internal.
3) hardware signal to indicate the occurence of event.
4) maskable & Non-maskable interrupt sources.

5) microprocessor (or) microcontroller :-

1) microprocessor :- used when large embedded software has to be located in external memory chips
2) microcontroller :- used when a small embedded software has to be located in the internal memory

6) System Timers :-

1) system clock drives timer for various timing and counting needs in a system
2) Reset circuit can change the program counter which increments or changes to power-up default value.
3) -> A system program that executes from the begining
4) A system boot-up program.

7) Memory :-

1) A system embeds
   * Initialization data
   * boot-up program.
2) temporary data
3) stack buffers,
4) RAM - stack, I/P (or) O/P buffers.

1.7.3. Software tools for designing embedded system:

A programming tool (or) software development tool is a program that software developers used to create, debug, maintain and support other programs and applications.

Editor :

- It is used for writing assembly mnemonics (or) high level programs like C/C++
- It allows entry, addition, deletion, insert, modification of programs.

Interpreter :

- It is used for expression by expression (line-by-line) translation to machine executable codes.

Compiler :

- Compiler is a software program that converts a source code from high level language *(human readable language)* into low level language *(object code or machine code)* to create an executable program.
- The compiler that runs on a computer platform and produces code for that same computer platform is called as Host compiler.
- A Compiler that runs on one computer platform and produces code for another computer platform is called as **Cross-compiler**.

Assembler :

- An assembler is a software program that converts the assembly language instructions into machine codes.
- An assembler that runs on host but produces binary codes appropriate for some other target is called **Cross-assembler**.

Linker:

- A linker is a program that takes one (or) more object files generated by compilers and combines them into a single executable file. Source code may be contained in more than one file. Hence, it must be combined.

Loader :

- In computer systems a **loader** is the part of an operating system that is responsible for loading programs and libraries.

Locator :

- A Locator is a tool that performs the conversion from re-locatable program to executable binary image that can be loaded into the target ROM.

Debugger :

- A **debugger** (or) **debugging tool** is a computer program that is used to test and debug other programs (the "target" program).

Simulator:

- It is a software tool used to simulate all functions of an embedded system including memory and peripherals. It also simulates the processes that will execute when the codes of a particular processor executes.

Integrated Development Environment (IDE)

- IDE is a fully integrated tool that consolidates basic software tools required for edit and test the embedded software. IDE consists of Editor, Compiler, Assembler, Cross assembler, Debugger, Simulator, Emulator, Logic analyzers and EPROM application codes burner.

1.9 Programming Languages

Code is typically written in C or C++, but various high-level programming languages, such as Python and JavaScript, are now also in common use to target microcontrollers and embedded systems. ADA is used in some military and aviation projects.

Machine code

Each instruction has a binary value called an op code. It is unrecognizable to humans, unless you spent a very long time on low-level debugging. Some very early computers had to be programmed in machine code, but that was long ago, thank goodness. You will see it, however, because the contents of memory are shown in the debugger and machine code is included in the-disassembly.

Assembly language

The instructions are written as words called mnemonics rather than binary values and a program called an assembler translates the mnemonics into machine code. It does a little more than direct translation, but not a lot, nothing like a compiler for a high-level language.

A major disadvantage of assembly language is that it is intimately tied to a processor and is therefore different for each architecture. Even worse, the detailed usage varies between development environments for the same processor. Most programming of small microcontrollers was done in assembly language until recently, despite these problems, mainly because compilers for C produced less-efficient code. Now the compilers are

better and modern processors are designed with compilers in mind, so assembly language has been pushed to the fringes. A few operations, notably bitwise rotations, cannot be written directly in C, and for these assembly languages may be much more efficient. However, the main argument for learning assembly language is for debugging. There is no escape if you need to check the operation of the processor, one instruction at a time. Disassembly is the opposite process to assembly, the translation of machine code to assembly language.

C

This is the most common choice for small microcontrollers now days. A compiler translates C into machine code that the CPU can process. This brings all the power of a high-level language—data structures, functions, type checking and so on—but C can usually be compiled into efficient code. Compilation used to go through assembly language but this is now less common and the compiler produces machine code directly.

C++

An object-oriented language that is widely used for larger devices. A restricted set can be used for small microcontrollers but some features of C++ are notorious for producing highly inefficient code. Embedded C++ is a subset of the language intended for embedded systems. Java is another object-oriented language, but it is interpreted rather than compiled and needs a much more powerful processor.

BASIC

Available for a few processors, of which the Parallax Stamp is a well-known example. The usual BASIC language is extended with special instructions to drive the peripherals. This enables programs to be developed very rapidly, without detailed understanding of the peripherals. Disadvantages are that the code often runs very slowly and the hardware is expensive if it includes an interpreter.

Embedded C

Embedded C is one of the most popular and most commonly used Programming Languages in the development of Embedded Systems. Embedded C is a set of language extensions for the C programming language. he Embedded C Programming Language uses the same syntax and semantics of the C Programming Language like main function, declaration of data types, defining variables, loops, functions, statements, etc. The extension in Embedded C from standard C Programming Language include I/O Hardware Addressing, fixed point arithmetic operations, accessing address spaces, etc.

1.8. SKILLS REQUIRED FOR EMBEDDED SYSTEM DESIGNER

An Embedded System designer has to develop a product using the available tools within the given specifications, cost and time frame.

Skills for Small Scale Embedded System Designer:

- ✓ Good knowledge of Microprocessor or Microcontroller to be used
- ✓ Computer architecture and organization
- ✓ Memories
- ✓ Memory allocation
- ✓ Interfacing memories
- ✓ Burning the executable machine codes in ROM
- ✓ Use of decoders and de-multiplexers
- ✓ Direct Memory Access
- ✓ Ports
- ✓ Device drivers
- ✓ Simple and Sophisticated Buses
- ✓ Timers
- ✓ Interrupt Service Mechanism
- ✓ C-Programming
- ✓ Memory optimization
- ✓ Selection of hardware and Microcontroller
- ✓ Use of ICE (In-Circuit Emulator), Cross Assembler and Testing equipment
- ✓ Debugging the software and Hardware bugs by Test vectors
- ✓ Basic Knowledge in other areas - Software engineering, data communication, Digital electronic design, control engineering, motors, actuators, sensors, measurements, analog electronic design and IC design and manufacturing

Skills for Medium Scale Embedded System Designer

- ✓ Knowledge of C /C++ /Java programming, RTOS programming and program modelling skills are must to design medium scale embedded-system.
- ✓ Tasks or threads and their scheduling by RTOS.
- ✓ Cooperative and pre-emptive scheduling
- ✓ Inter Process Communication functions
- ✓ Use of shared data ,programming critical sections and re-entrant functions
- ✓ Use of semaphores, mail boxes, queues, sockets and pipes
- ✓ Handling of interrupt latencies and meeting task deadlines.
- ✓ Use of various RTOS functions.
- ✓ Use of physical and virtual device drivers.
- ✓ Application Programming Interfaces (API)

Skills for Sophisticated Embedded System Designer

- ✓ A team is needed to co-design and solve the high level complexities of hardware and software design.
- ✓ Hardware engineers should have skills in hardware units and basic knowledge of C/C++ and JAVA,RTOS and other programming tool required.
- ✓ Software engineers should have basic knowledge in hardware and knowledge of C, RTOS and other programming tools.
- ✓ A final optimum design solution is then obtained by system integration.

examples of embedded systems:-

* Digital cameras
* digital wristwatches
* MP3 layers.
* appliances such as refrigrators, washing machine
* calculators
* temperature measurement system
* DVD players
* GPS Receivers
* Industrial Robotos.
* wireless routers
* Medical devices
* central heating systems.
* Fitness Trackers.
* automative systems.
* ATMs.
* Factory Robots.
* Transit & Fare collection.