

Basic Embedded Systems

Definition

An embedded system is a dedicated computer inside a larger product, designed for one or a limited set of tasks. Examples: washing-machine controller, digital camera, vehicle ECU and thermostat.

Microprocessor versus microcontroller

Microprocessor	Microcontroller
CPU usually needs external RAM, storage and peripherals	CPU, memory and I/O peripherals integrated on one chip
general-purpose computing	dedicated control applications
higher external hardware complexity	lower-cost, compact embedded design

Core components

- CPU: executes instructions.
- Flash/ROM: non-volatile program storage.
- RAM: volatile temporary data storage.
- GPIO: general-purpose input/output pins.
- Timer/counter: measures time or counts events.
- ADC: converts analog sensor voltage to digital value.
- DAC: converts digital value to analog signal.
- PWM: simulates variable analog output using duty cycle.
- Watchdog timer: resets/reports when software hangs.

Interrupt versus polling

Interrupt	Polling
hardware/event notifies CPU	CPU repeatedly checks device/status
efficient for infrequent events	simple, but can waste CPU time
ISR must be short and safe	response depends on polling frequency

An interrupt service routine should save necessary state, do minimum urgent work, clear/acknowledge the event, and return quickly.

Communication interfaces

Protocol	Wires/basic model	Main trait
UART	TX, RX; asynchronous	simple serial communication, no shared clock
SPI	clock plus data/select lines	synchronous, fast, full duplex, master-peripheral
I2C	SDA and SCL	two-wire addressed bus, multiple devices
CAN	differential bus	robust, priority-based vehicle/industrial network

Real-time systems

Correctness depends on result and timing.

- Hard real-time: missing deadline can cause serious failure; example safety controller.
- Soft real-time: late result reduces quality but may be tolerated; example multimedia.
- RTOS: operating system designed for predictable timing/scheduling.

Embedded design answer pattern

For a sensor-controller MCQ, map:

sensor → ADC/input → microcontroller/firmware → output/actuator.

If the system may hang unattended, choose watchdog. If an urgent external event arrives, choose interrupt. If analog data must be read, choose ADC.

[[03_Technical/Networking|Networking]] · [[03_Technical/Technical_Quick_Recall|Quick recall]]