Module 4 PART A: EMBEDDED SYSTEM

Definition: An embedded system is a microcontroller or microprocessor based system which is designed to perform a specific task in combination of both hardware and software.

E.g. Electronic Toys, Mobile Handsets, Washing Machines, Air Conditioners, Automotive Control Units, Set Top Box, DVD Player etc…

For example, a fire alarm; it will sense only smoke.

Laser printers; it only prints

Firmware: programming instructions, referred to as firmware, are stored in read-only memory (ROM).

Embedded Systems Vs General Computing Systems

General Computing System	Embedded Systems
It is microprocessor based system	It is microcontroller based system
A computer needs human interaction to perform tasks.	Embedded device does not need human interaction to perform tasks.
Architecture examples: Analog / Digital computer, Hybrid computer, Harvard /Von Neumann architecture, Reduced instruction set computer	**Architecture examples**: Small Scale Embedded System, Medium Scale Embedded Systems, Sophisticated or Complex Embedded Systems
It has 2 parts: Hardware and Software.	It has 3 parts: Hardware, Firmware and Software.
It can perform many tasks.	It performs specific tasks
(End user programmable) can be reprogrammed to for a new purpose.	(Not end user programmable) only for a specific set of purposes.
Computers are usually bigger in size with larger hardware and input output devices attached to it.	Embedded Devices are smaller in size than Computers, with limited hardware.
Power consumption is high	Power consumption is less

Classification of Embedded Systems

- Based on Generation
- Based on Complexity & Performance Requirements
- Based on deterministic behavior
- Based on Triggering

Based on Generation

First Generation

The earlier first-generation embedded systems were built around 8-bit microprocessors and 4-bit microcontrollers. Such embedded system possesses simple hardware and firmware developed using assembly code.

Ex: Digital telephone keypads, stepper motor control units.

Second Generation

After the evolution of the second generation embedded systems, the 8-bit processor and 4-bit controllers are replaced by 16-bit microprocessors and 8-bit microcontrollers. They are more powerful and complex compared to previous generation processors.

Ex: Data acquisition systems, SCADA systems.

Third Generation

Embedded Systems built around high performance 32-bit microprocessors and 16-bit microcontrollers. Hence, its operation has become much more powerful and complex than the second generation.

During this period, domain-specific processors/controllers like Digital Signal Processors (DSP), Application-Specific Integrated Circuits (ASICs) and the concept of instruction pipelining, embedded real-time operating system evolved into the embedded system industry.

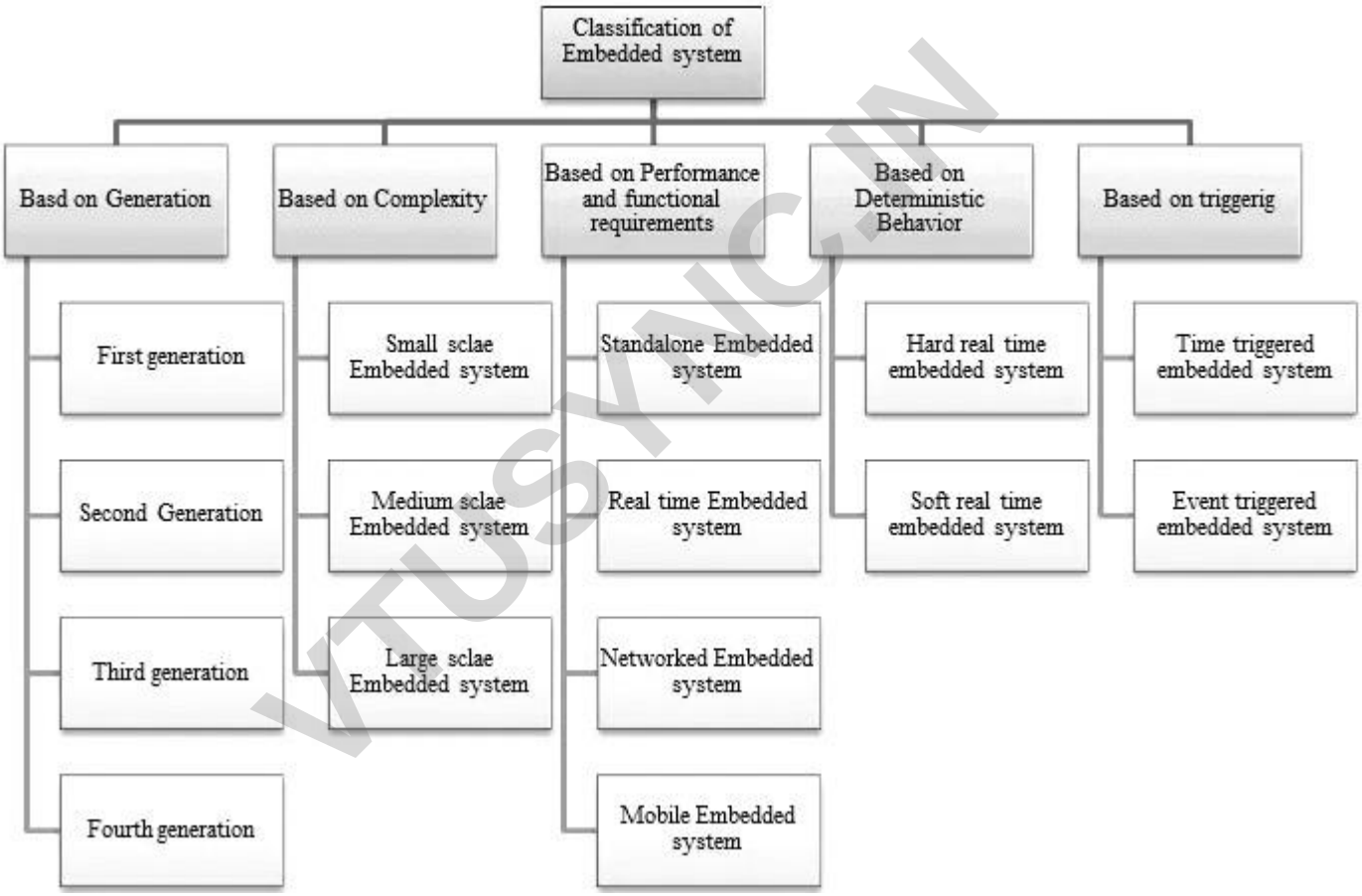
Ex: Robotics, industrial process control, embedded networking.

Fourth Generation

Embedded Systems built around System on Chips (SoCs), Re-configurable processors and multi-core processors, coprocessors also emerged into the embedded market to add more powerful performance

These systems also make use of the high-performance real-time operating system for their operation.

Ex: Smart devices, digital cameras, etc.



Based on Complexity and Performance Requirements

Small Scale Embedded Systems

Small Scale Embedded Systems are built with a single 8 or 16-bit microprocessor or controller. The main programming tools used are an editor, assembler, cross assembler and integrated development environment (IDE). The hardware and software complexities in small-scale embedded system are very low. It may or may not contain an operating system for its functioning. An electronic toy is an example for a small-scale embedded system.

Medium Scale Embedded Systems

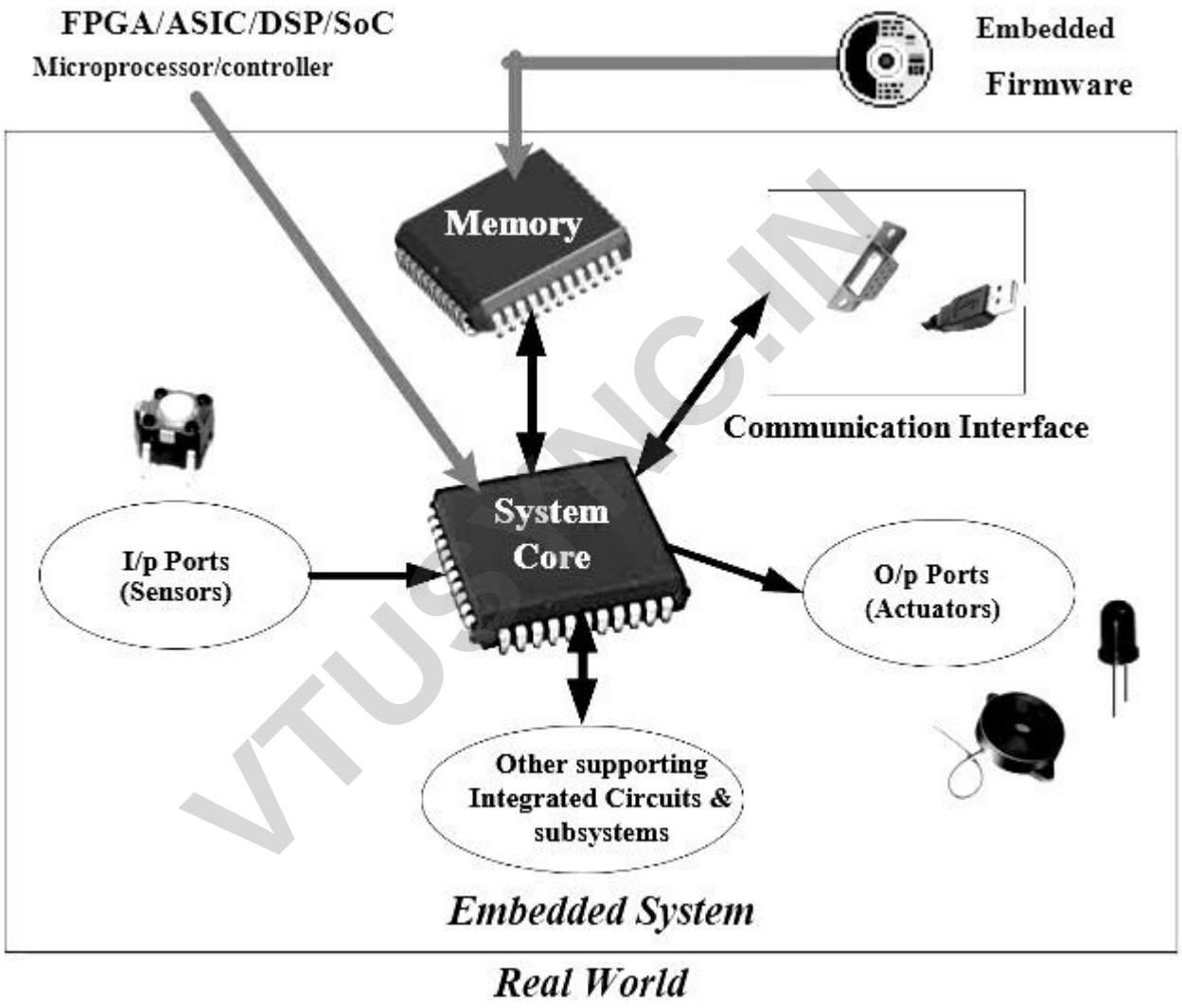
The Embedded system with medium performance 16-bit or 32-bit microprocessor or controller, ASICs or DSPs fall under the medium scale embedded systems. They have both hardware and software complexities.

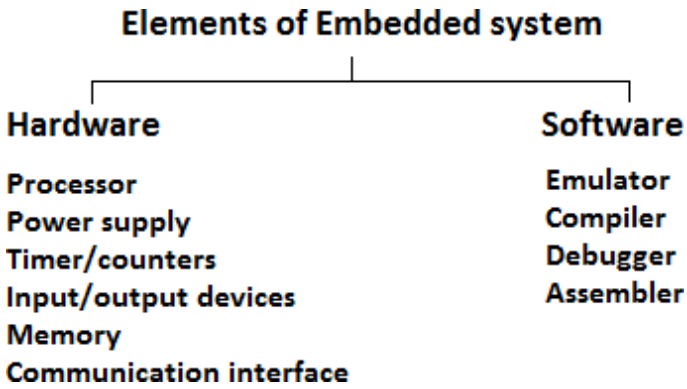
The main programming tools used are C, C++, JAVA, Visual C++, RTOS, debugger, source code engineering tool, simulator and IDE.

Large scale Embedded Systems

The embedded systems have highly complex hardware and software, built around 32-bit or 64-bit processors/controllers, RISC processors, SoC, scalable and configurable processors. They are also called sophisticated embedded systems.

They are used for cutting-edge applications that need hardware and software co-design, where components have to be assembled into the final system. They also contain a high-performance real-time operating system for task scheduling, prioritization and management.





Major Application Areas of Embedded Systems

- **Consumer Electronics**: Cam-corders, Digital Cameras, Laptop, CCTV etc.
- **Household Appliances**: Television, DVD players, Washing machine, Fridge, Microwave Oven etc.
- **Home Automation and Security Systems**: Air conditioners, sprinklers, Intruder detection alarms, Closed Circuit Television Cameras, Fire alarms etc.
- **Automotive Industry**: Anti-lock breaking systems (ABS), Engine Control, Ignition Systems, Automatic Navigation Systems etc.
- **Telecom**: Cellular Telephones, Telephone switches, Handset Multimedia Applications etc.
- **Computer Peripherals**: Printers, Scanners, Fax machines etc.
- **Computer Networking Systems**: Network Routers, Switches, Hubs, Firewalls etc.
- **Health Care**: X-ray, Scanners, EEG, ECG, BP monitor, pulse monitor etc.
- **Measurement & Instrumentation**: Digital multi meters, Digital CROs, Logic Analyzers PLC systems etc.
- **Banking & Retail**: Automatic Teller Machines (ATM) and Currency counters,
- **Card Readers**: Barcode, Smart Card Readers, Hand held Devices etc.

The Core of the Embedded Systems

The core of the embedded system falls into any one of the following categories.

- General Purpose and Domain Specific Processors
  - Microprocessors
  - Microcontrollers
  - Digital Signal Processors (DSP)
- Programmable Logic Devices (PLDs)
- Application Specific Integrated Circuits (ASICs)
- Commercial off the shelf Components (COTS)

Microprocessor Vs Microcontroller, RISC vs CISC

Microprocessor	Microcontroller
Consists of a CPU, performs Arithmetic and Logical operations **Ex: Intel 8086 microprocessor**	Highly integrated chip contains CPU, RAM, on chip ROM/flash memory, I/O ports **Ex: Intel 8051 microcontroller**
It is mainly used in Personal Computers	It is mainly used in an embedded system
Complex and expensive, with a large number of instructions to process.	Simple and inexpensive with less number of instructions to process.
Dependent Unit	Self contained unit
Consumes more power	Consumes less power
Limited power saving options	Includes lot of power saving features
Architecture is based on Von Neumann model	Architecture is based on Harvard architecture
Uses an external bus to interface to RAM, ROM, and other peripherals	Uses an internal controlling bus.

RISC	CISC
Reduced Instruction Set Computer.	Complex Instruction Set Computer.
Software centric design.	Hardware centric design.
Low power consumption.	High power consumption.
Requires more RAM	Requires a minimum amount of RAM
Simple decoding of instruction.	Complex decoding of instruction.
Processors are highly pipelined.	Processors are not pipelined or less pipelined.
Execution time is very less	Execution time is very high
Uses multiple registers.	Uses a single register.
It does not require external memory for calculations	It requires external memory for calculations
Compound addressing mode.	Limited addressing mode.
RISC architecture can be used with high-end applications like telecommunication, image processing, video processing, etc.	CISC architecture can be used with low-end applications like home automation, security system, consumer goods etc.
Small Code Size.	Large Code Size.
Fixed Instruction format (32-bit)	Varying formats (16 to 64 bits for each instruction).
Examples: ARM, PIC, Power Architecture, Alpha, AVR, ARC and the SPARC.	Examples: VAX, Motorola 68000 family, System/360, AMD and the Intel x86 CPUs.

PART B: Sensors and Interfacing

Instrumentation and control systems

Instrumentation: Technology of measurement

An instrument is a device that measures or manipulates process physical variables such as flow, temperature, level, or pressure etc.

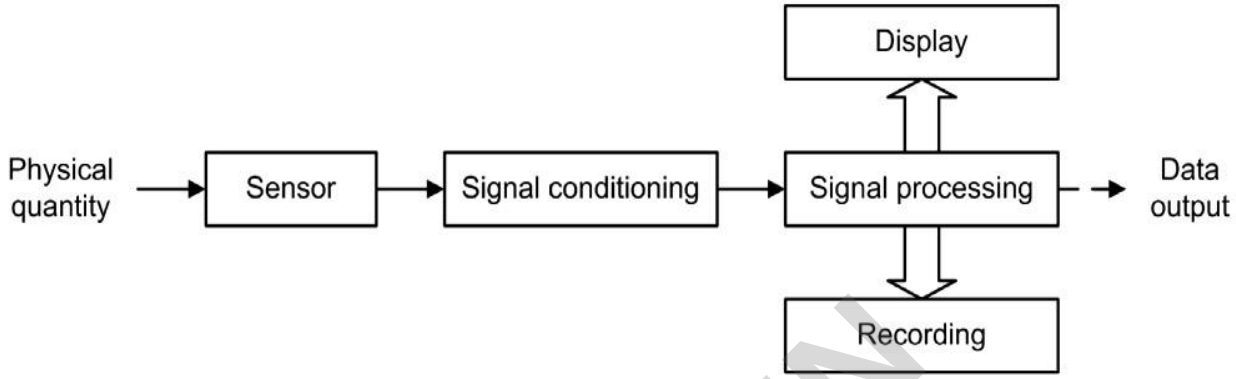


Fig. 1 Instrumentation system

Fig.1 shows the arrangement of an instrumentation system. The physical quantity to be measured (e.g. temperature) acts upon a sensor that produces an electrical output signal.

This signal is an electrical analogue of the physical input but there may not be a linear relationship between the physical quantity and its electrical equivalent.

Also, the output produced by the sensor may be small or may suffer from the presence of noise (i.e. unwanted signals). Therefore, further signal conditioning will be required before the signal will be at an acceptable level and in an acceptable form for signal processing, display and recording. The signal processing may use digital rather than analog signals for this purpose ADC may be required.

Control systems

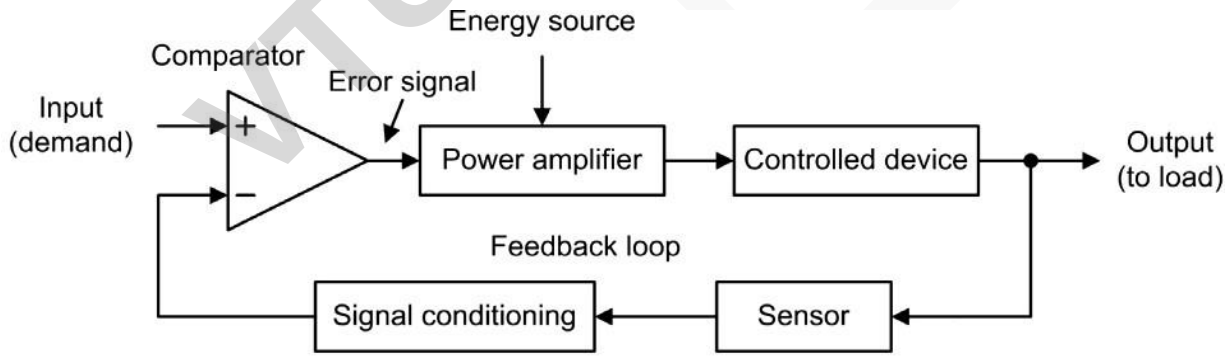


Fig.2 shows the arrangement of a control system

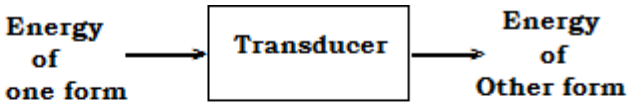
As seen from the fig.2, it uses negative feedback in order to regulate and stabilize the output. It thus becomes possible to set the input or demand (i.e. what we desire the output to be) and leave the system to regulate itself by comparing it with a signal derived from the output (via a sensor and appropriate signal conditioning).

A comparator is used to sense the difference in these two signals and where any changes detected, then input to the power amplifier is adjusted accordingly. This signal is referred to as an *error signal* (it should be zero when the output exactly matches the demand).

The input (demand) is often derived from a simple potentiometer connected across a stable DC voltage source while the controlled device can take many forms (Ex: a DC motor, linear actuator, heater, etc.).

Transducers

Transducers are devices that convert energy in the form (sound, light, heat, etc.,) into an equivalent electrical signal, or vice versa.



Transducer Type

Sensor: (Physical form) ⟶ (Electrical form)

Actuator: (Electrical form) ⟶ (Physical form)

- Examples:
  - A microphone is a transducer converting sound pressure variations into voltage or current.
  - A loudspeaker is a transducer that converts low frequency electric current into audible sounds.

Difference between Sensor and Actuator

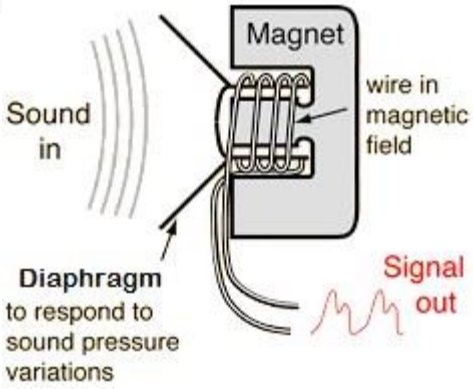
SENSOR	ACTUATOR
It converts physical quantity into electrical signals.	It converts electrical signals into physical quantity.
It takes input from environment.	It takes input from the electric or electronic system.
Sensor generated electrical signals.	Actuator generates heat, motion, vibration,etc.
It is placed at input port of the system.	It is placed at output port of the system.
It is used to measure the physical quantity.	It is used to measure the continuous and discrete process parameters.
It gives information to the system about environment.	It accepts command from the system to perform a function.
Example: Photo-voltaic cell which converts light energy into electrical energy.	Example: Stepper motor where electrical energy drives the motor.

Sensor

A sensor is a transducer which converts energy from physical form to electrical form for any measurement or control purpose. Sensors act as input device.

Example 1:

Physical quantity: **sound**
Input transducer: **Microphone**



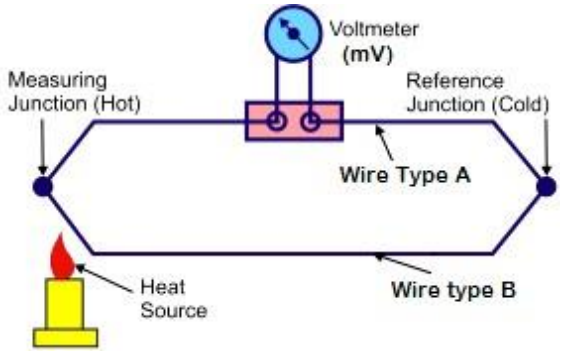
Diaphragm attached to a coil is suspended in a magnetic field. Movement of the diaphragm causes current to be induced in the coil.

Sensors can be categorized as either active or passive.
An *active sensor* generates a current or voltage output.
A *passive sensor* requires a source of current or voltage and it modifies this by virtue of a change in the sensor's resistance. The result may still be a voltage or current but it is not generated by the sensor on its own.
Sensors can also be classified as either *digital* or *analog*.

The output of a digital sensor can exist in only two discrete states, either 'ON' or 'OFF', 'LOW' or 'HIGH', 'logic 1' or 'logic 0', etc. The output of an analogue sensor can take of voltage or current levels. It is thus said to be continuously variable.

Example 2:

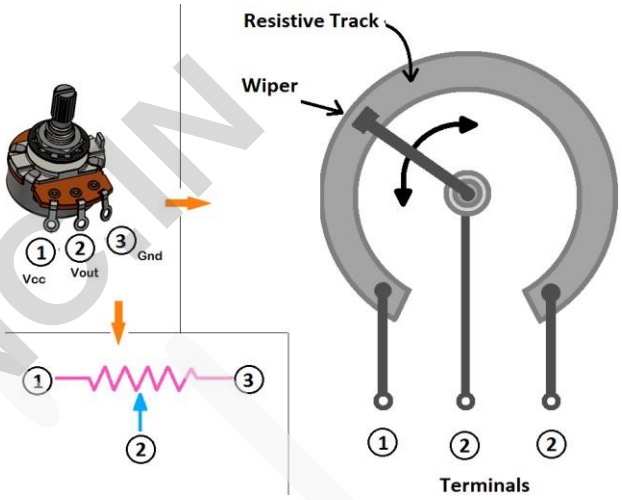
Physical quantity: **Temperature**
Input transducer: **Thermocouple**



Small e.m.f (mV) generated at the junction between two dissimilar metals (e.g. copper & constantan). Requires reference junction (cold) and compensated cables for accurate measurement.

Example 3:

Physical quantity: **Angular position**
Input transducer: **Rotary potentiometer**



Fine wire resistive element is wound around a circular frame. Slider (wiper) attached to the control shaft makes contact with the resistive element.
A stable DC voltage source is connected across the ends of the potentiometer. Voltage appearing at the slider will then be proportional to angular position.

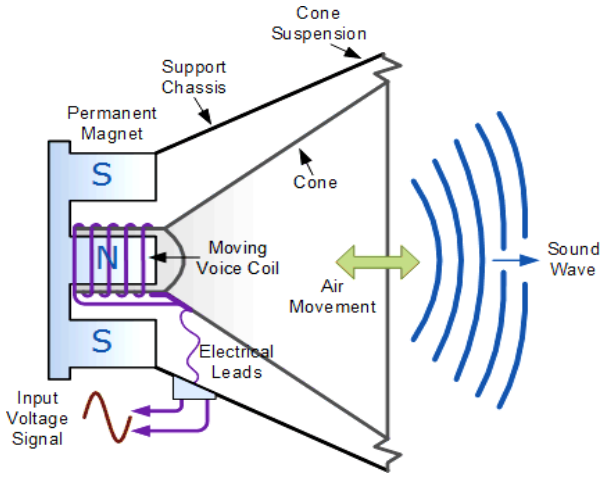
Transducers are also known as Input-Output subsystems. The I/O subsystem facilitates the interaction of the embedded system with external world. The interaction happens through the sensors and actuators connected to the input and output ports of the embedded system. The sensors may not be directly interfaced to the input ports, instead they may be interfaced through signal conditioning and translating systems like ADC, Opto-couplers etc.

Actuators

Actuator is a transducer which converts electrical signals to corresponding physical action (motion). Actuator acts as an output device.

Example :1

Physical Quantity: **Sound** (pressure change)
Output transducer: **Loudspeaker**



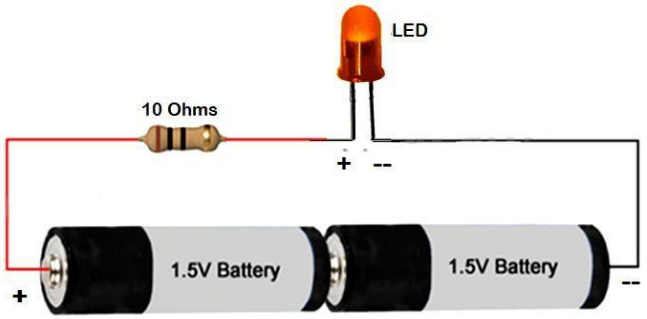
Diaphragm attached to a coil is suspended in a magnetic field. Current in the coil causes movement of the diaphragm which alternately compresses and rarefies the air mass in front of it.

Examples: electric motors, stepper motors, jackscrews, electric muscular stimulators in robots,

Light Emitting Diode (LED) and 7-Segment LED Display

Light Emitting Diode (LED): is an output device for visual indication in any embedded system. It can be used as an indicator for the status of various signals or situations. Typical examples are indicating the presence of power conditions like device ON, battery low or charging of battery etc.

LED is a p-n junction diode and it contains an anode (+) and a cathode (-). For proper functioning of the LED, the anode is connected to +ve terminal and cathode to the –ve terminal of supply voltage (forward bias condition). The current flowing through the LED must limited to a value below the maximum current that it can conduct. A resister is used in series between the power supply and the resistor to limit the current through it.



- LED can be interfaced to the port pin of microcontroller in two methods:
- **Method:1-** (current sourcing) Anode of LED is connected to port pin. Cathode is connected to ground (0V) through resistor. When port pin of microprocessor goes logic 1, the LED is forward biased and emits light. When the port pin goes 0, LED is *off*. That means port pin *sources* current to LED.
- **Method:2-** (current sinking) Cathode of LED is connected to port pin. Anode is connected to external supply through resistor. LED turns *on* when the port pin is at logic 0. Here port pin *sinks* current, such that the brightness of LED can be increased to the required level. See fig.3.

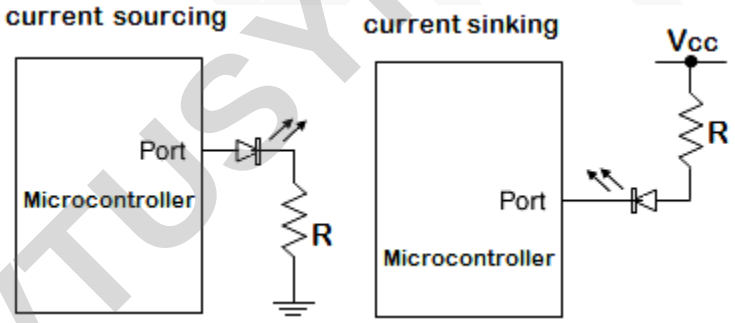


Fig.3 LED interfacing to the port pin of microcontroller

7-Segment LED Display

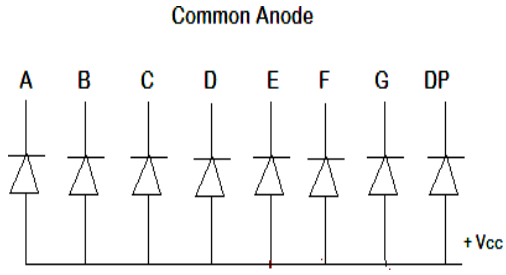
The 7 – segment LED display is an output device for displaying alpha numeric (0–9 and A–F) characters. It contains eight LED segments arranged in a special form. Out of the 8 LED segments, 7 are used for displaying alpha numeric characters. The LED segments are named A to G and the decimal point LED segment is named as DP. The LED Segments A to G and DP should be lit accordingly to display numbers and characters.

The 7 – segment LED displays are available in two different configurations, namely;

i) common anode and ii) common cathode

In the *common anode* configuration, the anodes of all LEDs connected together to +Vcc, and in the *common cathode* configuration, the cathodes of all LEDs connected together to ground as shown in the fig.4.

Based on the configuration of the 7 – segment LED unit, the LED segment anode or cathode is connected to the port of microcontroller in the order. A segment to the least significant port pin and DP segment to the most significant port pin or vice versa. The current through each segment can be limited by connecting a resistor.



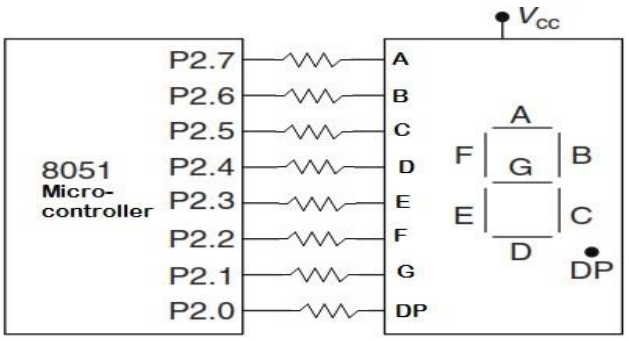
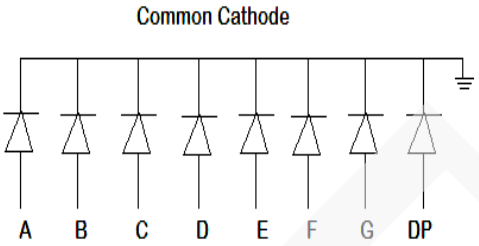
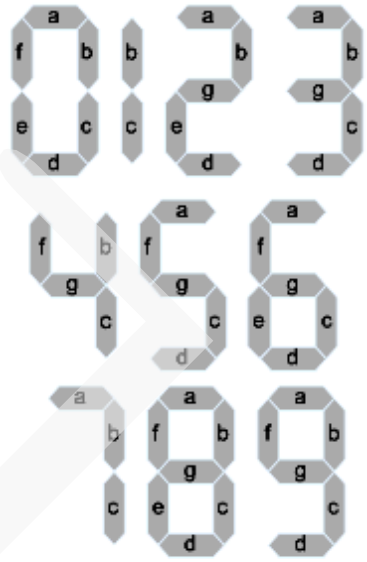


Fig.4 7-segment LED interfacing with microcontroller



Number	Code	LED status							
		A	B	C	D	D	E	G	DP
0	03	0	0	0	0	0	0	1	1
1	9F	1	0	0	1	1	1	1	1
2	25	0	0	1	0	0	1	0	1
3	0D	0	0	0	0	1	1	0	1
4	D9	1	1	0	1	1	0	0	1
5	49	0	1	0	0	1	0	0	1
6	41	0	1	0	0	0	0	0	1
7	1F	0	0	0	1	1	1	1	1
8	01	0	0	0	0	0	0	0	1
9	19	0	0	0	1	1	0	0	1



7 segment Display for (0 - 9)

In order to display the required numbers (0 - 9) or HEX characters (A - F), the correct combination of LED segments need to be illuminated based on the type of configuration. In the above example *common cathode* configuration is used.