Module-3

Embedded Systems

MODULE III

- Embedded Systems: Definition, Embedded systems vs. General computing systems, Classification of Embedded Systems, Major application areas of Embedded Systems, Elements of an Embedded System, Core of the Embedded System, Microprocessor vs Microcontroller, RISC vs. CISC, and Harvard vs. Von-Neumann.
- Sensors and Interfacing: Instrumentation and control systems, Transducers, Sensors
- Actuators: LED, 7-Segment LED Display, Stepper Motor, Relay, Piezo Buzzer, Push Button, Switch,Keyboard.
- Communication Interface: UART, Parallel Interface, USB, Bluetooth, Wi-Fi, GPRS.

Embedded Systems

- An Electronic/Electro mechanical system which is designed to perform a specific function and is a combination of both hardware and firmware (Software).
- E.g. Electronic Toys, Mobile Handsets, Washing Machines, Air Conditioners, Automotive Control Units, Set Top Box, DVD Player

Embedded Systems vs. General Computing Systems

General Purpose Computing System	Embedded Systems
A system which is a combination of generic hardware and General Purpose Operating Systemfor executing a variety of applications.	A system which is a combination of special purpose hardware and embedded Operating System for executing a specific set of applications.
Contain a General Purpose Operating System (GPOS).	May or may not contain an operating system for functioning.
Applications are alterable (programmable) by user. (The end user can re-install the OS, and add or remove userapplications).	The firmware of the embedded system is pre-programmed and it is non-alterable by end-user.
Performance is the key deciding factor on the selection of the system. Always "Faster is Better".	Application specific requirements (like performance, power requirements, memory etc) are the key deciding factors.
Less/not at all tailored towards reduced operating power requirements, options for different levels of power management.	Highly tailored to take advantage of the power saving modes supported by hardwareand Operating System.
Response requirements are not time critical.	For certain category of ES, the response time requirement is highly critical.
Need not be deterministic in execution behavior.	Execution behavior is deterministic for few ES like "Hard Real Time‟ systems.

Classification of Embedded Systems

The criteria used in the classification of embedded systems are:

1. Based on generation.
2. Complexity and performance requirements.
3. Based on deterministic behaviour.
4. Based on triggering.

Classification Based on Generation

- **First Generation**: The early embedded systems were built around 8-bit microprocessors like 8085 and 280, and 4-bit microcontrollers. Simple in hardware circuits with firmware developed in Assembly code. Eg: Digital telephone keypads, stepper motor control units etc.

- **Second Generation:** Embedded systems built around 16-bit microprocessors and 8 or 16-bit microcontrollers. The instruction set became much more complex and powerful than the first generation processors/controllers. Eg: Data Acquisition Systems, SCADA systems etc.

- **Third Generation:** ES built around powerful 32-bit processors and 16-bit microcontrollers. Application and domain specific processors /controllers like Digital Signal Processors (DSP) and Application Specific Integrated Circuits (ASICs) came into the picture. The instruction set is more complex and powerful. Eg: Robotics, media, industrial process control, networking, etc.

- **Fourth Generation:** Embedded system built around System on Chips (SOC), reconfigurable processors and multicore processors. The fourth generation embedded systems are making use of high performance real time embedded operating systems for their functioning. Eg: Smart phone devices, mobile internet devices (MIDs), etc

Classification Based on Complexity and Performance

- **Small-Scale Embedded Systems**: Small-scale ES are usually built around low performance and low cost 8 or 16 bit microprocessors / microcontrollers. These ES are suitable for simple applications and where performance is not time critical. It may or may not contain an operating system (OS) for functioning. Eg: Electronic toy.

- **Medium-Scale Embedded Systems**: Embedded systems built around medium performance, low cost 16 or 32 bit microprocessors/microcontrollers or DSPs. These embedded systems which are slightly complex in hardware and firmware (software) requirements. It usually contains an Embedded OS for functioning.

Large-Scale Embedded Systems/Complex Systems:

- ES built around 32 or 64 bit RISC processors/controllers or Reconfigurable System on Chip (RSC) or multicore processors and programmable logic devices.
- These embedded systems involve highly complex hardware and firmware. They may contain multiple processors/controllers and co-units/hardware accelerators.
- Complex embedded systems usually contain a high performance Real Time Operating System (RTOS).

Classification based on deterministic system behaviour

- It is applicable for Real Time systems. The application/task execution behavior for an embedded system can be either deterministic or non-deterministic.

Classified into

- Soft Real Time Systems: Missing a deadline may not be critical and can be tolerated to a certain degree. Eg: ATM.
- Hard Real Time Systems: Missing any deadline may produce disastrous results (financial, human loss of life, etc.). Eg: ABS, Air bags etc.

Classification based on triggering

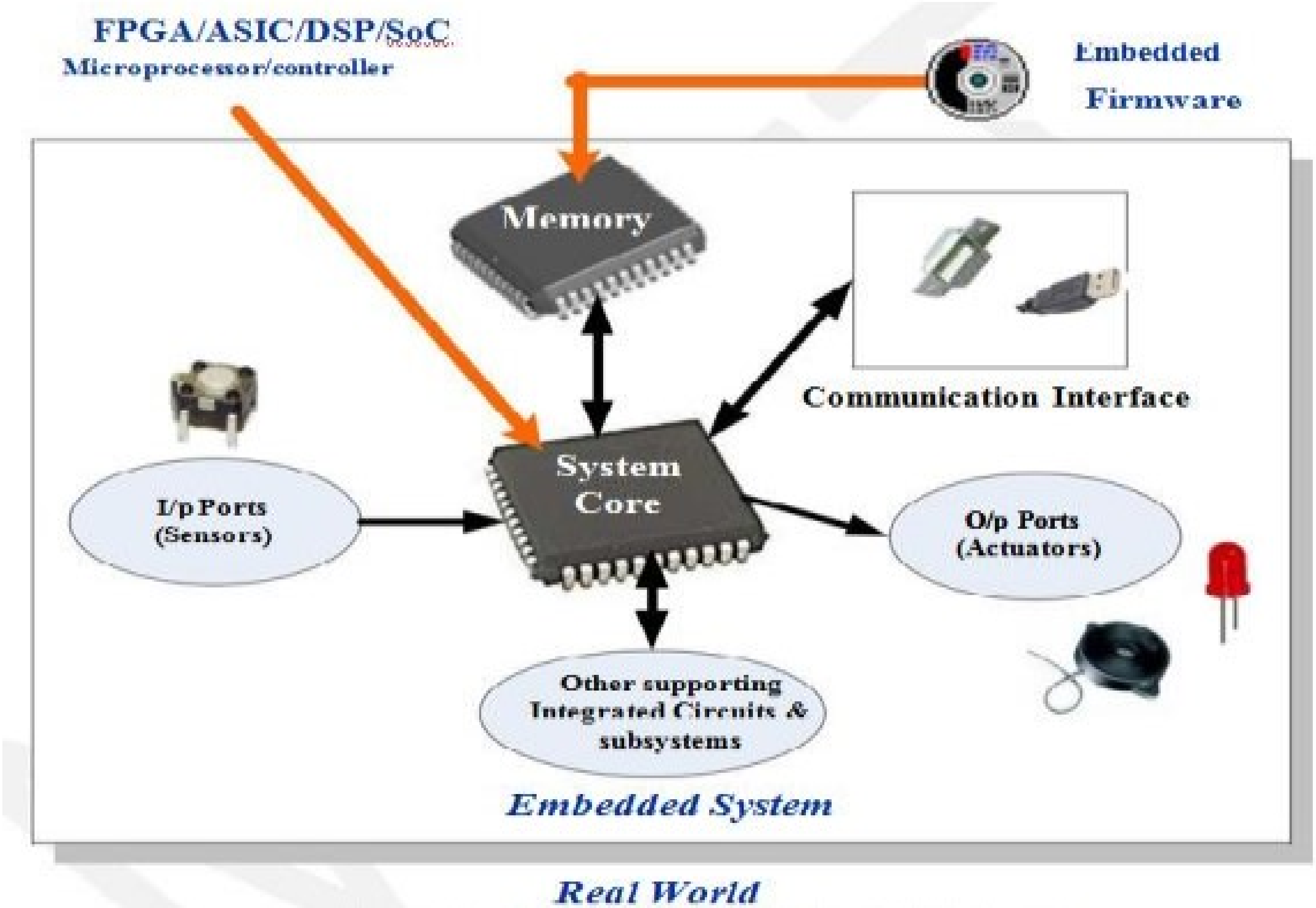
- Embedded Systems which are 'Reactive' in nature can be classified based on the trigger.
- Reactive systems can be classified as event triggered or time triggered.

Major Application Areas of Embedded Systems

- Consumer Electronics: Camcorders, Cameras etc.
- Household Appliances: Television, Washing Machine, Fridge, Microwave Ovenetc.
- Home Automation and Security Systems: Air conditioners, CCTVs, Fire alarms etc.
- Automotive Industry: Anti-lock Breaking Systems (ABS), Engine Control, Automatic Navigation Systems etc.
- Telecom: Cellular Telephones, Telephone switches, Handset etc.
- Computer Peripherals: Printers, Scanners, Fax machines etc.
- Computer Networking Systems: Network Routers, Switches, Hubs, Firewalls etc

- Health Care: Different Kinds of Scanners, EEG, ECG Machines etc.
- Measurement & Instrumentation: Digital multi meters, Digital CRO, PLCsystems etc.
- Banking & Retail: Automatic Teller Machines (ATM), Currency counters etc.
- Card Readers: Barcode, Smart Card Readers, Hand held Devices etc.

Elements of Embedded Systems



- An embedded system is a combination of 3 things: Hardware, Software, Mechanical Components and it is supposed to do one specific task only.
- A typical embedded system contains a single chip controller which acts as the master brain of the system.
- Diagrammatically an embedded system can be represented as shown in figure.
- ESs are designed to regulate a physical variable or to manipulate the state of some devices by sending signals to the actuators or devices connected to the output system, in response to the input signal provided by the end users or sensors which are connected to the input ports.
- Hence the embedded systems can be viewed as a reactive system.

- The control is achieved by processing the information coming from the sensors and user interfaces, and controlling some actuators that regulate the physical variable.
- Keyboards, push button, switches, etc. are examples of input devices and LEDs, LCDs, Piezoelectric buzzers, etc. are examples for output devices for a typical ES.
- Some ES can automatically sense input parameters from real world through sensors. Sensor information is passed to the processor after signal conditioning and digitization.
- The core of the system performs some predefined operations on input data with the help of embedded firmware and sends some actuating signals to the actuator.

- The memory of the system is responsible for holding the code.

There are two types:

- Fixed memory (ROM) is used for storing code or program. The user cannot do any modifications in this type of memory. Common types used are OTP, PROM, EPROM, EEPROM & Flash memory.
- Temporary memory (RAM) is used for performing arithmetic operations or control algorithm executions. Common types used are SRAM, DRAM and NVRAM.
- Memory for implementing the code may be present on the processor or may be implemented as a separate chip interfacing the processor.
- In a controller based ES, the controller may contain internal memory for storing code and such controllers are called Micro-controllers with on-chip ROM, eg. Atmel AT89C51

Core of the Embedded System

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

1. General Purpose and Domain Specific Processors

- Microprocessors
- Microcontrollers
- Digital Signal Processors

2. Programmable Logic Devices (PLDs)
3. Application Specific Integrated Circuits (ASICs)
4. Commercial off-the-shelf Components (COTS)

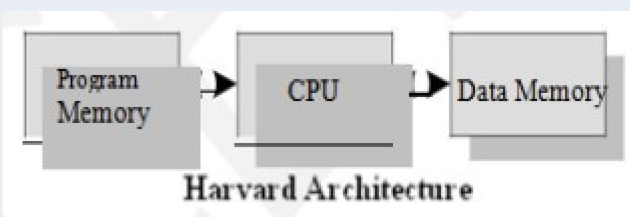
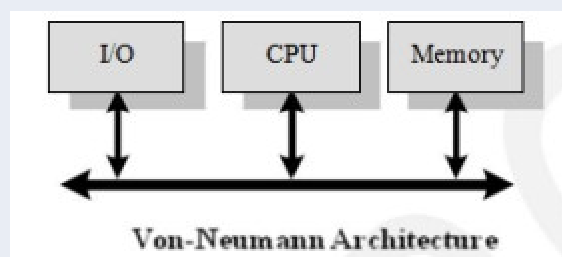
Microprocessors vs. Microcontrollers

Microprocessors	Microcontrollers
A silicon chip representing a Central Processing Unit(CPU), which is capable of performing arithmetic as well as logical operations according to a pre-defined set of Instructions.	A microcontroller is a highly integrated chip that contains a CPU, RAM, Special and General purpose Register Arrays, On Chip ROM/FLASH memory for program storage, Timer and Interrupt control units and dedicated I/O ports
It is a dependent unit. It requires the combination of other chips like Timers, Program and data memory chips, Interrupt controllers etc. for functioning.	It is a self contained unit and d o e s n ' t require external Interrupt Controller, Timer, and UART etc. for its functioning.
Most of the time general purpose in design and operation.	Mostly application oriented or domain specific.
Doesn't contain a built in I/O port. The I/O Port functionality needs to be implemented with the help of external Programmable Peripheral Interface Chips	Most of the processors contain multiple built-in I/O ports which can be operated as a single 8 or 16 or 32 bit Port or as individual port pins.
Targeted for high end market where performance is important.	Targeted for embedded market where performance is not so critical.
Limited power saving options.	Includes lot of power saving features

RISC v/s CISC Processors/Controllers:

RISC Processors/Controllers	CISC Processors/Controllers
Lesser no. of instructions.	Greater no. of Instructions.
Instruction Pipelining and increased execution speed.	Generally no instruction pipelining feature.
Orthogonal Instruction Set.	Non Orthogonal Instruction Set.
Operations are performed on registers only, the only memory operations are load and store	Operations are performed on registers or memory depending on the instruction
Large number of registers are available	Limited no. of general purpose registers
Programmer needs to write more code to execute a task since the instructions are simpler ones.	A programmer can achieve the desired functionality with a single instruction.
Single, Fixed length Instructions.	Variable length Instructions.
Less Silicon usage and pin count.	More silicon usage.
With Harvard Architecture	Harvard or Von-Neumann Architecture

Harvard V/s Von-Neumann Processor/Controller Architecture

Harvard Architecture	Von-Neumann Architecture
Microprocessors/controllers based on the Harvard architecture will have separate data bus and instruction bus. This allows the data transfer and program fetching to occur simultaneously on both buses.	Microprocessors/controllers based on the Von-Neumann architecture shares a single bus for fetching both instructions and data. Program instructions & data are stored in a common main memory.
Separate buses for Instruction and Data fetching.	Single shared bus for Instruction and Data fetching.
Easier to Pipeline, so high performance can be achieved.	Low performance Compared to Harvard Architecture.
Comparatively high cost.	Cheaper.
No memory alignment problems	Allows self modifying codes
	

sensors and Interfacing

Instrumentation and control systems, Transducers, Sensors.

Actuators

LED, 7-Segment LED Display, Stepper Motor, Relay, Piezo Buzzer, Push Button Switch, Keyboard.

Communication Interface

UART, Parallel Interface, USB, Wi-Fi, GPRS.

Sensors and Interfacing

Sensors provide us with a means of generating signals that can be used as inputs to electronic circuits. The things that we might want to sense include physical parameters such as temperature, light level and pressure. Being able to generate an electrical signal that accurately represents these quantities allows us not only to measure and record these values but also to control them.

Sensors are, in fact, a subset of a larger family of devices known as **transducers** so we will consider these before we look at sensors and how we condition the signals that they produce in greater detail. We begin, however, with a brief introduction to the instrumentation and control systems in which sensors, transducers and signal conditioning circuits are used.

Instrumentation and Control system

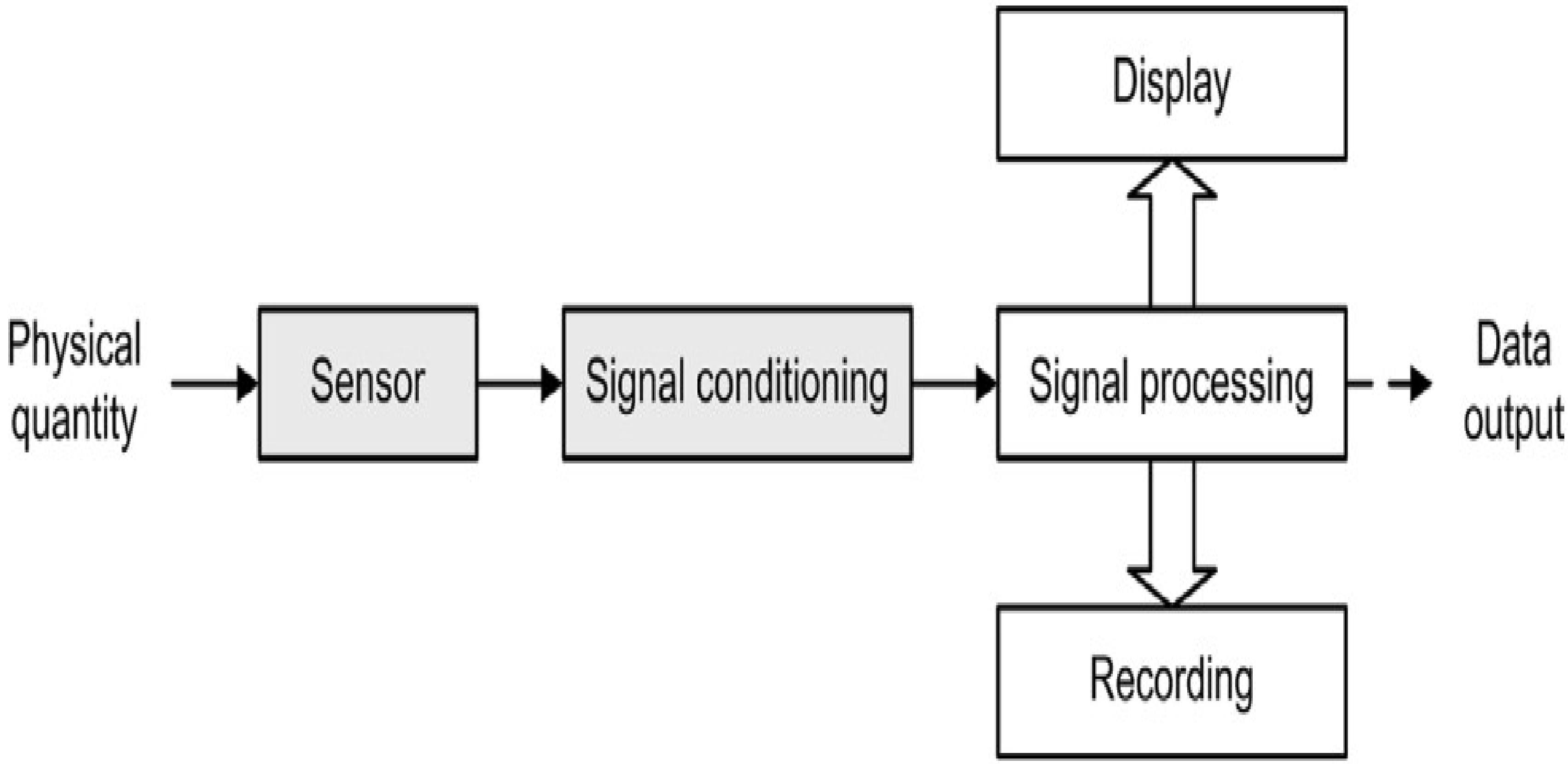
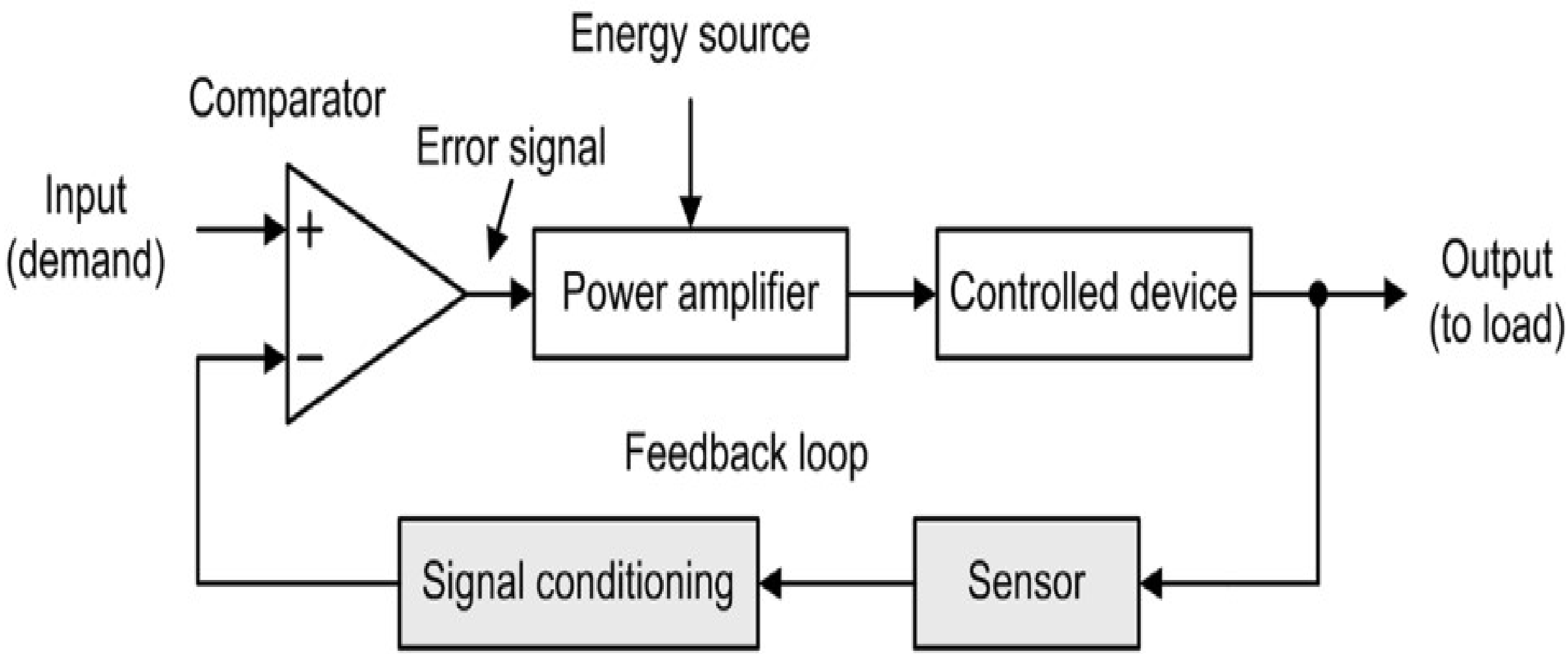


Fig. shows the arrangement of an instrumentation system.

- Figure 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 note that there may not be a linear relationship between the physical quantity and its electrical equivalent.
- Because of this and since the output produced by the sensor may be small or may suffer from the presence of noise (i.e. unwanted signals) 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.
- Furthermore, because the signal processing may use digital rather than analogue signals an additional stage of analogue-to-digital conversion may be required.

Control System



(b) A control system

- Figure (b) shows the arrangement of a control system. This 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 discrepancy is detected the 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 d.c. voltage source while the controlled device can take many forms (e.g. a d.c. motor, linear actuator, heater, etc.).

TRANSDUCERS

- Transducers are devices that convert energy in the form of sound, light, heat, etc., into an equivalent electrical signal, or vice versa.
- A loudspeaker is a transducer that converts low frequency electric current into audible sounds.
- A microphone, on the other hand, is a transducer that performs the reverse function, i.e. that of converting sound pressure variations into voltage or current.
- Transducers may be used both as inputs to electronic circuits and outputs from them.
- From the two previous examples, it should be obvious that a loudspeaker is an **output transducer** designed for use in conjunction with an audio system.
- A microphone is an **input transducer** designed for use with a recording or sound reinforcing system.

Table 15.1 Some examples of input transducers

Physical quantity	Input transducer	Notes
Sound (pressure change)	Dynamic microphone (see Fig. 15.3)	Diaphragm attached to a coil is suspended in a magnetic field. Movement of the diaphragm causes current to be induced in the coil.
Temperature	Thermocouple (see Fig. 15.2)	Small e.m.f. generated at the junction between two dissimilar metals (e.g. copper and constantan). Requires reference junction and compensated cables for accurate measurement.
Angular position	Rotary potentiometer	Fine wire resistive element is wound around a circular former. Slider attached to the control shaft makes contact with the resistive element. A stable d.c. voltage source is connected across the ends of the potentiometer. Voltage appearing at the slider will then be proportional to angular position.

Table 15.2 Some examples of output transducers

Physical quantity	Output transducer	Notes
Sound (pressure change)	Loudspeaker (see Fig. 15.3)	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.
Temperature	Heating element (resistor)	Metallic conductor is wound onto a ceramic or mica former. Current flowing in the conductor produces heat.
Angular position	Rotary potentiometer	Multi-phase motor provides precise rotation in discrete steps of 15° (24 steps per revolution), 7.5° (48 steps per revolution) and 1.8° (200 steps per revolution).

SENSORS

- A sensor is a special kind of transducer that is used to generate an input signal to a measurement, instrumentation or control system. The signal produced by a sensor is an **electrical analogy of a physical quantity, such as distance,** velocity, acceleration, temperature, pressure, light level, etc. The signals returned from a sensor, together with control inputs from the user or controller (as appropriate) will subsequently be used to determine the output from the system.
- The choice of sensor is governed by a number of factors including accuracy, resolution, cost and physical size.
- Sensors can be categorized as either **active or passive.**
- **An active sensor generates a current or** voltage output.
- A passive transducer requires a source of current or voltage and it modifies this in some way (e.g. 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 classed as either **digital or analogue.**
- **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 any one of an infinite number of voltage or current levels.

ACTUATORS

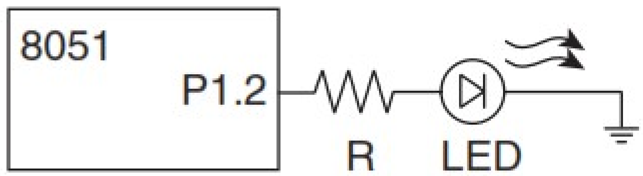
1. LED
2. 7-Segment LED Display
3. Stepper Motor
4. Relay
5. Piezo Buzzer
6. Push Button Switch
7. Keyboard

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

Eg. Micro motor actuator which adjusts the position of the cushioning element in the Smart Running shoes from adidas

Wearable Devices – certain smart watches use Ambient Light sensors to detect surrounding light intensity and adjust the screen brightness for better readability using electrical / electronic actuators.

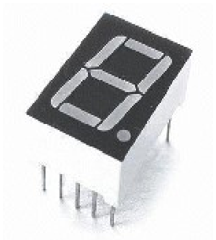
INTERFACING A LED



LED interfaced to 8051

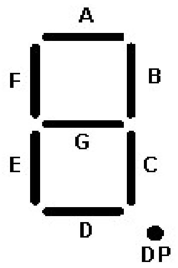
- Light Emitting Diode (LED) emits light when forward biased.
- When the port pin P1.2 goes high in Fig., the LED is forward biased and emits light.
- When the pin P1.2 goes low, LED is off.

7SEGMENT LED DISPLAY



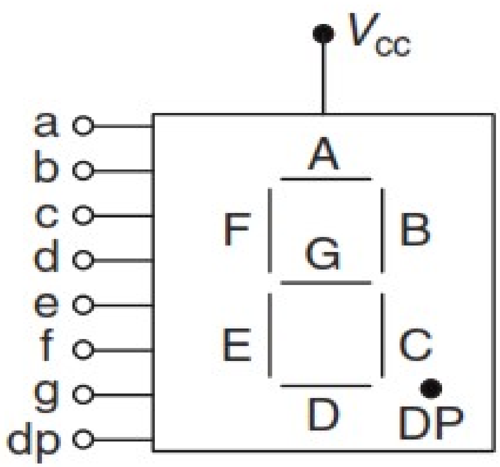
The 7 – segment LED display is an **output device** for displaying alpha numeric characters

- It contains 8 light-emitting diode (LED) segments arranged in a special l form.
- Out of the 8 LED segments, 7 are used for displaying alpha numeric characters and 1 is used for representing decimal point.
- 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; **Common anode and Common cathode**



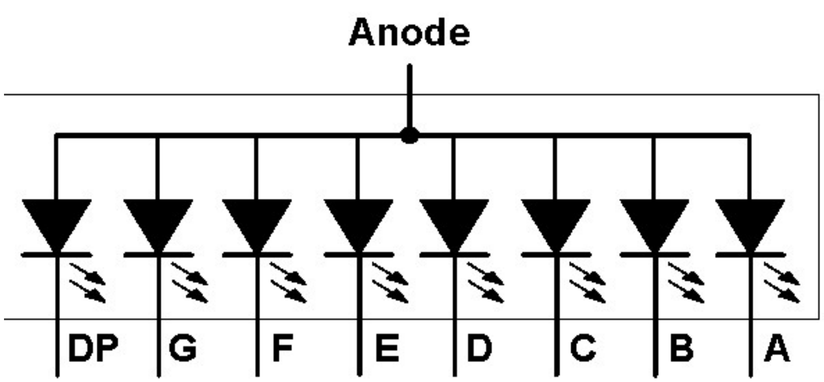
7segment display arrangement

- An array of LEDs arranged in a two-dimensional plane to display numbers (0–9 and A–F) is the 7-segment display.
- The array of LEDs with all the anodes connected together, is called **common-anode display.**
- Similarly, with all the cathodes connected together, it is called a **common-cathode display.**



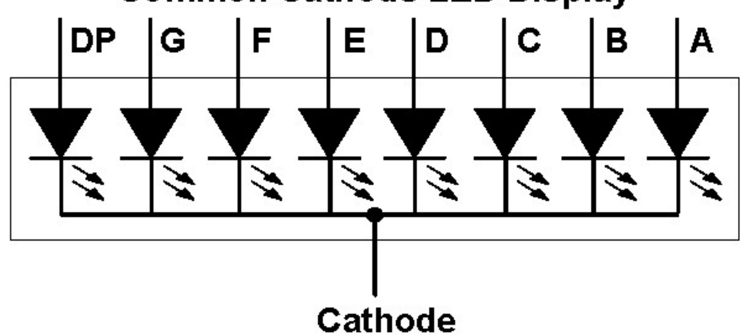
(c) Common anode 7-segment display

- In the Common anode configuration, the anodes of the 8 segments are connected commonly To display ‘0’, the inputs a, b, c, d, e, f should be made “low” to forward bias the corresponding LEDs, as the anodes are already connected to V_{cc}.
- In the Common cathode configuration, the 8 LED segments share a common cathode line (connected to ground). To display the number 0 (zero), LEDs A, B, C, D, E, F should be switched “on” and LEDs G and DP should be made “off” in the 7-segment display.

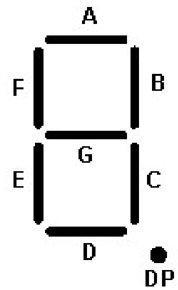


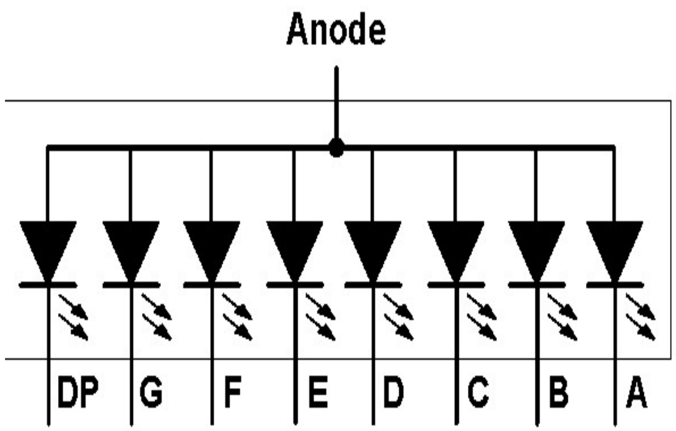
Common Anode LED Display

Common Cathode LED Display



- ***Display numbers 0–9 on a common-anode 7-segment display***





Common Anode LED Display

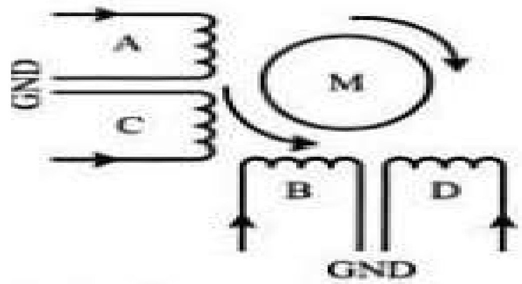
Number	Code	LED status							
		A	B	C	D	D	E	G	DP
0	03	0	0	0	0	0	0	1	1
1	9F	1	0	0	1	1	1	1	1
2	25	0	0	1	0	0	1	0	1
3	0D	0	0	0	0	1	1	0	1
4	D9	1	1	0	1	1	0	0	1
5	49	0	1	0	0	1	0	0	1
6	41	0	1	0	0	0	0	0	1
7	1F	0	0	0	1	1	1	1	1
8	01	0	0	0	0	0	0	0	1
9	19	0	0	0	1	1	0	0	1

STEPPER MOTOR

- A Stepper motor is an electro-mechanical device which generates discrete displacement (motion) in response to dc electrical signals.
- It differs from the normal dc motor in its operation.
- The dc motor produces continuous rotation on applying dc voltage whereas a stepper motor produces discrete rotation in response to the dc voltage applied to it.
- Stepper motors are widely used in industrial embedded applications, consumer electronic products and robotics control system, for position control applications (paper feed mechanism) such as dot matrix printers, disk drives, etc.
- Based on coil winding arrangements, a two phase stepper motor is classified into two types:
- 1. Unipolar
- 2. Bipolar

UNIPOLAR

- A unipolar stepper motor contains two windings per phase.
- The direction of rotation (clockwise or anticlockwise) of a stepper motor is controlled by changing the direction of current flow.
- Current in one direction flows through one coil and in the opposite direction flows through the other coil.
- It is easy to shift the direction of rotation by just switching the terminals to which the coil are connected.
- The coils are represented as A, B, C and D.
- Coils A and C carry current in opposite directions for phase 1 (only one of them will be carrying current at a time).
- Similarly, B and D carry current in opposite directions for phase 2 (only one of them will be carrying current at a time).



2-Phase Unipolar stepper motor

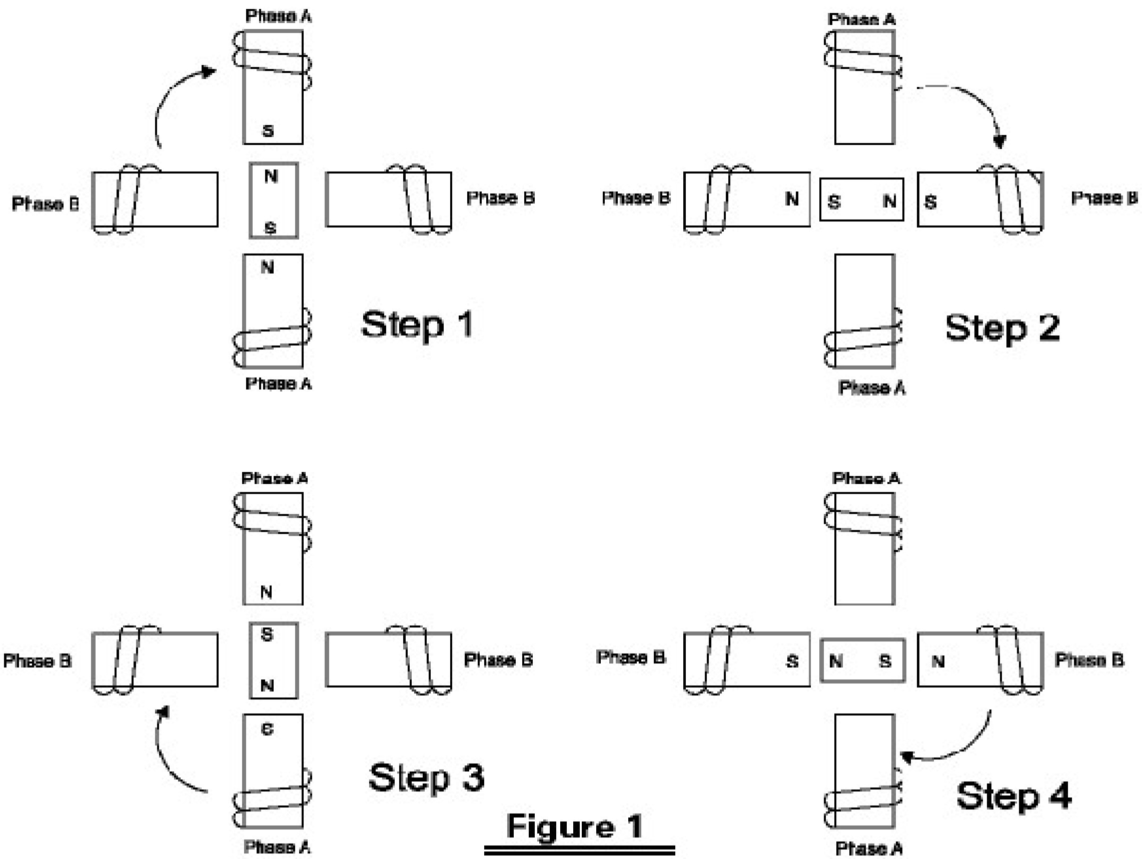

Phase A
S
N
S
Phase B
Phase B
N
Phase A
Step 1
Phase A
Phase B
N
S
N
S
Phase B
Phase A
Step 2
Phase A
N
S
N
Phase B
Phase B
S
Phase A
Step 3
Phase A
Phase B
S
N
S
N
Phase B
Phase A
Step 4

Figure 1

- **Full Step**
- In the full step mode both the phases are energised simultaneously.
- The coils A,B,C and D are energised as shown in the Table.
- It should be noted that out of the two windings, only one winding of a phase is energised at a time.

Step	Coil A	Coil B	Coil C	Coil D
1	H	H	L	L
2	L	H	H	L
3	L	L	H	H
4	H	L	L	H

- **Wave Step**
- In the wave step mode only one phase is energised at a time and each coils of the phase is energised alternatively.
- The coils A, B, C and D are energised in the following order:

Step	Coil A	Coil B	Coil C	Coil D
1	H	L	L	L
2	L	H	L	L
3	L	L	H	L
4	L	L	L	H

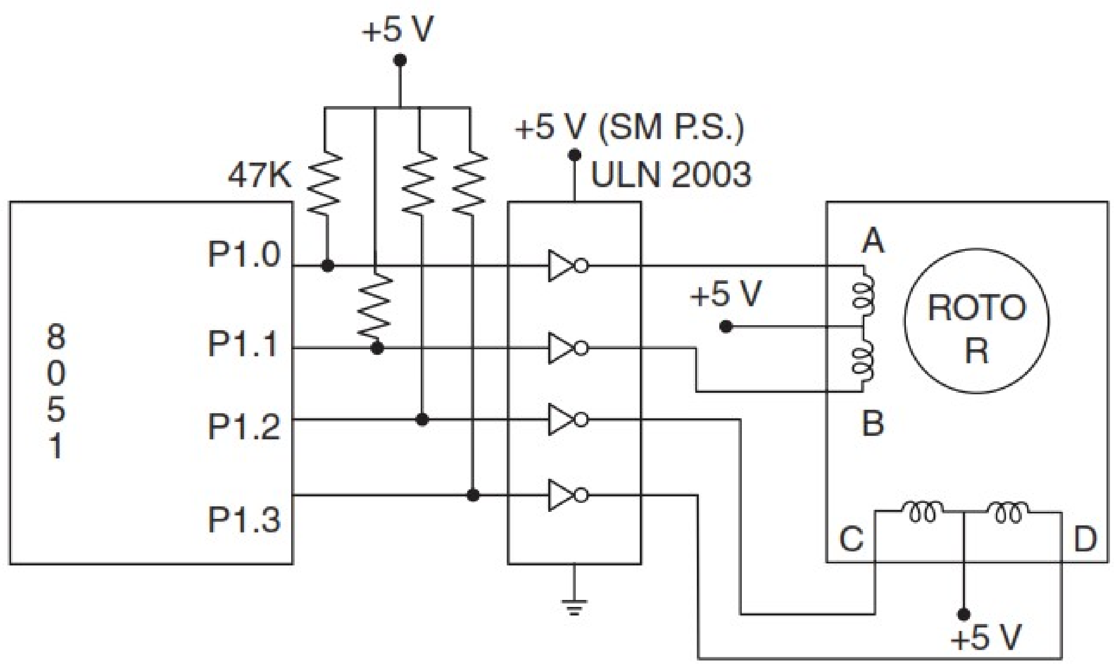
Half Step: It uses the combination of wave and full step. It has the highest torque and stability.

Step	Coil A	Coil B	Coil C	Coil D
1	H	L	L	L
2	H	H	L	L
3	L	H	L	L
4	L	H	H	L
5	L	L	H	L
6	L	L	H	H
7	L	L	L	H
8	H	L	L	H

- The rotation of the stepper motor can be reversed by reversing the order in which the coil is energised.
- Two-phase unipolar stepper motors are the popular choice for embedded applications.
- The current requirement for stepper motor is little high and hence the port pins of a microcontroller/processor may not be able to drive them directly.
- Also, the supply voltage required to operate stepper motor varies normally in the range 5V to 24 V.
- Depending on the current and voltage requirements, special driving circuits are required to interface the stepper motor with microcontroller/processors.
- Commercial off-the-shelf stepper motor driver ICs are available in the market and they can be directly interfaced to the microcontroller port.
- ULN2803 is an octal peripheral driver array available from Texas Instruments and ST microelectronics for driving a 5V stepper motor.
- Simple driving circuit can also be built using transistors

- The following circuit diagram illustrates the interfacing of a stepper motor through a driver circuit connected to the port pins

  microcontroller/processor for interfacing the stepper motor to 8051, we need to connect the four winding leads to four-port pins, say, P1.0 to P1.3. Since the port pins do not have the sufficient current, to drive the stepper motor windings (needs > 10 mA) a driver such as ULN 2003 is used.



RELAY

- An electromechanical device
- In an embedded application, the 'Relay Unit' acts as dynamic path selectors for signals and powe
- The 'Relay' unit contains a relay coil made up of insulated wire on a metal core and a metal armature with one or more contacts.

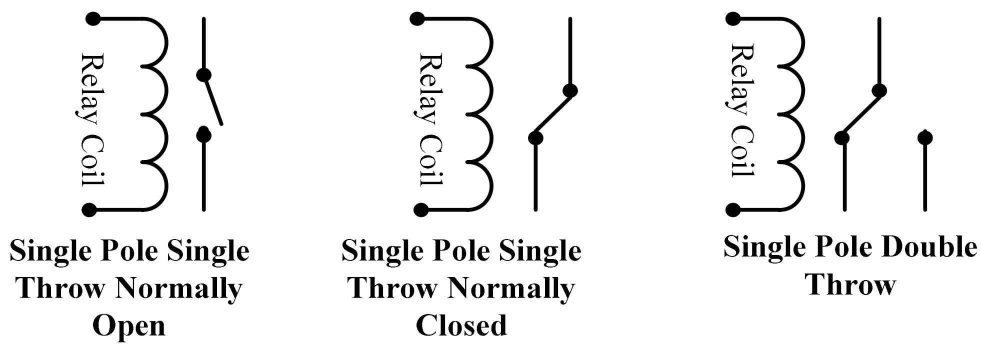


FIG: Relay configurations

- Relay’ has a relay coil on a metal core and a metal armature with one or more contacts.
- ‘Relay’ works on electromagnetic principle.
- When a voltage is applied to the relay coil, current flows through the coil, which in turn generates a magnetic field. The magnetic field attracts the armature core and moves the contact point. The movement of the contact point changes the power/signal flow path.
- Single Pole Single Throw configuration has only one path for information flow.
- The path is either open or closed in normal condition.
- For Normally Open SPST relay, circuit is normally open and it becomes closed when relay is energized. Vice versa for NC SPST relay.
- Single Pole Double throw relay, there are two paths for information flow & they are selected by energizing and de-energizing the relay.

RELAY DRIVER CIRCUIT

- The Relay is normally controlled using a relay driver circuit connected to the port pin of the processor/controller.
- A transistor can be used as the relay driver. The transistor can be selected depending on the relay driving current requirements.
- A free-wheeling diode – to protect the relay & transistor – used to free-wheel the voltage produced in the opposite direction when the relay coil is de-energized.

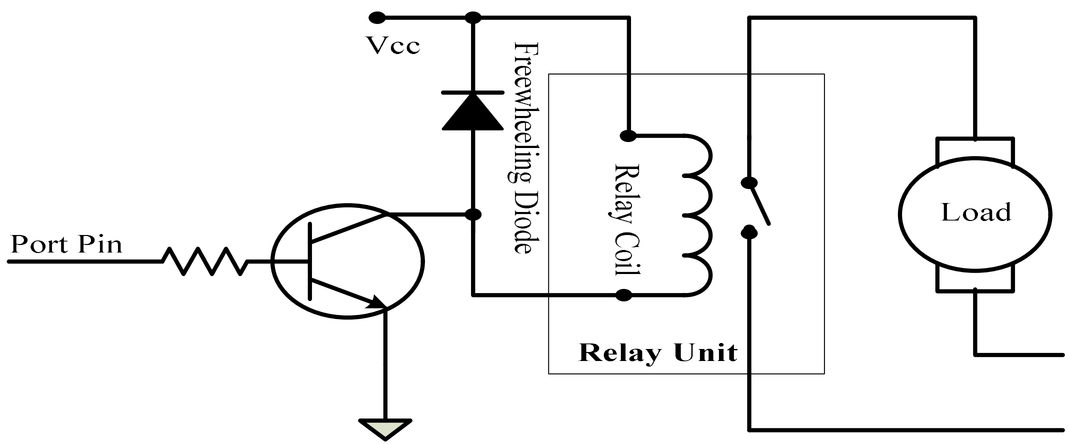


Figure:Transistor based relay driving circuit

PIEZO BUZZER

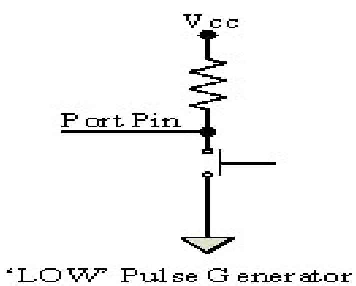
- Piezo buzzer is a piezoelectric device for generating audio indications in embedded application.
- A piezoelectric buzzer contains a piezoelectric diaphragm which produces audible sound in response to the voltage applied to it.
- Piezoelectric buzzers are available in two types:

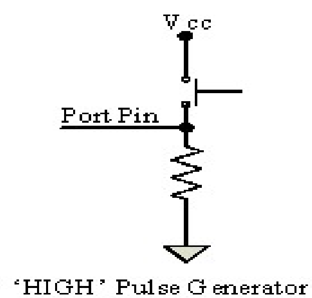
1. Self driving
2. External driving

- The 'Self-driving' circuit contains all the necessary components to generate sound at a predefined tone.
- It will generate a tone on applying the voltage.
- External driving piezo buzzers supports the generation of different tones.
- The tone can be varied by applying a variable pulse train to the piezoelectric buzzer.
- A piezo buzzer can be directly interfaced to the port pin of the processor / control.

PUSH BUTTON SWITCH

- Push Button switch is an input device.
- Push button switch comes in two configurations, namely ‘Push to Make’ and ‘Push to Break’.
- The switch is normally in the open state and it makes a circuit contact when it is pushed or pressed in the ‘Push to Make’ configuration.
- In the ‘Push to Break’ configuration, the switch is normally in the closed state and it breaks the circuit contact when it is pushed or pressed.
- The push button stays in the ‘closed’ (For Push to Make type) or ‘open’ (For Push to Break type) state as long as it is kept in the pushed state and it breaks/makes the circuit connection when it is release.
- Push button is used for generating a momentary pulse.
- In embedded application push button is generally used as reset and start switch and pulse generator.
- The Push button is normally connected to the port pin of the host processor





KEYBOARD

- Keyboard is an input device for user interfacing.
- If the number of keys required is very limited, push button switches can be used and they can be directly interfaced to the port pins for reading.
- Matrix keyboard is an optimum solution for handling large key requirements. It greatly reduces the number of interface connections.
- Ex: For interfacing 16 keys, in the direct interfacing technique 16 port pins are required, whereas in the matrix keyboard only 8 lines are required. The 16 Keys are arranged in a 4 column, 4 Rows matrix.
- Fig illustrates the connection of keys in a matrix keyboard.

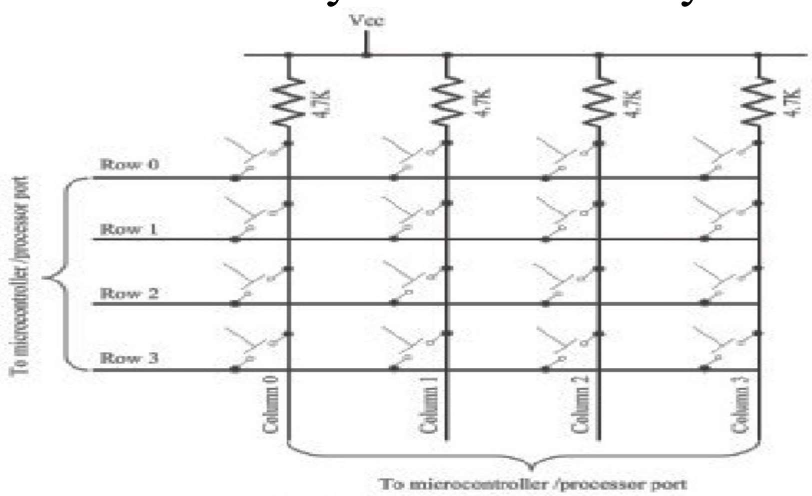
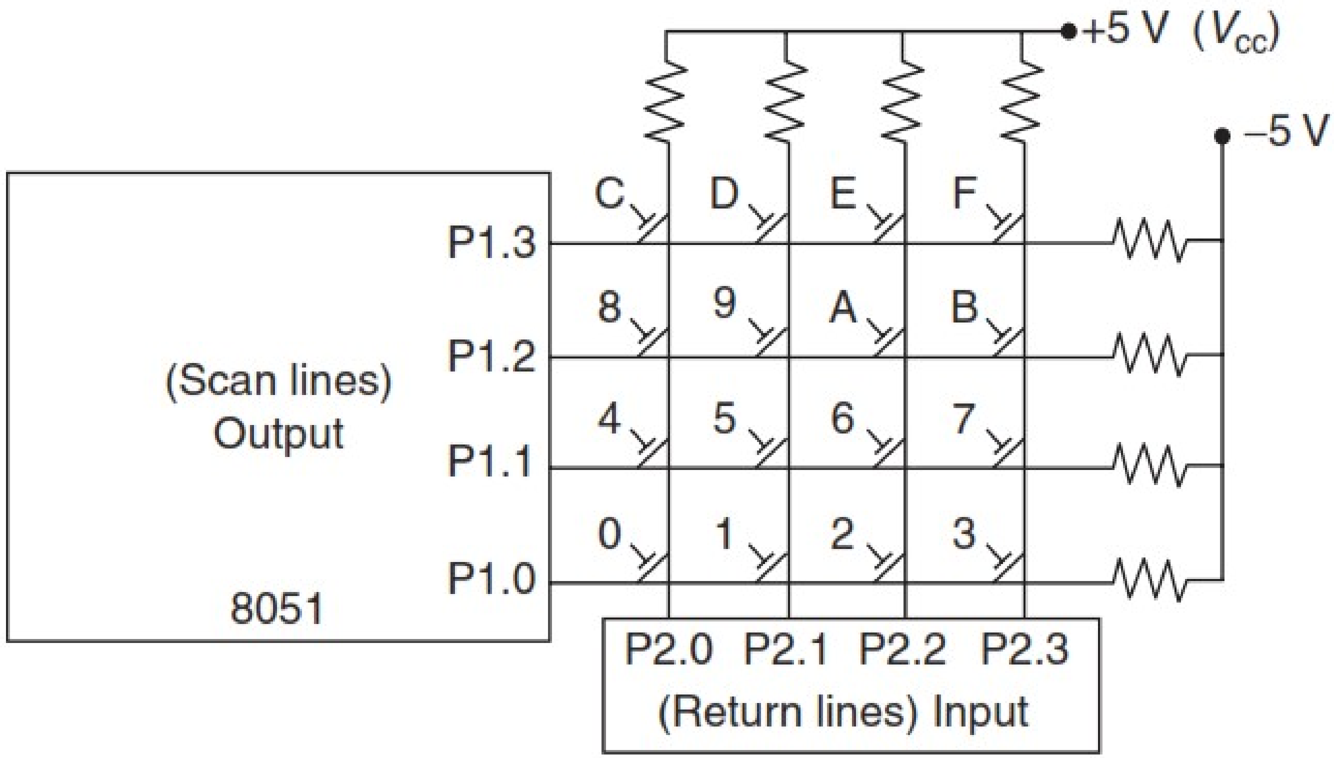


Fig. 2.24 Matrix keyboard Interfacing

Interfacing Keyboard / Matrix Keypad to 8051



- In a matrix keyboard, the keys are arranged in matrix fashion (i.e., they are connected in a row and column style).
- For detecting a key press, the keyboard uses the scanning technique, where each row of the matrix is pulled low and the columns are read.
- After reading the status of each columns corresponding to a row, the row is pulled high & the next row is pulled low and the status of the columns are read.
- This process is repeated until the scanning for all rows are completed.
- Since keys are mechanical devices, there is a possibility for de-bounce issues, which may give multiple key press effect for a single key press.
- To prevent this, a proper key de-bouncing technique should be applied.
- Hardware key de-bouncer circuits and software key de-bounce techniques are the key de-bouncing techniques available.
- The software key de-bouncing technique doesn't require any additional hardware and is easy to implement.
- In the software de-bouncing technique, on detecting a key press, the key is read again after a de-bounce delay.

COMMUNICATION INTERFACE

- Communication interface is essential for communicating with various subsystems of the embedded system and with the external world.
- For an embedded product, the communication interface can be viewed in two different perspectives; namely;

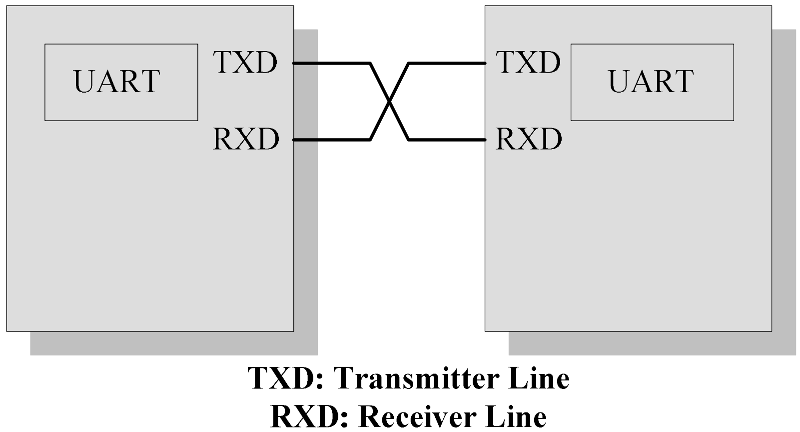
1. Device/board level communication interface (Onboard Communication Interface)
2. Product level communication interface (External Communication Interface)

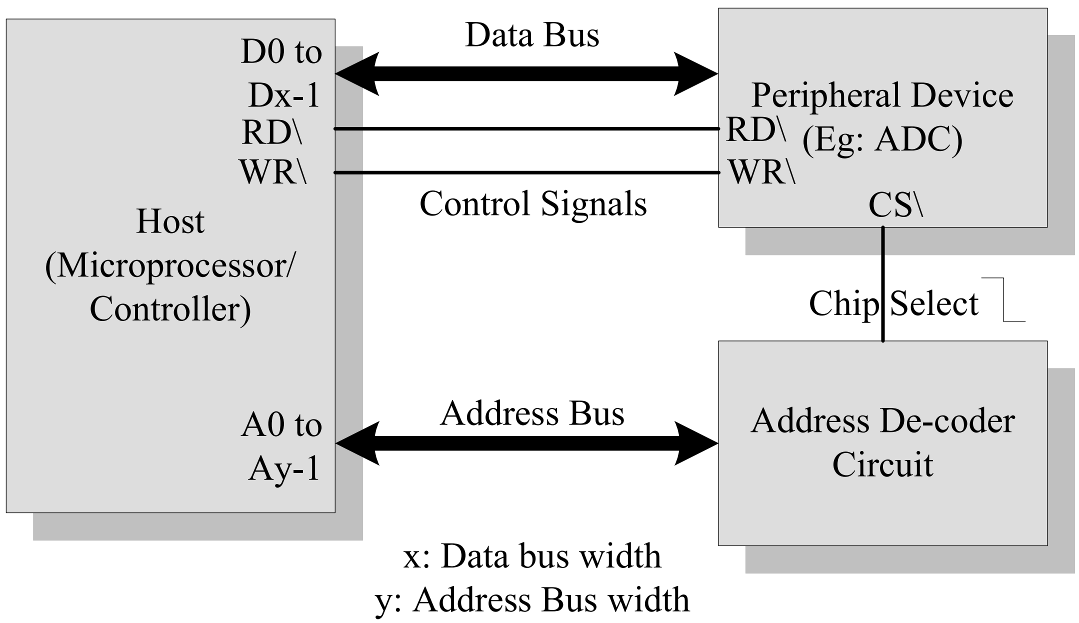
- Embedded product is a combination of different types of components (chips/devices) arranged on a Printed Circuit Board (PCB).
- The communication channel which interconnects the various components within an embedded product is referred as Device/board level communication interface (Onboard Communication Interface).
- Serial interfaces like I2C, SPI, UART, 1-Wire etc and Parallel bus interface are examples of 'Onboard Communication Interface'

- The ‘Product level communication interface’ (External Communication Interface) is responsible for data transfer between the embedded system and other devices or modules.
- The external communication interface can be either wired media or wireless media and it can be a serial or parallel interface. Infrared (IR), Bluetooth (BT), Wireless LAN (Wi-Fi), Radio Frequency waves (RF), GPRS etc are examples for wireless communication interface.
- RS-232C/RS-422/RS 485, USB, Ethernet (TCP-IP), IEEE 1394 port, Parallel port, CF-II Slot, SDIO, PCMCIA etc are examples for wired interfaces.
- Mobile Communication Equipment – an example of an embedded system with external communication interface

On-board Communication Interface – Universal Asynchronous Receiver Transmitter (UART)

- UART based data transmission is an asynchronous form of serial data transmission.
- The serial communication settings (Baudrate, No. of bits per byte, parity, No. of start bits and stop bit and flow control) for both transmitter and receiver should be set as identical.
- The start and stop of communication is indicated through inserting special bits in the data stream.
- While sending a byte of data, a start bit is added first and a stop bit is added at the end of the bit stream. The least significant bit of the data byte follows the start bit.
- The ‘Start’ bit informs the receiver that a data byte is about to arrive. The receiver device starts polling its ‘receive line’ as per the baudrate settings..
- If parity is enabled for communication, the UART of the transmitting device adds a parity bit .
- The UART of the receiving device calculates the parity of the bits received and compares it with the received parity bit for error checking.
- The UART of the receiving device discards the ‘Start’, ‘Stop’ and ‘Parity’ bit from the received bit stream and converts the received serial bit data to a word.





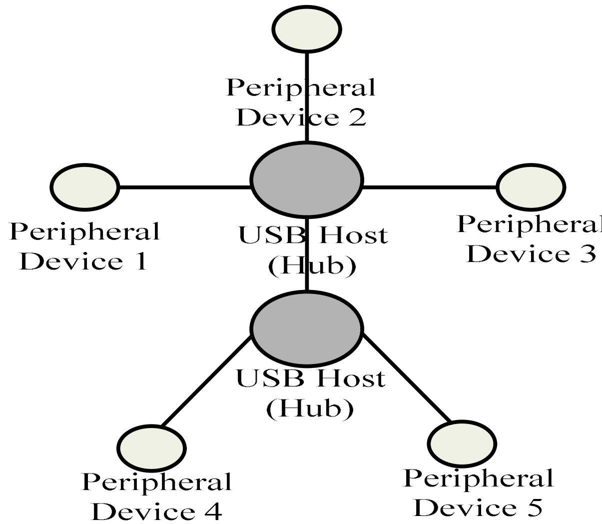
On-board Communication Interface

Parallel Interface

- Parallel interface is normally used for communicating with peripheral devices which are memory mapped to the host of the system.
- The host processor/controller of the embedded system contains a parallel bus and the device which supports parallel bus can directly connect to this bus system.
- The communication through the parallel bus is controlled by the control signal interface between the device and the host.
- The ‘Control Signals’ for communication includes ‘Read/Write’ signal and device select signal.
- The device normally contains a device select line and the device becomes active only when this line is asserted by the host processor.
- The direction of data transfer (Host to Device or Device to Host) can be controlled through the control signal lines for ‘Read’ and ‘Write’. Only the host processor has control over the ‘Read’ and ‘Write’ control signals

External Communication Interface – Universal Serial Bus (USB)

- Universal Serial Bus (USB) is a wired high speed serial bus for data communication.
- The USB communication system follows a star topology with a USB host at the center and one or more USB peripheral devices/USB hosts connected to it.
- A USB host can support connections up to 127, including slave peripheral devices and other USB hosts.
- USB transmits data in packet format. Each data packet has a standard format. The USB communication is a host initiated one.
- The USB Host contains a host controller which is responsible for controlling the data communication, including establishing connectivity with USB slave devices, packetizing and formatting the data packet.
- There are different standards for implementing the USB Host Control interface; namely Open Host Control Interface (OHCI) and Universal Host Control Interface (UHCI)



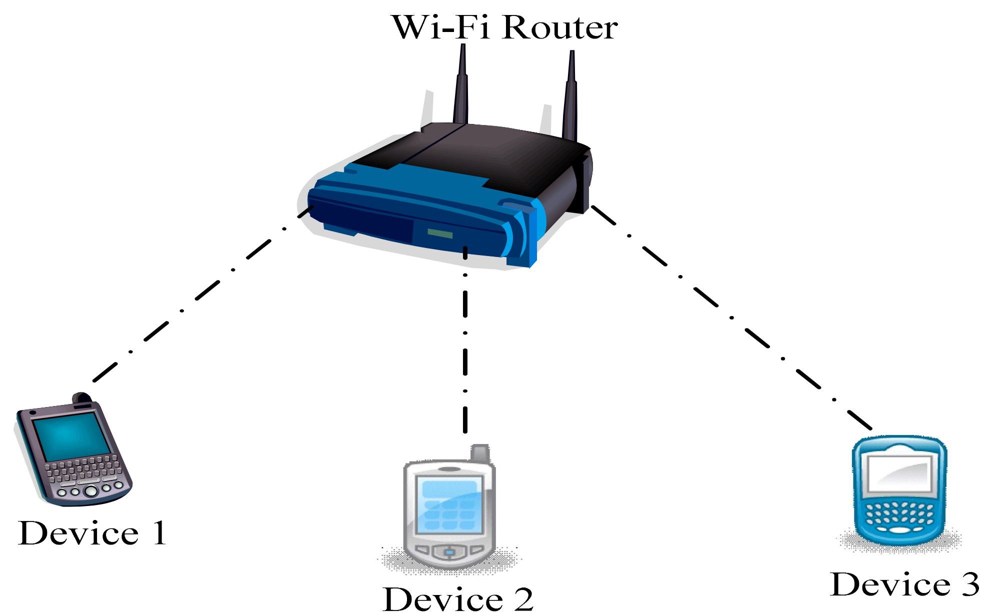
Pin No:	Pin Name	Description
1	V_{BUS}	Carries power (5V)
2	D-	Differential data carrier line
3	D+	Differential data carrier line
4	GND	Ground signal line

- The Physical connection between a USB peripheral device and master device is established with a USB cable.
- The USB cable supports communication distance of up to 5 meters.
- The USB standard uses two different types of connectors namely ‘Type A’ and ‘Type B’ at the ends of the USB cable for connecting the USB peripheral device and host device.
- ‘Type A’ connector is used for upstream connection (connection with host) and ‘Type B’ connector is used for downstream connection (connection with slave device)
- Each USB device contains a Product ID (PID) and a Vendor ID (VID).
- The PID and VID are embedded into the USB chip by the USB device manufacturer.
- The VID for a device is supplied by the USB standards forum.
- PID and VID are essential for loading the drivers corresponding to a USB device for communication.
- USB supports four different types of data transfers, namely; Control, Bulk, Isochronous and Interrupt.
- Control transfer is used by USB system software to query, configure and issue commands to the USB device

- Bulk transfer is used for sending a block of data to a device. Bulk transfer supports error checking and correction. Transferring data to a printer is an example for bulk transfer.
- Isochronous data transfer is used for real time data communication. In Isochronous transfer, data is transmitted as streams in real time. Isochronous transfer doesn't support error checking and re-transmission of data in case of any transmission loss.
- Interrupt transfer is used for transferring small amount of data. Interrupt transfer mechanism makes use of polling technique to see whether the USB device has any data to send.
- The frequency of polling is determined by the USB device and it varies from 1 to 255 milliseconds. Devices like Mouse and Keyboard, which transmits fewer amounts of data, uses Interrupt transfer.

External Communication Interface – Wi-Fi

- Popular wireless communication technique for networked communication of devices. Wi-Fi follows the IEEE 802.11 standard.
- Wi-Fi is intended for network communication and it supports Internet Protocol (IP) based communication – each device identified an IP address – unique to each device on the network. (Required – device identities in a multipoint communication to address specific devices).
- Wi-Fi based communications require an intermediate agent called Wi-Fi router/ Wireless Access point to manage the communications.
- The Wi-Fi router is responsible for
  - ✓ restricting the access to a network,
  - ✓ assigning IP address to devices on the network,
  - ✓ routing data packets to the intended devices on the network.


Wi-Fi Router
Device 1
Device 2
Device 3

- Wi-Fi enabled devices contain a wireless adaptor for transmitting and receiving data in the form of radio signals through an antenna.
- Wi-Fi operates at 2.4GHZ or 5GHZ of radio spectrum and they co-exist with other ISM band devices like Bluetooth.
- A Wi-Fi network is identified with a Service Set Identifier (SSID). A Wi-Fi device can connect to a network by selecting the SSID of the network
- Wi-Fi networks implements different security mechanisms for authentication and data transfer.
- Wireless Equivalency Protocol (WEP), Wireless Protected Access (WPA) etc are some of the security mechanisms supported by Wi-Fi networks in data communication

- For communicating with devices over a Wi-Fi network,
  - the device when its Wi-Fi radio is turned ON,
  - searches the available Wi Fi network in its vicinity and
  - lists out the Service Set Identifier (SSID) of the available networks.
- If the network is security enabled,
  - a password may be required to connect to a particular SSID.
- for securing the data communication, Wi-Fi employs different security mechanisms like
  - Wired Equivalency Privacy (WEP)
  - Wireless Protected Access (WPA), etc.
- Wi-Fi supports data rates ranging from 1 Mbps to 1300Mbps (Growing towards higher rates as technology progresses), depending on the standards (802.11a/b/g/n/ac) and access / modulation method.
- Depending on the type of antenna and usage location (indoor / outdoor), Wi-Fi offers a range of 100 to 1000 feet

External Communication Interface – General Packet Radio Service (GPRS), 3G, 4G, LTE

- A communication technique for transferring data over a mobile communication network like GSM & CDMA.
- Data is sent as packets. The transmitting device splits the data into several related packets. At the receiving end the data is re-constructed by combining the received data packets.
- GPRS supports a theoretical maximum transfer rate of 171.2kbps.
- In GPRS communication, the radio channel is concurrently shared between several users instead of dedicating a radio channel to a cell phone user. The GPRS communication divides the channel into 8 timeslots and transmits data over the available channel.
- GPRS supports Internet Protocol (IP), Point to Point Protocol (PPP) and X.25 protocols for communication.

- GPRS is mainly used by mobile enabled embedded devices for data communication. The device should support the necessary GPRS hardware like GPRS modem and GPRS radio. Also, the carrier network should support GPRS communication.
- GPRS is an old technology and it is being replaced by new generation data communication techniques like 3G, High Speed Downlink Packet Access (HSDPA), 4G, LTE, etc which offers higher bandwidths for communication
- 3G – data rates – 144Kbps to 2Mbps or higher.
- 4G – 2 to 100+ Mbps depending on network & underlying technology.