Classification of Embedded Systems

1. Based on Generation

- First Generation

Built around 4- and 8-bit microprocessors/microcontrollers with firmware developed in Assembly code.
Examples: Digital telephone keypads, Digital door-locks.

- Second Generation

Built around 8- and 16-bit microprocessors/microcontrollers with more complex instruction set and powerful computing.Some of them may use an operating system for a supervised operation.
Example: Supervisory control and data acquisition (SCADA) systems like the Apollo Guidance System.

- Third Generation

Built around more powerful 16- and 32-bit microprocessors supporting instruction pipelining. Availability of application and domain specific processors/controllers like DSPs and ASICs.

- Fourth Generation

Advent of (System on Chip) SoCs, reconfigurable processors and multicore processors facilitated high performance, tight integration and miniaturization of embedded systems. SoC tech. aims to implement a complete system by integrating different functionalities with the processor core on an integrated circuit (IC).
One of the outputs of this generation is in everyone's (almost) hand - the ever smarter smartphone.

2. Based on Complexity

- Small Scale Embedded Systems

Systems designed with a single 8- or 16-bit microcontroller.Little hardware and software complexities and involve board-level design. May be battery operated.An editor, assembler and cross assembler, specific to the microcontroller used, are the main programming tools.

- Medium Scale Embedded Systems

Usually designed with a single or few 16- or 32-bit microcontrollers or DSPs or Reduced Instruction Set Computers (RISCs).
Involve both hardware and software complexities.

Programming tools involved: RTOS, Source code engineering tool, Simulator, Debugger and Integrated Development Environment (IDE).May employ readily available ASSPs and IPs for the various functions—for example, for the bus interfacing, encrypting, deciphering, discrete cosine transformation and inverse transformation, TCP/IP protocol stacking and network connecting functions.

- Sophisticated Embedded Systems:

Have enormous hardware and software complexities and may need scalable processors or configurable processors and programmable logic arrays. Development tools for these systems may not be readily available at a reasonable cost or may not be available at all. In some cases, a compiler or retargetable compiler might have to be developed.

APPLICATIONS

1. Consumer Applications

Device	Use of embedded systems
Washing machines	Control water & spin cycles
Remote controls	Process key touches and emit infrared (IR) pulses to base systems
Exercise equipment	Measures speed, distance, calories, heart rate, logs workouts
Clocks and Watches	Maintains the time, alarm and display

2. Communication

Device	Use of embedded systems
Telephone system	Interactive switching
Cellular phones and pagers	Keypad, touch-screens, Sound/Voice I/O, Digital Signal Processing
ATMs	Interfaces, Cash dispensing and retraction, Communication

3. Automotive

Device	Use of embedded systems
ABS	Optimizes stopping on slippery surfaces
Electronic ignition	Controls spark plugs and fuel injectors

4. Medical

Device	Use of embedded systems
Apnea monitors	Detects breathing and alarms if the baby stops breathing
Cardiac monitors	Measures heart functioning
Cancer treatments	Controls doses of radiation, drugs, or heat
Prosthetic devices	Increases mobility for the handicapped
Pacemaker	Helps the heart beat regularly