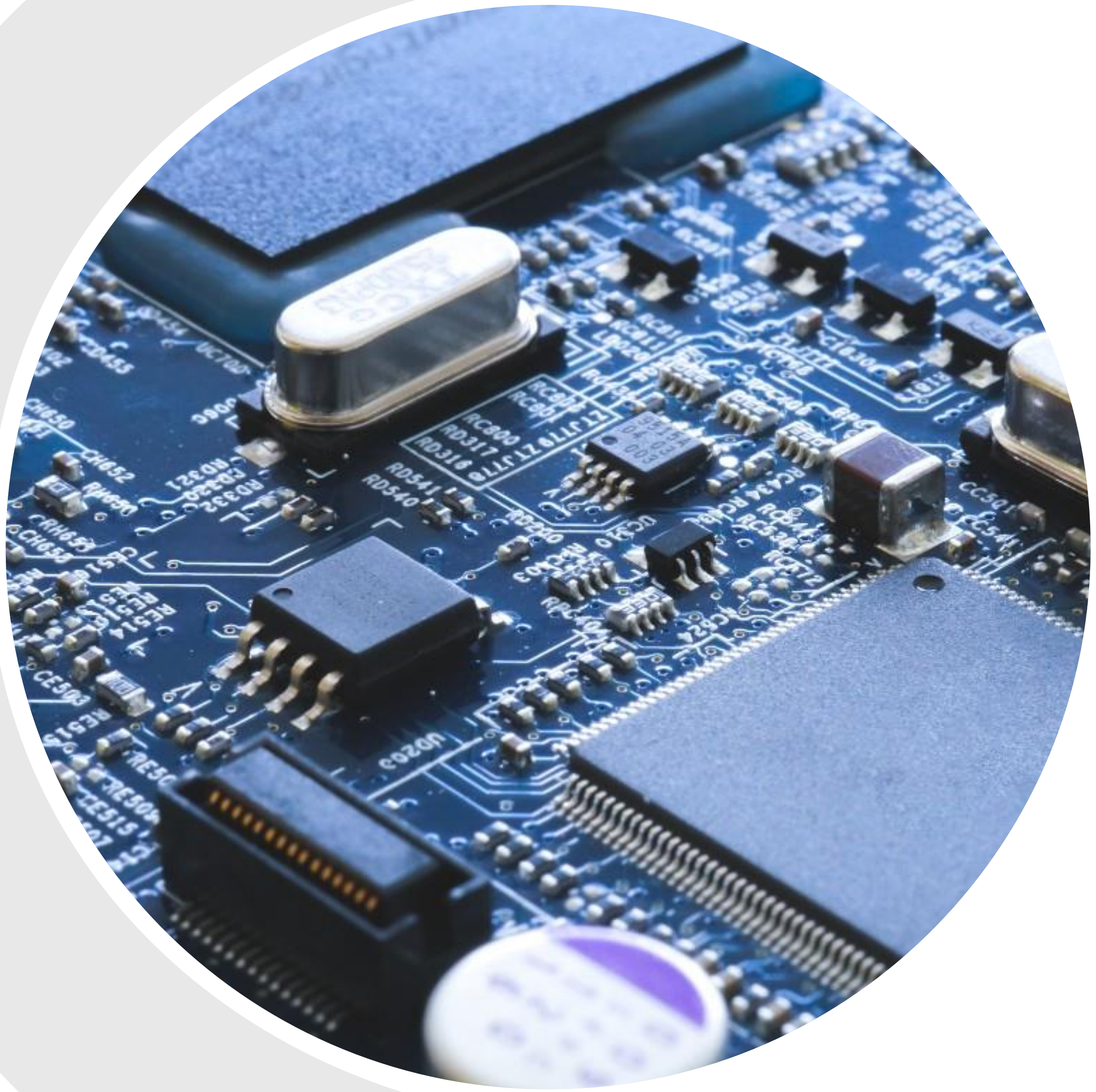


ANJUMAN
Institute of Technology
and Management

Bhatkal, Karnataka, India

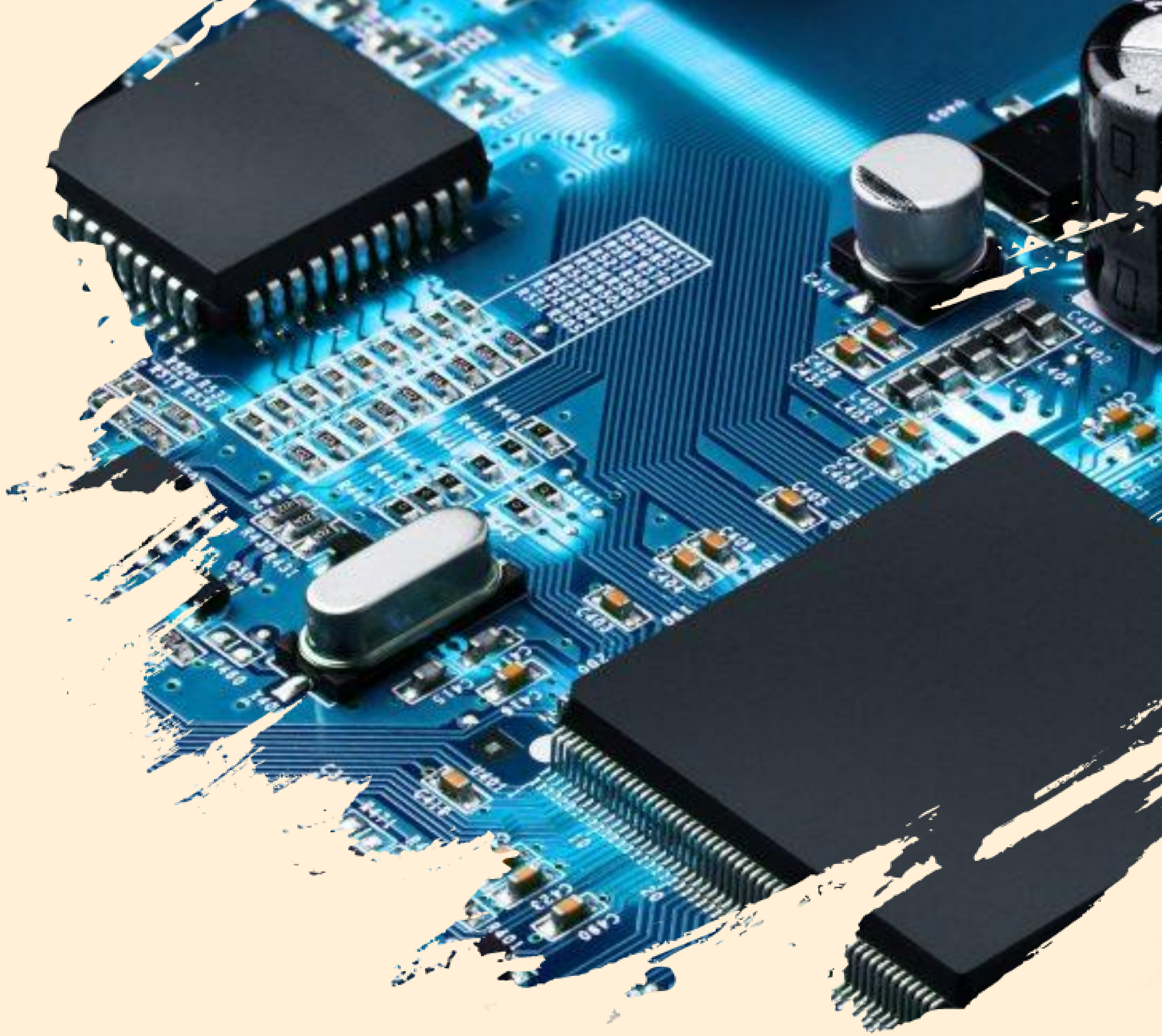
BASIC ELECTRONICS & COMMUNICATION ENGINEERING

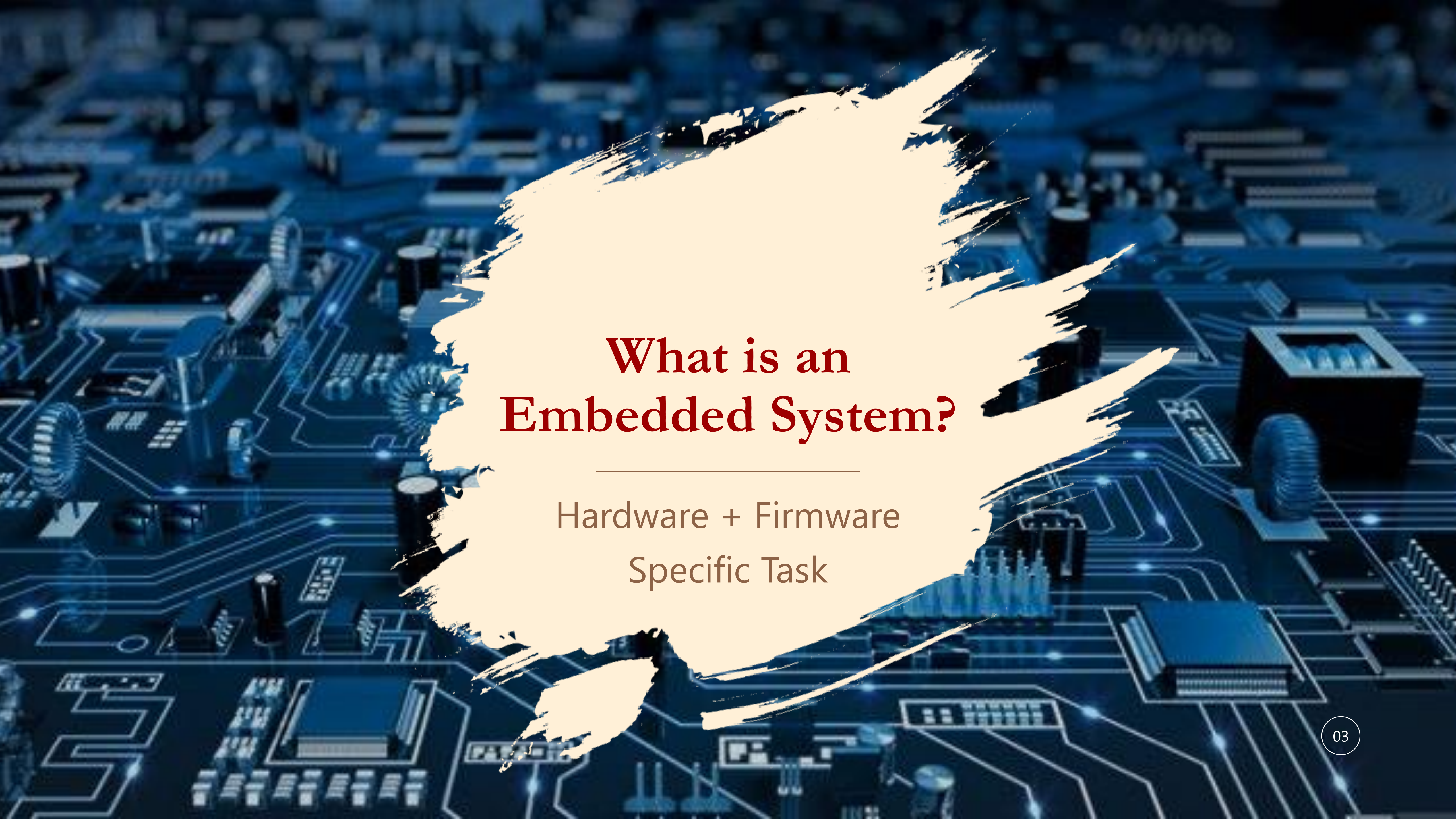
(21ELN14/21ELN24)

Mr. Shrishail Bhat
Assistant Professor
Department of Electronics and Communication Engineering

Module 3

EMBEDDED SYSTEMS





What is an Embedded System?

Hardware + Firmware

Specific Task

Where do we use Embedded Systems?

SYLLABUS

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, 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, Wi-Fi, GPRS.

Text Books

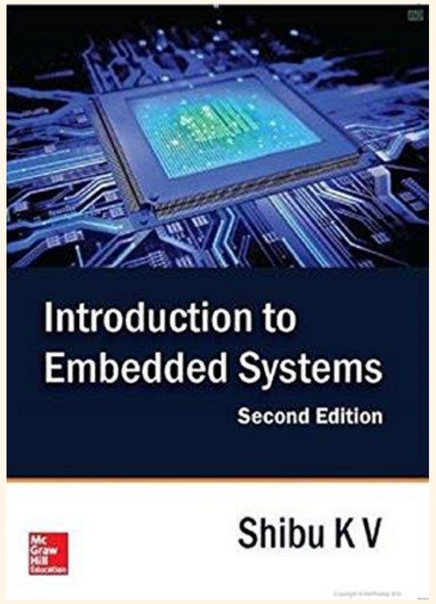
1

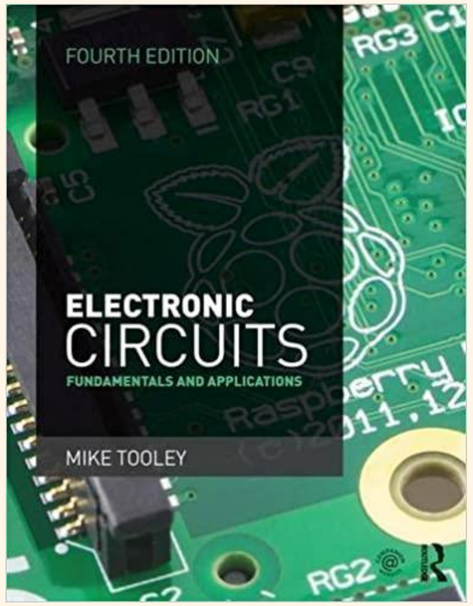
Introduction to Embedded Systems

Shibu K V

Tata McGraw Hill Education Private Limited

2nd Edition, 2017





2

Electronic Circuits – Fundamentals & Applications

Mike Tooley

Elsevier

4th Edition, 2015

Introduction

What is an Embedded System?

- An embedded system is an electronic/electro-mechanical system designed to perform a specific function and is a combination of both hardware and firmware (software).
- Every embedded system is unique and the hardware as well as the firmware is highly specialised to the application domain.

Embedded Systems vs. General Computing Systems

- The computing revolution began with the general purpose computing requirements. Later it was realised that the general computing requirements are not sufficient for the embedded computing requirements.
- The embedded computing requirements demand ‘something special’ in terms of response to stimuli, meeting the computational deadlines, power efficiency, limited memory capability, etc.

General Purpose Computing System	Embedded System
A system which is a combination of a generic hardware and a General Purpose Operating System for executing a variety of applications	A system which is a combination of special purpose hardware and embedded OS for executing a specific set of applications
Contains a General Purpose Operating System (GPOS)	May or may not contain an operating system for functioning
Applications are alterable (programmable) by the user (It is possible for the end user to re-install the operating system, and also add or remove user applications)	The firmware of the embedded system is pre-programmed and it is non-alterable by the end-user (There may be exceptions for system supporting OS kernel image flashing through special hardware settings)
Performance is the key deciding factor in the selection of the system. Always, ‘Faster is Better’	Application-specific requirements (like performance, power requirements, memory usage, 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 the hardware and the operating system
Response requirements are not time-critical	For certain category of embedded systems like mission critical systems, the response time requirement is highly critical
Need not be deterministic in execution behaviour	Execution behaviour is deterministic for certain types of embedded systems like ‘Hard Real Time’ systems

Classification of Embedded Systems

- Some of 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
- Second Generation
- Third Generation
- Fourth Generation
- Next Generation

Classification Based on Generation (continued)

- First Generation
  - Early embedded systems were built around 8-bit microprocessors like 8085 and Z80 and 4-bit microcontrollers.
  - Simple in hardware circuits with firmware developed in assembly code.
  - E.g.: Digital telephone keypads, stepper motor control units, etc.

Classification Based on Generation (continued)

- Second Generation
  - Embedded systems built around 16-bit microprocessors and 8-bit or 16-bit microcontrollers.
  - Instruction set were much more complex and powerful than the first generation.
  - Some of the second generation embedded systems contained embedded operating systems for their operation.
  - E.g.: Data acquisition systems, SCADA systems, etc.

Classification Based on Generation (continued)

- Third Generation
  - Embedded systems built around 32-bit microprocessors and 16-bit microcontrollers.
  - Application and domain specific processors/controllers like Digital Signal Processors (DSP) and Application Specific Integrated Circuits (ASICs) came into picture.
  - The instruction set of processors became more complex and powerful and the concept of instruction pipelining also evolved.
  - Dedicated embedded real time and general purpose operating systems entered into the embedded market.
  - Embedded systems spread its ground to areas like robotics, media, industrial process control, networking, etc.

Classification Based on Generation (continued)

- Fourth Generation
  - The advent of System on Chips (SoC), reconfigurable processors and multicore processors are bringing high performance, tight integration and miniaturisation into the embedded device market.
  - The SoC technique implements a total system on a chip by implementing different functionalities with a processor core on an integrated circuit.
  - They make use of high performance real time embedded operating systems for their functioning.
  - E.g.: Smart phone devices, Mobile Internet Devices (MIDs), etc.

Classification Based on Generation (continued)

- Next Generation
  - The processor and embedded market is highly dynamic and demanding.
  - The next generation embedded systems are expected to meet growing demands in the market.

Classification Based on Complexity and Performance

- Small-Scale Embedded Systems
- Medium-Scale Embedded Systems
- Large-Scale Embedded Systems/Complex Systems

Classification Based on Complexity and Performance (continued)

- Small-Scale Embedded Systems
    - Simple in application needs and the performance requirements are not time critical.
    - E.g.: An electronic toy
    - Usually built around low performance and low cost 8-bit or 16-bit microprocessors/microcontrollers.
    - May or may not contain an operating system for its functioning.

Classification Based on Complexity and Performance (continued)

- Medium-Scale Embedded Systems
  - Slightly complex in hardware and firmware (software) requirements.
  - Usually built around medium performance, low cost 16-bit or 32-bit microprocessors/microcontrollers or digital signal processors.
  - Usually contain an embedded operating system (either general purpose or real time operating system) for functioning.

Classification Based on Complexity and Performance (continued)

- Large-Scale Embedded Systems/Complex Systems
  - Highly complex in hardware and firmware (software) requirements.
  - They are employed in mission critical applications demanding high performance.
  - Usually built around high performance 32-bit or 64-bit RISC processors/controllers or Reconfigurable System on Chip (RSoC) or multi-core processors and programmable logic devices.
  - May contain multiple processors/controllers and co-units/hardware accelerators for offloading the processing requirements from the main processor of the system.
  - Decoding/encoding of media, cryptographic function implementation, etc. are examples of processing requirements which can be implemented using a co-processor/hardware accelerator.
  - Usually contain a high performance real time operating system (RTOS) for task scheduling, prioritization and management.

Classification Based on Deterministic Behaviour

- Applicable for ‘Real Time’ systems.
- The application/task execution behaviour can be either deterministic or non-deterministic.
- Based on the execution behaviour, real time embedded systems are classified into Hard Real Time and Soft Real Time systems.

Classification Based on Triggering

- Embedded systems which are ‘Reactive’ in nature (like process control systems in industrial control applications) can be classified based on the trigger.
- Reactive systems can be either event-triggered or time-triggered.

Major Application Areas of Embedded Systems

1. Consumer electronics: Camcorders, cameras, etc.
2. Household appliances: Television, DVD players, washing machine, refrigerators, microwave oven, etc.
3. Home automation and security systems: Air conditioners, sprinklers, intruder detection alarms, closed circuit television (CCTV) cameras, fire alarms, etc.
4. Automotive industry: Anti-lock braking systems (ABS), engine control, ignition systems, automatic navigation systems, etc.
5. Telecom: Cellular telephones, telephone switches, handset multimedia applications, etc.

Major Application Areas of Embedded Systems (continued)

6. Computer peripherals: Printers, scanners, fax machines, etc.
7. Computer networking systems: Network routers, switches, hubs, firewalls, etc.
8. Healthcare: Different kinds of scanners, EEG, ECG machines, etc.
9. Measurements & Instrumentation: Digital multimeters, digital CROs, logic analyzers, PLC systems, etc.
10. Banking & Retail: Automated teller machines (ATM) and currency counters, point of sales (POS), etc.
11. Card readers: Barcode, smart card readers, hand held devices, etc.

Major Application Areas of Embedded Systems (continued)

12. Wearable Devices: Health and fitness trackers, Smartphone screen extension for notifications, etc.
13. Cloud Computing and Internet of Things (IoT)

The Typical Embedded System

ELEMENTS OF AN EMBEDDED SYSTEM

The Typical Embedded System

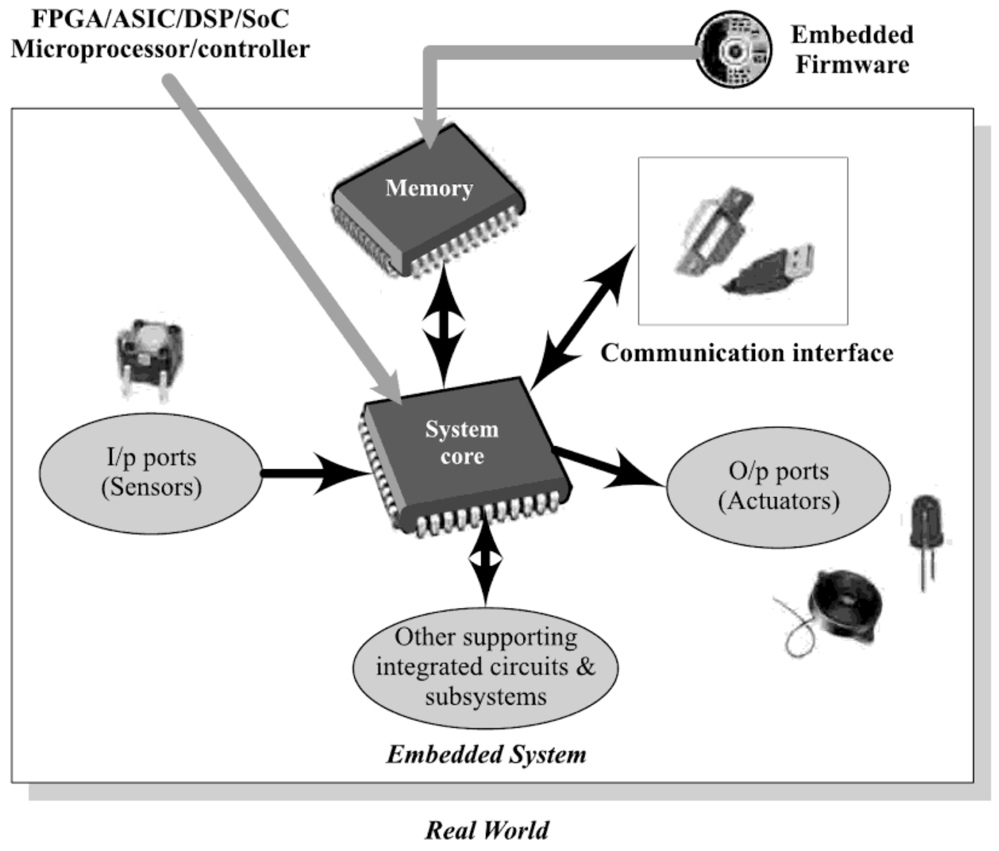


Fig: Elements of an Embedded System

The Typical Embedded System (continued)

- It contains a single chip controller, which acts as the master brain of the system.
- The controller can be
  - ✓ A microprocessor or
  - ✓ A microcontroller or
  - ✓ A Field Programmable Gate Array (FPGA) device or
  - ✓ A Digital Signal Processor (DSP) or
  - ✓ An Application Specific Integrated Circuit (ASIC)/Application Specific Standard Product (ASSP)

The Typical Embedded System (continued)

- An embedded system 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.
- Key boards, push button switches, etc. are examples for common user interface input devices.
- LEDs, liquid crystal displays, piezoelectric buzzers, etc. are examples for common user interface output devices for a typical embedded system.

The Typical Embedded System (continued)

- The memory of the system is responsible for holding the control algorithm and other important configuration details.
- For most of embedded systems, the memory for storing the algorithm or configuration data is of fixed type, which is a kind of Read Only Memory (ROM).
  - It is not available for the end user for modifications
  - The memory is protected from unwanted user interaction by implementing some kind of memory protection mechanism.
  - The most common types of memories used in embedded systems for control algorithm storage are OTP, PROM, UVEPROM, EEPROM and FLASH.
- Sometimes the system requires temporary memory for performing arithmetic operations or control algorithm execution and this type of memory is known as "working memory".
  - Random Access Memory (RAM) is used in most of the systems as the working memory.
  - Various types of RAM like SRAM, DRAM and NVRAM are used for this purpose.

The Typical Embedded System (continued)

- Apart from these, communication interface is essential for communicating with various subsystems of the embedded system and with the external world.
- The communication interfaces may be used to achieve onboard (I2C, SPI, UART, parallel bus interface, etc.) or external communication (wireless interfaces like Infrared, Bluetooth, Wi-Fi, etc.)

Core of the Embedded System

- Embedded systems are domain and application specific and are built around a central core.
- The core of the embedded system falls into any one of the following categories:

1. General Purpose and Domain Specific Processors

   1.1 Microprocessors

   1.2 Microcontrollers

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

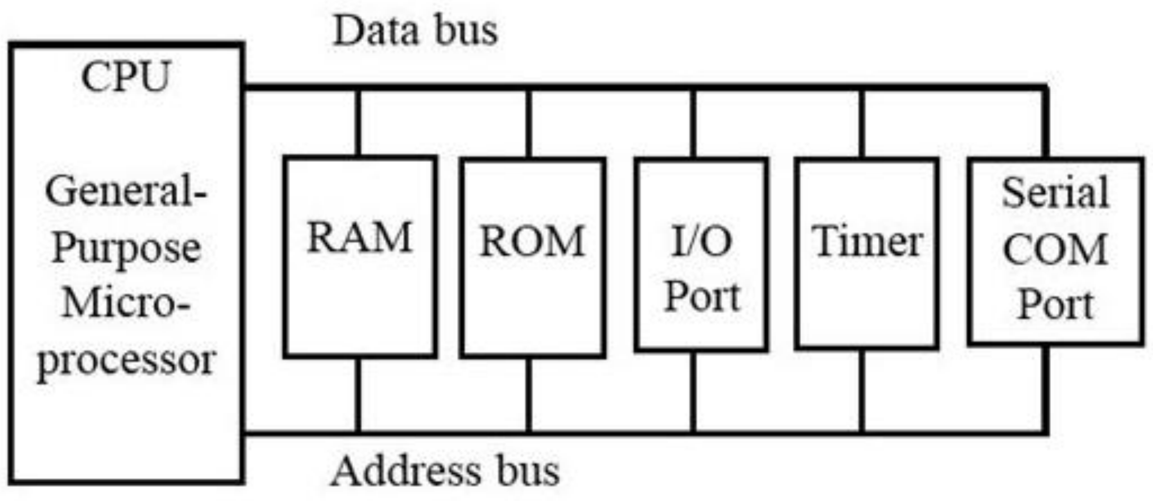
Microprocessor vs. Microcontroller

- A Microprocessor is 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, scratch pad 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.

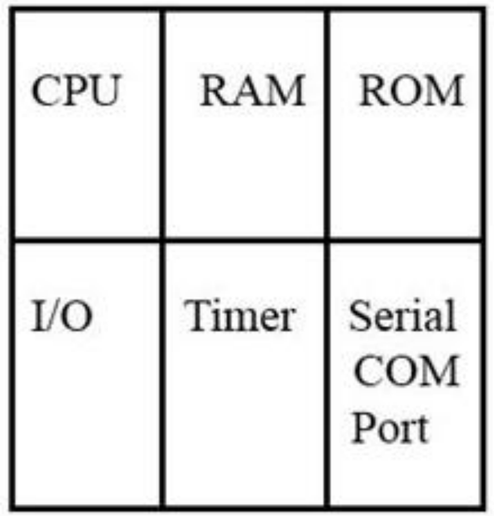
Microprocessor vs. Microcontroller (continued)

Microprocessor	Microcontroller
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, scratchpad 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 it doesn't require external interrupt controller, timer, 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 like 8255	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 compared to microcontrollers	Includes lot of power saving features

Microprocessor vs. Microcontroller (continued)



Microprocessor-based system



Microcontroller

RISC vs. CISC Processors/Controllers

- RISC stands for Reduced Instruction Set Computing.
  - All RISC processors/controllers possess lesser number of instructions, typically in the range of 30 to 40.
  - E.g.: Atmel AVR microcontroller – its instruction set contains only 32 instructions.
- CISC stands for Complex Instruction Set Computing.
  - The instruction set is complex and instructions are high in number.
  - E.g.: 8051 microcontroller – its instruction set contains 255 instructions.

RISC	CISC
Lesser number of instructions	Greater number of instructions
Instruction pipelining and increased execution speed	Generally no instruction pipelining feature
Orthogonal instruction set (Allows each instruction to operate on any register and use any addressing mode)	Non-orthogonal instruction set (All instructions are not allowed to operate on any register and use any addressing mode. It is instruction-specific)
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
A large number of registers are available	Limited number of general purpose registers
Programmer needs to write more code to execute a task since the instructions are simpler ones	Instructions are like macros in C language. A programmer can achieve the desired functionality with a single instruction which in turn provides the effect of using more simpler single instructions in RISC
Single, fixed length instructions	Variable length instructions
Less silicon usage and pin count	More silicon usage since more additional decoder logic is required to implement the complex instruction decoding
With Harvard Architecture	Can be Harvard or Von-Neumann Architecture

Harvard vs. Von-Neumann Processor/Controller Architecture

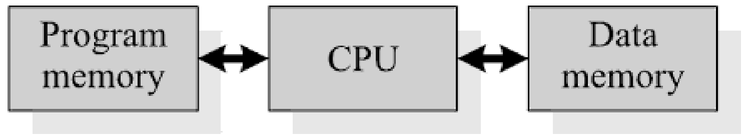
- Von-Neumann Architecture
  - Microprocessors/controllers based on the Von-Neumann architecture share a single common bus for fetching both instructions and data.
  - Program instructions and data are stored in a common main memory.
  - They first fetch an instruction and then fetch the data to support the instruction from code memory.
    - The two separate fetches slows down the controller's operation.
  - Von-Neumann architecture is also referred as Princeton architecture, since it was developed by the Princeton University.

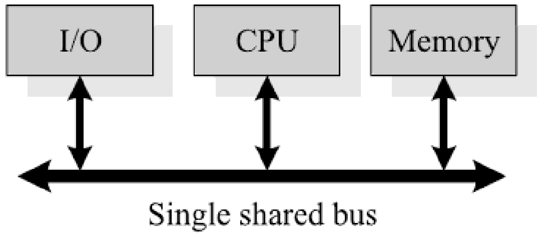
Harvard vs. Von-Neumann Processor/Controller Architecture (continued)

- Harvard 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.
  - The data memory can be read and written while the program memory is being accessed.
  - These separated data memory and code memory buses allow one instruction to execute while the next instruction is fetched ("pre-fetching").
    - The pre-fetch theoretically allows much faster execution than Von-Neumann architecture.

Harvard vs. Von-Neumann Processor/Controller Architecture (continued)

Harvard Architecture	Von-Neumann Architecture
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
Since data memory and program memory are stored physically in different locations, no chances for accidental corruption of program memory	Since data memory and program memory are stored physically in the same chip, chances for accidental corruption of program memory





Sensors and Actuators

Sensors and Actuators

- An embedded system is in constant interaction with the real world and the controlling/monitoring functions executed by the embedded system is achieved in accordance with the changes happening to the real world.
- The changes in system environment or variables are detected by the sensors connected to the input port of the embedded system.
- If the embedded system is designed for any controlling purpose, the system will produce some changes in the controlling variable to bring the controlled variable to the desired value.
  - It is achieved through an actuator connected to the output port of the embedded system.

Sensors and Actuators (continued)

- A sensor is a transducer device that converts energy from one form to another for any measurement or control purpose.
  - E.g.: Temperature sensor, magnetic hall effect sensor, humidity sensor, etc.
- An actuator is a form of transducer device (mechanical or electrical) which converts signals to corresponding physical action (motion).
  - Actuator acts as an output device.
  - E.g.: Stepper motor

Sensors and Interfacing

Instrumentation and Control Systems

- Fig. shows the arrangement of an instrumentation system.

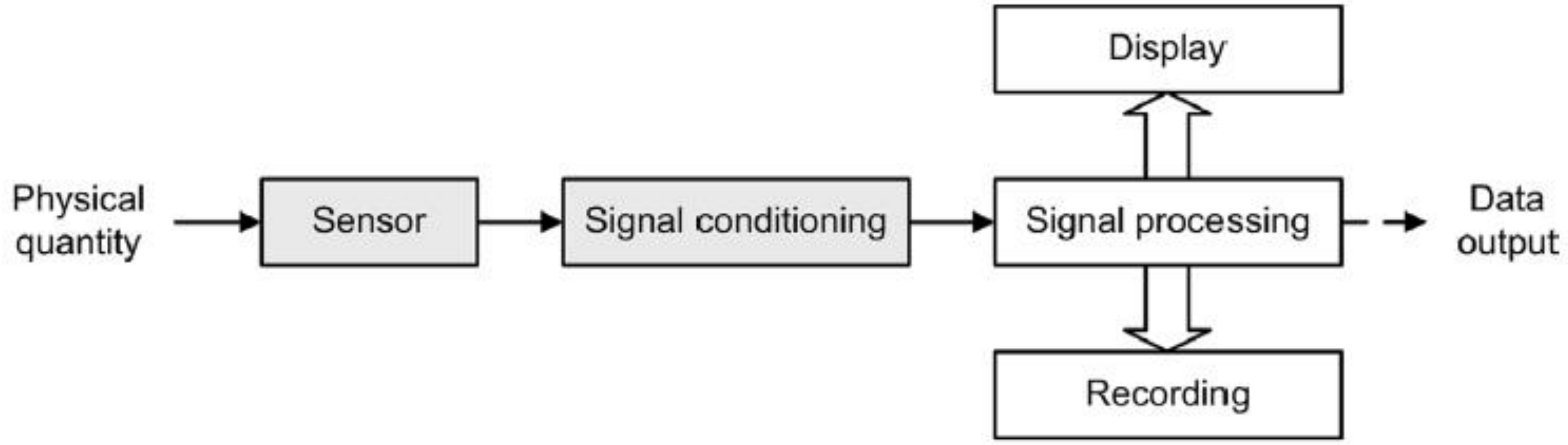


Fig: An Instrumentation System

Instrumentation and Control Systems (continued)

- 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.
- 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.

Instrumentation and Control Systems (continued)

- Fig. shows the arrangement of a control system.

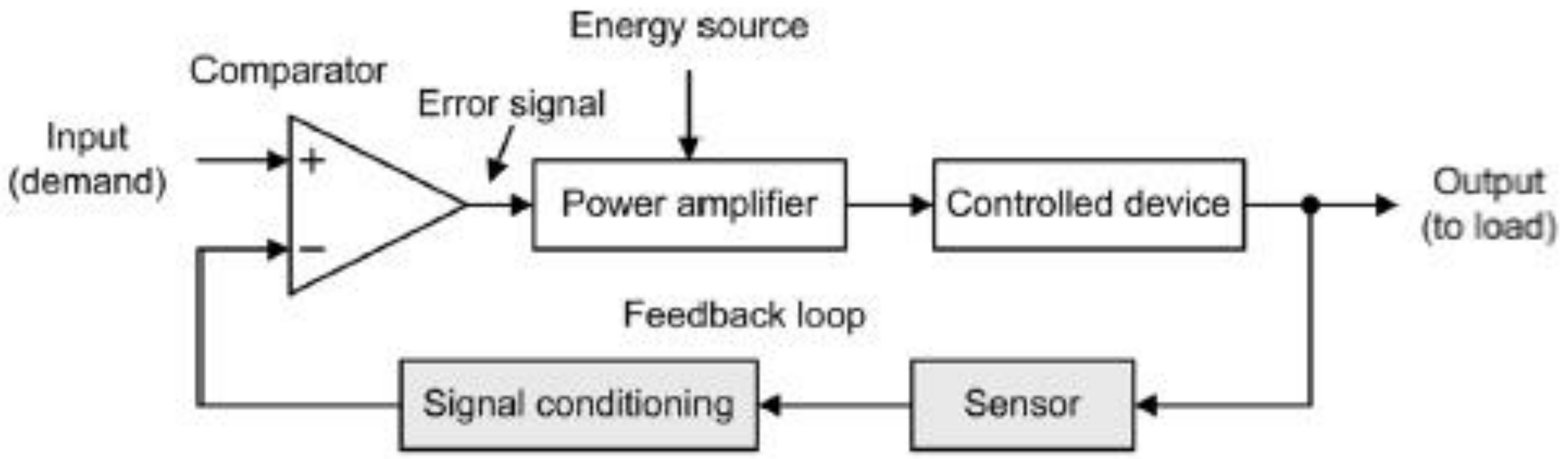


Fig: A Control System

Instrumentation and Control Systems (continued)

- The control system 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.

Instrumentation and Control Systems (continued)

- 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.
  - For example, 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.
  - Loudspeakers and microphones can thus be considered as complementary transducers.

Transducers (continued)

- Transducers may be used both as inputs to electronic circuits and outputs from them.
  - For example, 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.

Transducers (continued)

- Some examples of input transducers

Physical Quantity	Input Transducer	Notes
Sound (pressure change)	Dynamic microphone	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	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.

Transducers (continued)

A selection of thermocouple probes

A selection of audible transducers

Transducers (continued)

- Some examples of output transducers

Physical Quantity	Output Transducer	Notes
Sound (pressure change)	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.
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 (continued)

- 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 (continued)

- 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. It is thus said to be *continuously variable*.

Sensors (continued)

- Some examples of input transducers (sensors)

Physical Quantity	Input Transducer (Sensor)	Notes
Angular position	Resistive rotary position sensor	Rotary track potentiometer with linear law produces analogue voltage proportional to angular position.
	Optical shaft encoder	Encoded disk interposed between optical transmitter and receiver (infrared LED and photodiode or photo-transistor).
Angular velocity	Tachogenerator	Small d.c. generator with linear output characteristic. Analogue output voltage proportional to shaft speed.
	Toothed rotor tachometer	Magnetic pick-up responds to the movement of a toothed ferrous disk. The pulse repetition frequency of the output is proportional to the angular velocity.
Flow	Rotating vane flow sensor	Turbine rotor driven by fluid. Turbine interrupts infra-red beam. Pulse repetition frequency of output is proportional to flow rate.

Sensors (continued)

Resistive linear position sensor

Liquid flow sensor (digital output)

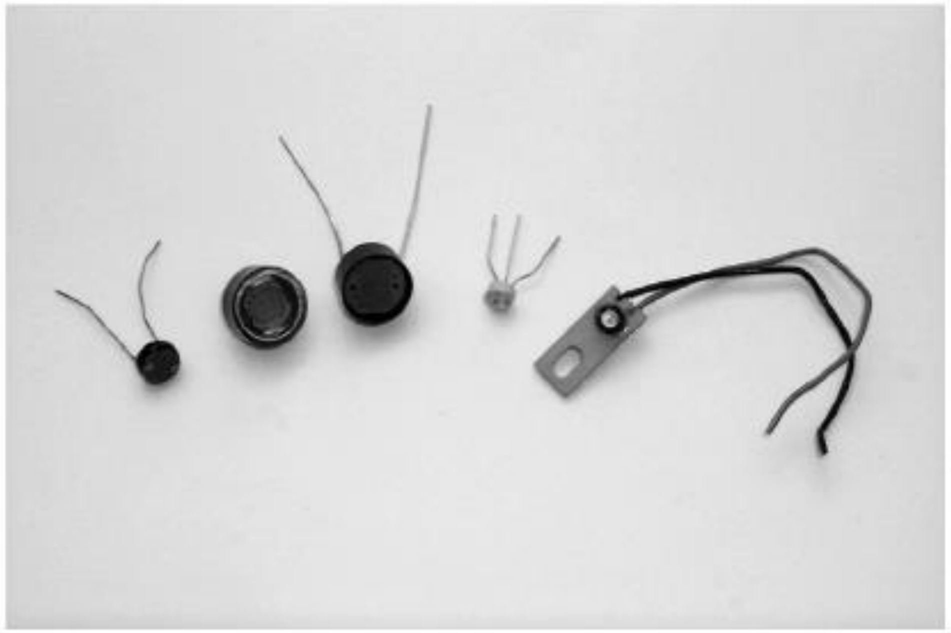
Sensors (continued)

Physical Quantity	Input Transducer (Sensor)	Notes
Linear position	Resistive linear position sensor	Linear track potentiometer with linear law produces analogue voltage proportional to linear position. Limited linear range.
	Linear variable differential transformer (LVDT)	Miniature transformer with split secondary windings and moving core attached to a plunger. Requires a.c. excitation and phase-sensitive detector.
	Magnetic linear position sensor	Magnetic pick-up responds to movement of a toothed ferrous track. Pulses are counted as the sensor moves along the track.

Sensors (continued)

Physical Quantity	Input Transducer (Sensor)	Notes
Light level	Photocell	Voltage-generating device. The analogue output voltage produced is proportional to light level.
	Light-dependent resistor (LDR)	An analogue output voltage results from a change of resistance within a cadmium sulphide (CdS) sensing element. Usually connected as part of a potential divider or bridge.
	Photodiode	Two-terminal device connected as a current source. An analogue output voltage is developed across a series resistor of appropriate value.
	Phototransistor	Three-terminal device connected as a current source. An analogue output voltage is developed across a series resistor of appropriate value.

Sensors (continued)


Various optical and light sensors

Sensors (continued)

Physical Quantity	Input Transducer (Sensor)	Notes
Liquid level	Float switch	Simple switch element which operates when a particular level is detected.
	Capacitive proximity switch	Switching device which operates when a particular level is detected. Ineffective with some liquids.
	Diffuse scan proximity switch	Switching device which operates when a particular level is detected. Ineffective with some liquids.

Sensors (continued)

Liquid level float switch

Sensors (continued)

Physical Quantity	Input Transducer (Sensor)	Notes
Pressure	Microswitch pressure sensor	Microswitch fitted with actuator mechanism and range-setting springs. Suitable for high-pressure applications.
	Differential pressure vacuum switch	Microswitch with actuator driven by a diaphragm. May be used to sense differential pressure. Alternatively, one chamber may be evacuated and the sensed pressure applied to a second input.
	Piezo-resistive pressure sensor	Pressure exerted on diaphragm causes changes of resistance in attached piezo-resistive transducers. Transducers are usually arranged in the form of a four active element bridge which produces an analogue output voltage.

Sensors (continued)

Various switch sensors

Sensors (continued)

Physical Quantity	Input Transducer (Sensor)	Notes
Proximity	Reed switch	Reed switch and permanent magnet actuator. Only effective over short distances.
	Inductive proximity switch	Target object modifies magnetic field generated by the sensor. Only suitable for metals (non-ferrous metals with reduced sensitivity).
	Capacitive proximity switch	Target object modifies electric field generated by the sensor. Suitable for metals, plastics, wood and some liquids and powders.
	Optical proximity switch	Available in diffuse and through scan types. Diffuse scan types require reflective targets. Both types employ optical transmitters and receivers (usually infra-red emitting LEDs and photo-diodes or photo-transistors). Digital input port required.

Sensors (continued)

Physical Quantity	Input Transducer (Sensor)	Notes
Strain	Resistive strain gauge	Foil type resistive element with polyester backing for attachment to body under stress. Normally connected in full bridge configuration with temperature-compensating gauges to provide an analogue output voltage.
	Semiconductor strain gauge	Piezo-resistive elements provide greater outputs than comparable resistive foil types. More prone to temperature changes and also inherently non-linear.

Sensors (continued)

Physical Quantity	Input Transducer (Sensor)	Notes
Temperature	Thermocouple	Small e.m.f. generated by a junction between two dissimilar metals. For accurate measurement, requires compensated connecting cables and specialized interface.
	Thermistor	Usually connected as part of a potential divider or bridge. An analogue output voltage results from resistance changes within the sensing element.
	Semiconductor temperature sensor	Two-terminal device connected as a current source. An analogue output voltage is developed across a series resistor of appropriate value.

Sensors (continued)

Various temperature and gas sensors

Sensors (continued)

Physical Quantity	Input Transducer (Sensor)	Notes
Weight	Load cell	Usually comprises four strain gauges attached to a metal frame. This assembly is then loaded and the analogue output voltage produced is proportional to the weight of the load.
Vibration	Electromagnetic vibration sensor	Permanent magnet seismic mass suspended by springs within a cylindrical coil. The frequency and amplitude of the analogue output voltage are respectively proportional to the frequency and amplitude of vibration.

Actuators

- An actuator is a form of transducer device (mechanical or electrical) which converts signals to corresponding physical action (motion).

The I/O Subsystem

- The I/O subsystem of the embedded system facilitates the interaction of the embedded system with the external world.
- The interaction happens through the sensors and actuators connected to the input and output ports respectively 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, optocouplers, etc.

Light Emitting Diode (LED)

- Light Emitting Diode (LED) is an important output device for visual indication in any embedded system.
- LED can be used as an indicator for the status of various signals or situations.
  - E.g.: 'Device ON', 'Battery low' or 'Charging of battery' conditions
- Light Emitting Diode 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 of the supply voltage and cathode to the -ve terminal of supply voltage.
- The current flowing through the LED must be limited to a value below the maximum current that it can conduct.
  - A resister is used in series to limit the current through the LED.
- The ideal LED interfacing circuit is shown in the figure.

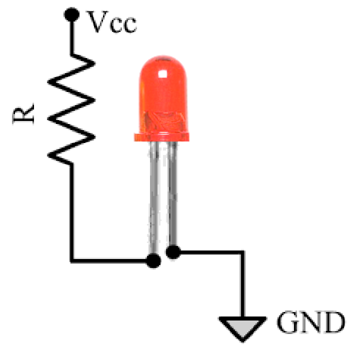


Fig: LED interfacing

Light Emitting Diode (LED) (continued)

- LEDs can be interfaced to the port pin of a processor/controller in two ways:
  - In the first method, the anode is directly connected to the port pin and the port pin drives the LED.
    - The port pin 'sources' current to the LED when the port pin is at logic High (Logic ‘1’).
  - In the second method, the cathode of the LED is connected to the port pin of the processor/controller and the anode to the supply voltage through a current limiting resistor.
    - The LED is turned on when the port pin is at logic Low (Logic '0’).
    - Here the port pin 'sinks' current.

7-Segment LED Display

- The 7-segment LED display is an output device for displaying alpha numeric characters.
- It contains 7 LED segments arranged in a special form used for displaying alpha numeric characters and 1 LED used for representing 'decimal point' in decimal number display.
- 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.

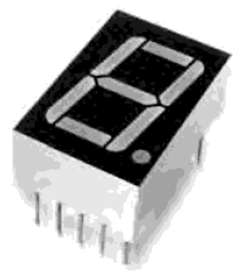
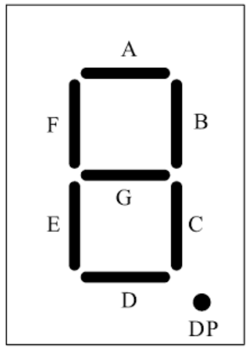
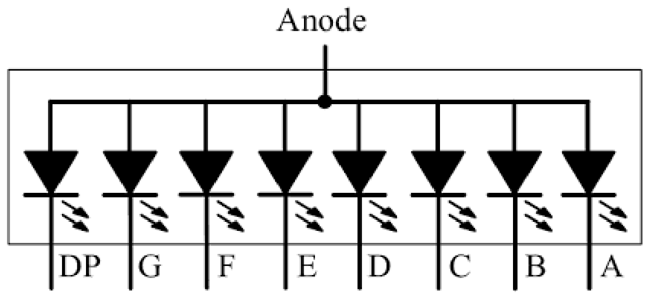



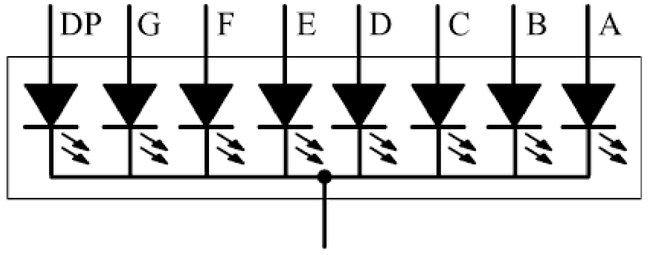
Fig: 7-Segment LED Display

7-Segment LED Display (continued)

- The 7-segment LED displays are available in two different configurations, namely; Common Anode and Common Cathode.
- In the common anode configuration, the anodes of the 8 segments are connected commonly whereas in the common cathode configuration, the cathodes of 8 LED segments are connected commonly.
- Figure illustrates the Common Anode and Cathode configurations.



Common Anode LED Display



Common Cathode LED Display

7-Segment LED Display (continued)

- Based on the configuration of the 7-segment LED unit, the LED segment's anode or cathode is connected to the port of the processor/controller in the order 'A' segment to the least significant port pin and DP segment to the most significant port pin.
- The current flow through each of the LED segments should be limited to the maximum value supported by the LED display unit.
  - The typical value is 20mA.
  - The current can be limited by connecting a current limiting resistor to the anode or cathode of each segment.
- 7-segment LED display is used in low cost embedded applications like Public telephone call monitoring devices, point of sale terminals, etc.

Stepper Motor

- A stepper motor is an electro-mechanical device which generates discrete displacement (motion) in response to dc electrical signals.
- Stepper motors are widely used in industrial embedded applications, consumer electronic products and robotics control systems, 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

Stepper Motor (continued)

1. 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.
     - The direction of rotation can be shifted 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).

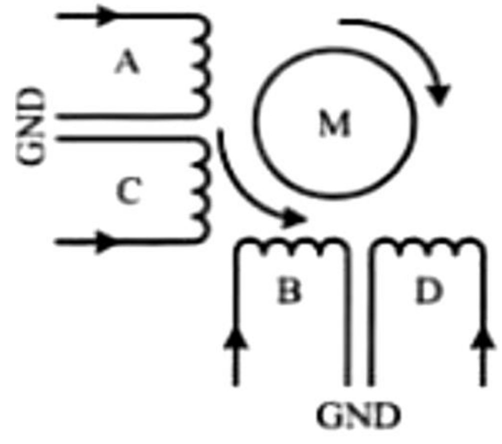


Fig: 2-Phase unipolar stepper motor

Stepper Motor (continued)

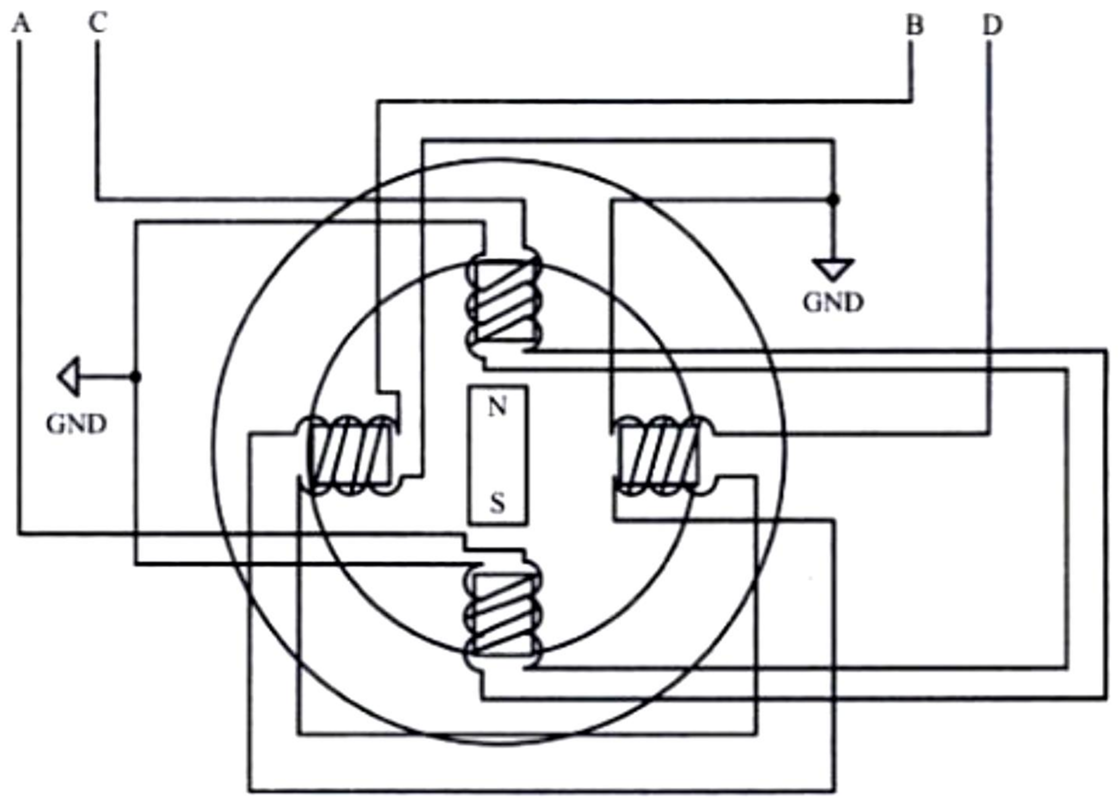


Fig: Stator Winding details for a 2-Phase unipolar stepper motor

Stepper Motor (continued)

2. Bipolar
   - A bipolar stepper motor contains single winding per phase.
   - For reversing the motor rotation, the current flow through the windings is reversed dynamically.
   - It requires complex circuitry for current flow reversal.

Stepper Motor (continued)

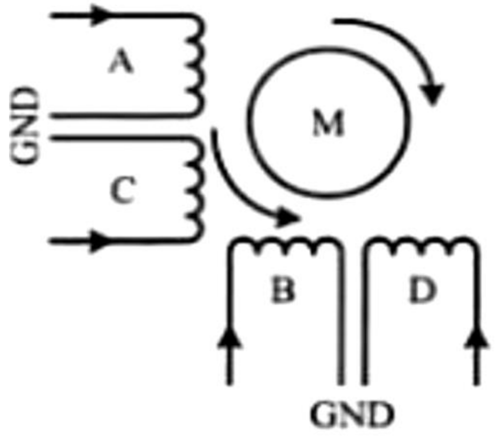
- The stepping of stepper motor can be implemented in different ways by changing the sequence of activation of the stator winding.
- Different stepping modes are supported by the stepper motor:
  - Full step
  - Wave Step
  - Half Step
- The rotation of the stepper motor can be reversed by reversing the order in which the coil is energised.

Stepper Motor (continued)

Full Step

- In the full step mode both the phases are energised simultaneously.
- The coils A, B, C and D are energised in the following order:

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



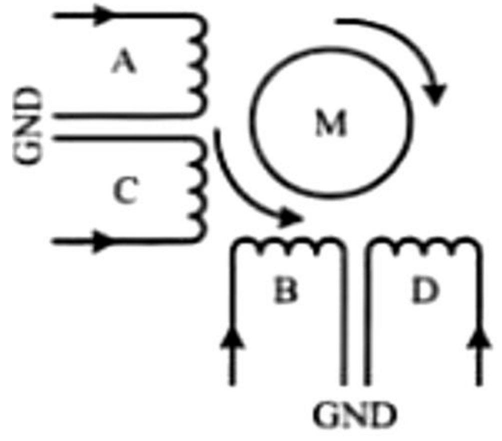
- It should be noted that out of the two windings, only one winding of a phase is energised at a time

Stepper Motor (continued)

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

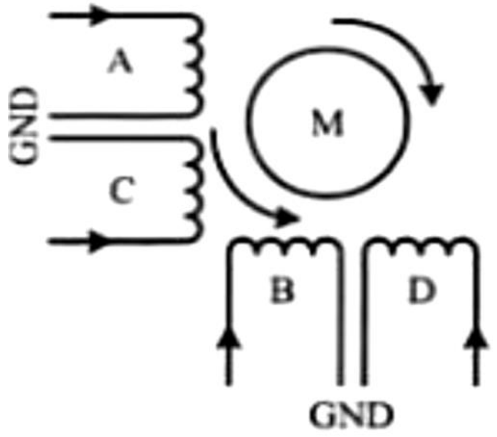


Stepper Motor (continued)

Half Step

- It uses the combination of wave and full step.
- It has the highest torque and stability.
- The coil energising sequence for half step is as shown in the table:

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



Stepper Motor (continued)

- 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

Stepper Motor (continued)

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

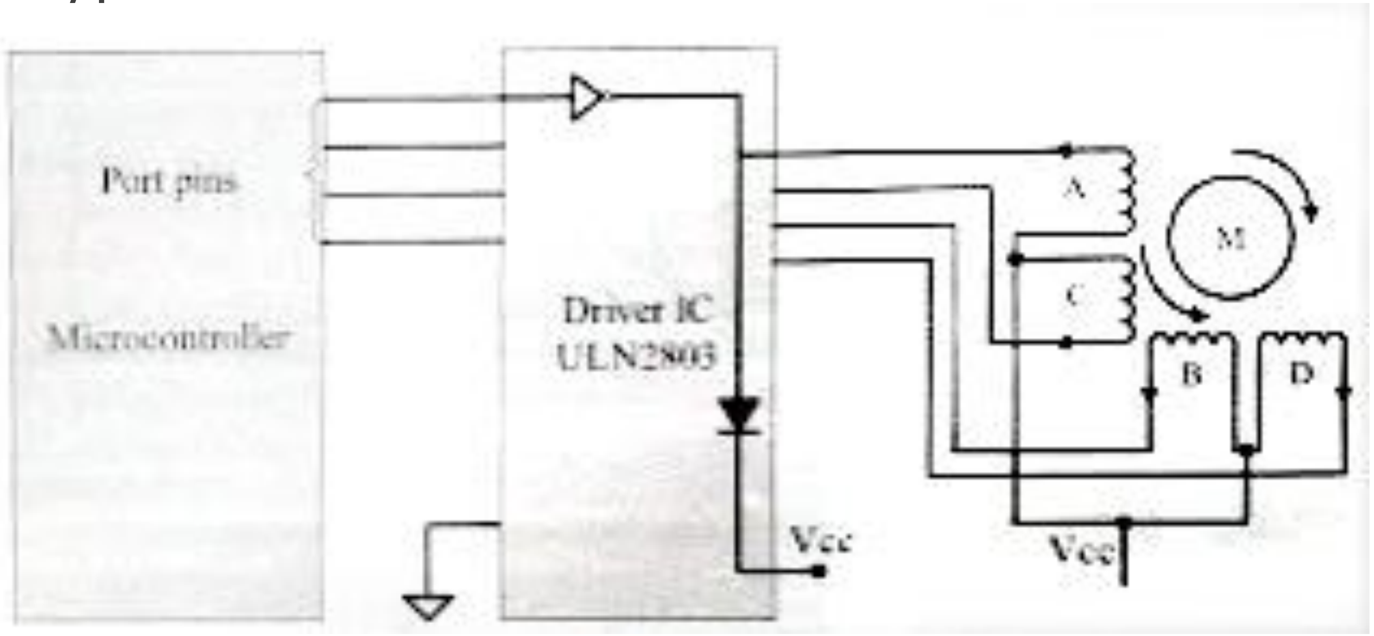


Fig: Interfacing of stepper motor through driver circuit

Relay

- Relay is an electro-mechanical device.
- In embedded application, the Relay unit acts as dynamic path selector for signals and power.
- 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.
- 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.

Relay (continued)

- Relays are available in different configurations.
- Figure given below illustrates the widely used relay configurations for embedded applications.

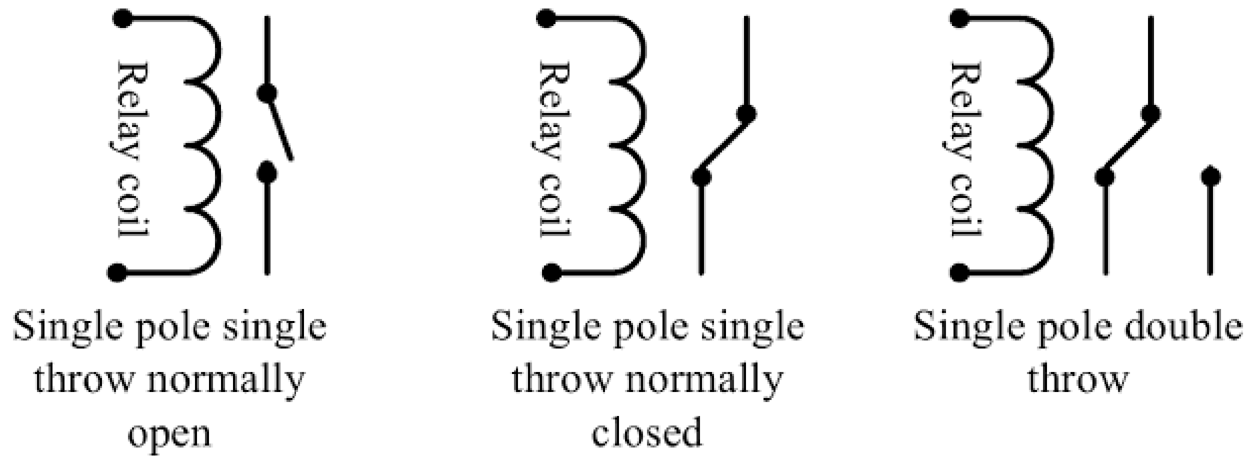

Relay coil
Single pole single throw normally open
Relay coil
Single pole single throw normally closed
Relay coil
Single pole double throw

Fig: Relay configurations

Relay (continued)

- The 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 Single Pole Single Throw relay, the circuit is normally open and it becomes closed when the relay is energised.
  - For Normally Closed Single Pole Single Throw relay, the circuit is normally closed and it becomes open when the relay is energised.
- For Single Pole Double Throw configuration, there are two paths for information flow and they are selected by energising or de-energising the relay.

Relay (continued)

- The Relay is normally controlled using a relay driver circuit connected to the port pin of the processor/controller.
- A transistor is used for building the relay driver circuit as shown in the figure.
- A free-wheeling diode is used for free-wheeling the voltage produced in the opposite direction when the relay coil is de-energised.
- The freewheeling diode is essential for protecting the relay and the transistor.
- Most of the industrial relays are bulky and require high voltage to operate.
- Special relays called 'Reed' relays are available for embedded application requiring switching of low voltage DC signals.

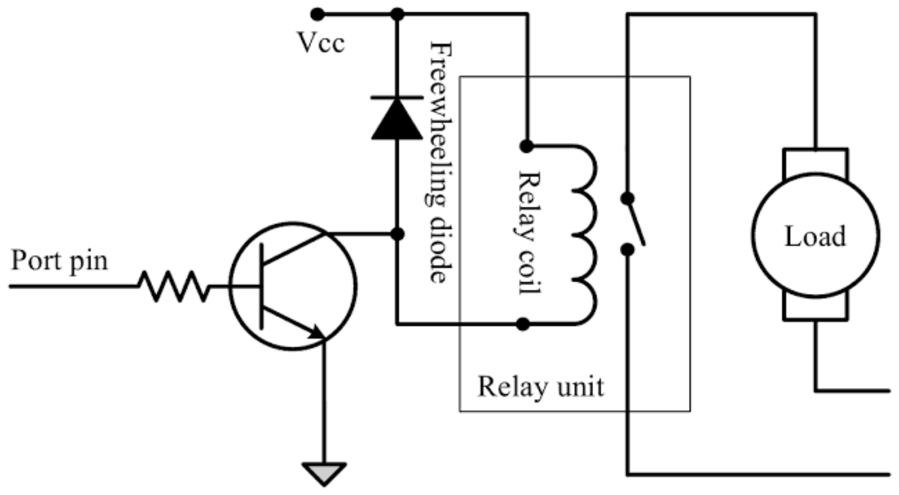


Fig: Transistor based Relay driving circuit

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 – 'Self-driving’ and '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 support 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.
- Depending on the driving current requirements, the piezo buzzer can also be interfaced using a transistor based driver circuit as in the case of a 'Relay'.

Push Button Switch

- It is an input device.
- Push button switch comes in two configurations, namely 'Push to Make' and 'Push to Break’.
- In the 'Push to Make' configuration, the switch is normally in the open state and it makes a circuit contact when it is pushed or pressed.
- 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 released.

Push Button Switch (continued)

- Push button is used for generating a momentary pulse.
- In embedded applications, 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/controller.
- Depending on the way in which the push button interfaced to the controller, it can generate either a 'HIGH' pulse or a 'LOW' pulse.
- Figure illustrates how the push button can be used for generating 'LOW' and 'HIGH' pulses.

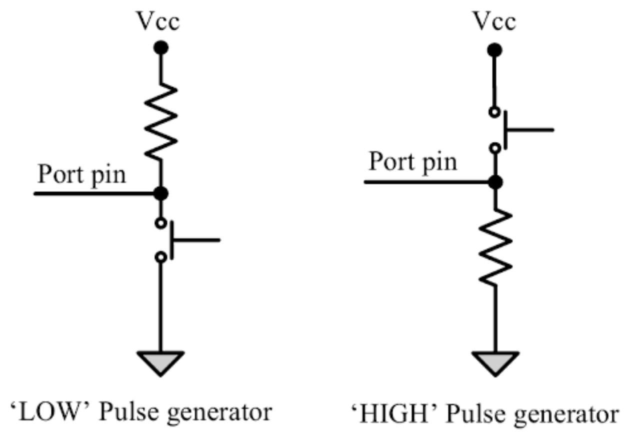


Fig: Push button switch configurations

Keyboard

- It 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.
  - For example, 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 x 4 Rows matrix.

Keyboard (continued)

- Figure illustrates the connection of keys in a matrix keyboard.

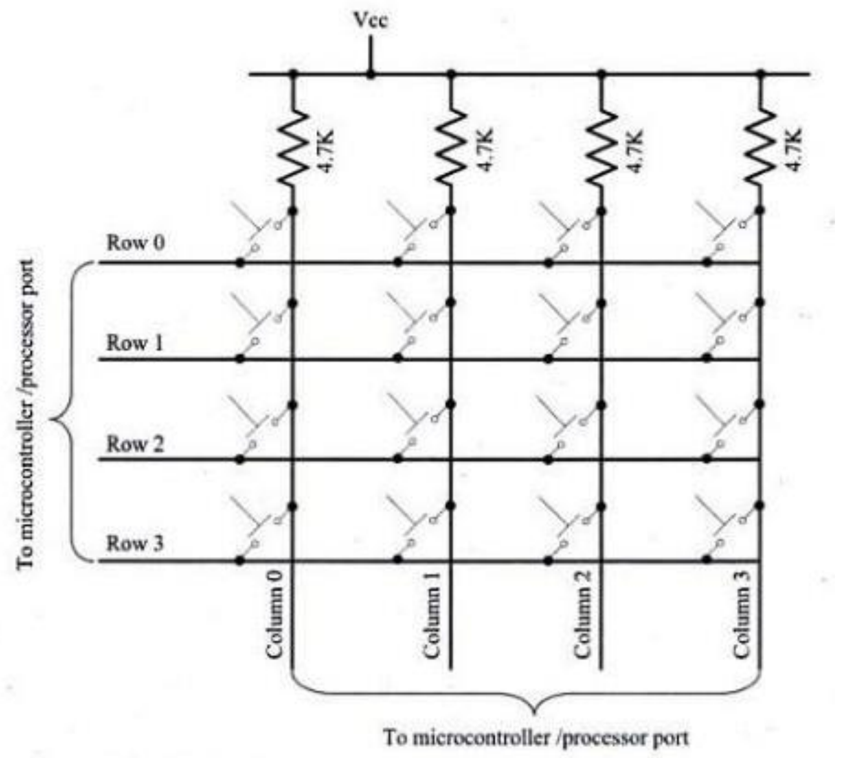


Fig: Matrix keyboard interfacing

Keyboard (continued)

- In a matrix keyboard, the keys are arranged in the form of a matrix, i.e., they are connected in rows and columns.
- 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.
- When a row is pulled low and if a key connected to the row is pressed, reading the column to which the key is connected will give logic 0.

Keyboard (continued)

- 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.
  - If the key press is a genuine one, the state of the key will remain as 'pressed’ on the second read also.

Keyboard (continued)

- Pull-up resistors are connected to the column lines to limit the current that flows to the Row line on a key press.

Communication Interface

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:
  - Onboard Communication Interface (Device/board level communication interface)
    - E.g.: Serial interfaces like I2C, SPI, UART, 1-Wire, etc and parallel bus interface.
  - External Communication Interface (Product level communication interface)
    - E.g.: Wireless interfaces like Infrared (IR), Bluetooth (BT), Wireless LAN (Wi-Fi), Radio Frequency waves (RF), GPRS, etc. and wired interfaces like RS-232C/RS-422/RS-485, USB, Ethernet IEEE 1394 port, Parallel port, CF-II interface, SDIO, PCMCIA/PCIex, etc.

Onboard Communication Interfaces

- An embedded system is a combination of different types of components (chips/devices) arranged on a printed circuit board (PCB).
- Onboard Communication Interface refers to the different communication channels/buses for interconnecting the various integrated circuits and other peripherals within the embedded system.
- E.g.: Serial interfaces like I2C, SPI, UART, 1-Wire, etc and parallel bus interface

Universal Asynchronous Receiver Transmitter (UART)

- Universal Asynchronous Receiver Transmitter (UART) based data transmission is an asynchronous form of serial data transmission.
- It doesn't require a clock signal to synchronise the transmitting end and receiving end for transmission.
- Instead it relies upon the pre-defined agreement between the transmitting device and receiving device.

Universal Asynchronous Receiver Transmitter (UART) (continued)

- The serial communication settings (Baudrate, number of bits per byte, parity, number 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.

Universal Asynchronous Receiver Transmitter (UART) (continued)

- 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 baud rate settings.
  - If the baud rate is 'x' bits per second, the time slot available for one bit is 1/x seconds.
- The receiver unit polls the receiver line at exactly half of the time slot available for the bit.
- If parity is enabled for communication, the UART of the transmitting device adds a parity bit (bit value is 1 for odd number of 1s in the transmitted bit stream and 0 for even number of 1s).
- 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
  - In the case of 8 bits/byte, the byte is formed with the received 8 bits with the first received bit as the LSB and last received data bit as MSB.

Universal Asynchronous Receiver Transmitter (UART) (continued)

- For proper communication, the 'Transmit line' of the sending device should be connected to the 'Receive line' of the receiving device.
- In addition to the serial data transmission function, UART provides hardware handshaking signal support for controlling the serial data flow.
- UART chips are available from different semiconductor manufacturers.
  - National Semiconductor's 8250 UART chip is considered as the standard setting UART. It was used in the original IBM PC.
- Nowadays most of the microprocessors/controllers are available with integrated UART functionality and they provide built-in instruction support for serial data transmission and reception.

Universal Asynchronous Receiver Transmitter (UART) (continued)

- Figure illustrates the UART interfacing.

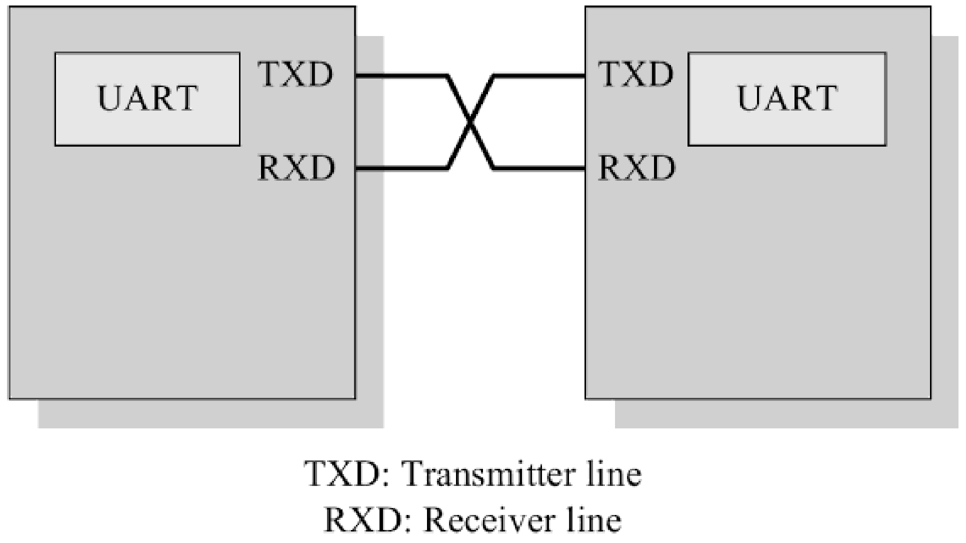


Fig: UART Interfacing

Parallel Interface

- The on-board 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.

Parallel Interface (continued)

- The device is normally memory mapped to the host processor and a range of address is assigned to it.
- An address decoder circuit is used for generating the chip select signal for the device.
- When the address selected by the processor is within the range assigned for the device, the decoder circuit activates the chip select line and thereby the device becomes active.
- The processor then can read or write from or to the device by asserting the corresponding control line (RD\ and WR\ respectively).

Parallel Interface (continued)

- The bus interface diagram shown in the figure illustrates the interfacing of devices through parallel interface.

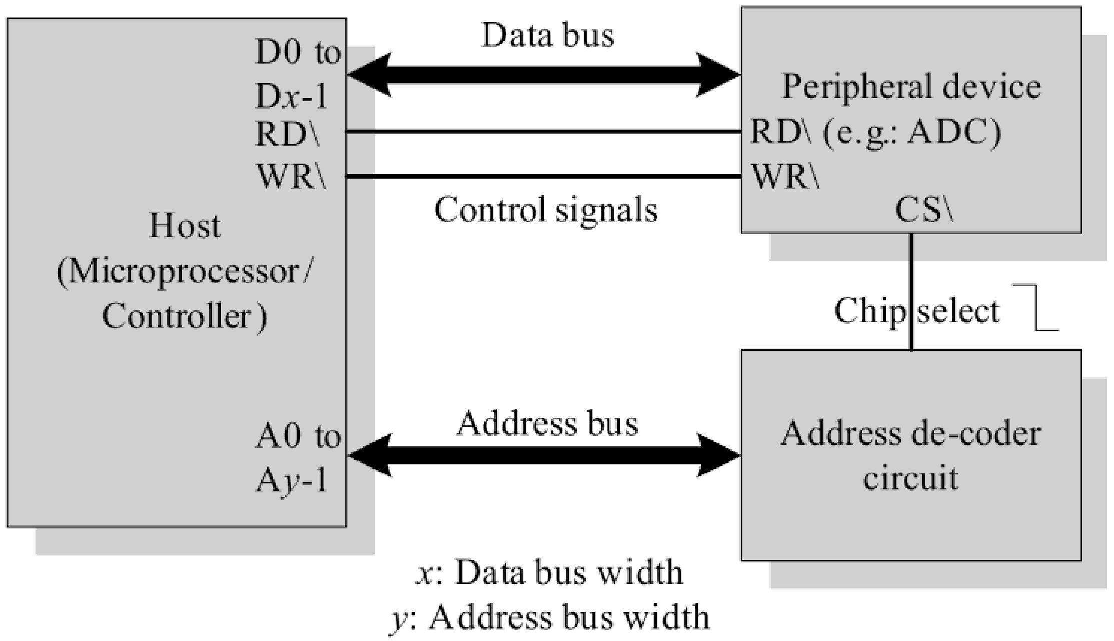


Fig: Parallel Interface Bus

Parallel Interface (continued)

- Parallel communication is host processor initiated.
- If a device wants to initiate the communication, it can inform the same to the processor through interrupts.
  - For this, the interrupt line of the device is connected to the interrupt line of the processor and the corresponding interrupt is enabled in the host processor.
- The width of the parallel interface is determined by the data bus width of the host processor.
  - It can be 4 bit, 8 bit, 16 bit, 32 bit or 64 bit etc.
  - The bus width supported by the device should be same as that of the host processor.
- Parallel data communication offers the highest speed for data transfer.

External Communication Interfaces

- External Communication Interface refers to the different communication channels/buses used by the embedded system to communicate with the external world.
- E.g.: RS-232 C & RS-485, Universal Serial Bus (USB), IEEE 1394 (Firewire), Infrared (IR), Bluetooth (BT), Wi-Fi, ZigBee, GPRS, etc.

Universal Serial Bus (USB)

- Universal Serial Bus (USB) is a wired high speed serial bus for data communication.
- The first version of USB (USB 1.0) was released in 1995.
- The USB communication system follows a star topology with a USB host at the centre 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.

Universal Serial Bus (USB) (continued)

- Figure illustrates the star topology for USB device connection.

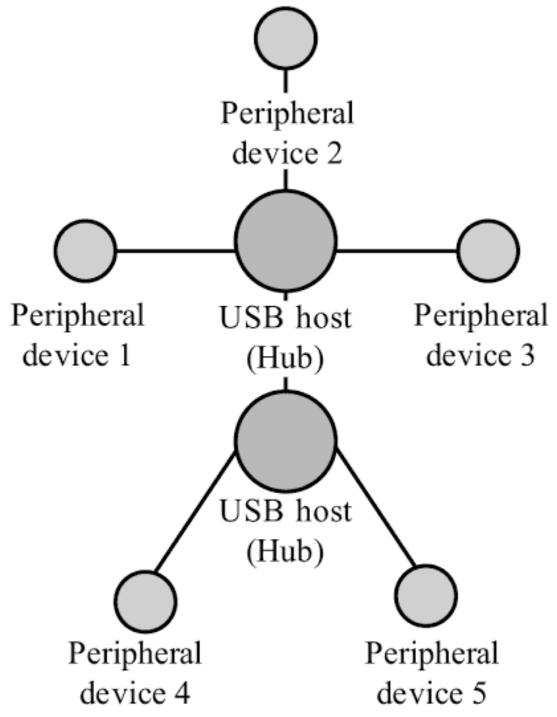


Fig: USB Device Connection topology

Universal Serial Bus (USB) (continued)

- 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.
- There are different standards for implementing the USB Host Control interface:
  - Open Host Control Interface (OHCI)
  - Universal Host Control Interface (UHCI)

Universal Serial Bus (USB) (continued)

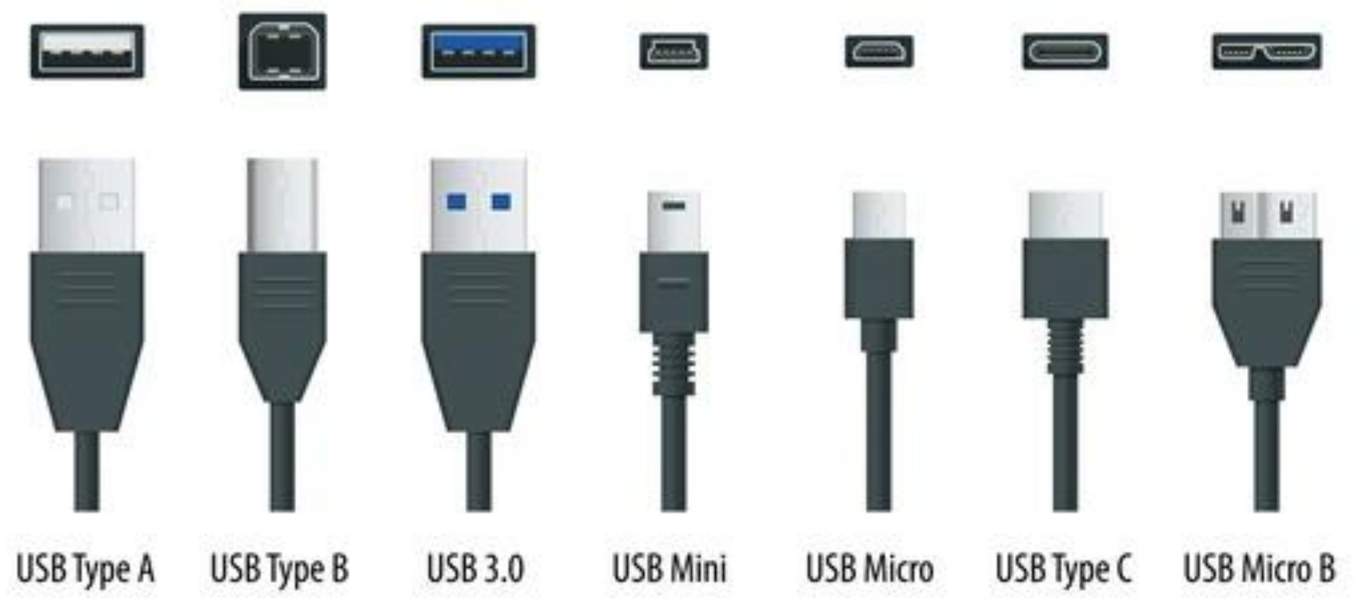
- The physical connection between a USB peripheral device and master device is established with a USB cable.
- The USB cable supports communication distance of up to 5 metres.
- The USB standard uses two different types of connector at the ends of the USB cable for connecting the USB peripheral device 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).
- The USB connector present in desktop PCs or laptops are examples for 'Type A' USB connector.

Universal Serial Bus (USB) (continued)

- Both Type A and Type B connectors contain 4 pins for communication.
- The Pin details for the connectors are listed in the table given below.

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

Universal Serial Bus (USB) (continued)



Universal Serial Bus (USB) (continued)

- USB uses differential signals for data transmission.
  - It improves the noise immunity.
- USB interface has the ability to supply power to the connecting devices.
  - Two connection lines (Ground and Power) of the USB interface are dedicated for carrying power.
  - It can supply power up to 500 mA at 5 V.
  - It is sufficient to operate low power devices.
- Mini and Micro USB connectors are available for small form factor devices like portable media players.
- Each USB device contains a Product ID (PID) and a Vendor ID (VID).
  - 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.

Universal Serial Bus (USB) (continued)

- USB supports four different types of data transfers:
- Control transfer : Used by USB system software to query, configure and issue commands to the USB device.
- Bulk transfer : Used for sending a block of data to a device.
  - Supports error checking and correction.
  - Transferring data to a printer is an example for bulk transfer.
- Isochronous data transfer : Used for real-time data communication.
  - Data is transmitted as streams in real-time.
  - Doesn't support error checking and re-transmission of data in case of any transmission loss.
  - All streaming devices like audio devices and medical equipment for data collection make use of the isochronous transfer.
- Interrupt transfer : 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.

Universal Serial Bus (USB) (continued)

- USB.ORG is the standards body for defining and controlling the standards for USB communication.
- Presently USB supports different data rates:
  - Low-Speed (LS) - 1.5Mbps – USB 1.0
  - Full-Speed (FS) - 12Mbps – USB 1.0
  - High-Speed (HS) - 480Mbps – USB 2.0
  - SuperSpeed (SS) - 5Gbps – USB 3.0
  - SuperSpeed+ (SS+) - 10Gbps – USB 3.1, 20 Gbps – USB 3.2

Wi-Fi

- Wi-Fi or Wireless Fidelity is the 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.
- It is essential to have device identities in a multipoint communication to address specific devices for data communication.
- In an IP based communication each device is identified by an IP address, which is unique to each device on the network.

Wi-Fi (continued)

- 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 enabled devices contain a wireless adaptor for transmitting and receiving data in the form of radio signals through an antenna.
- The hardware part of it is known as Wi-Fi Radio.
- Wi-Fi operates at 2.4 GHz or 5 GHz of radio spectrum and they co-exist with other ISM band devices like Bluetooth.

Wi-Fi (continued)

- Figure illustrates the typical interfacing of devices in a Wi-Fi network.

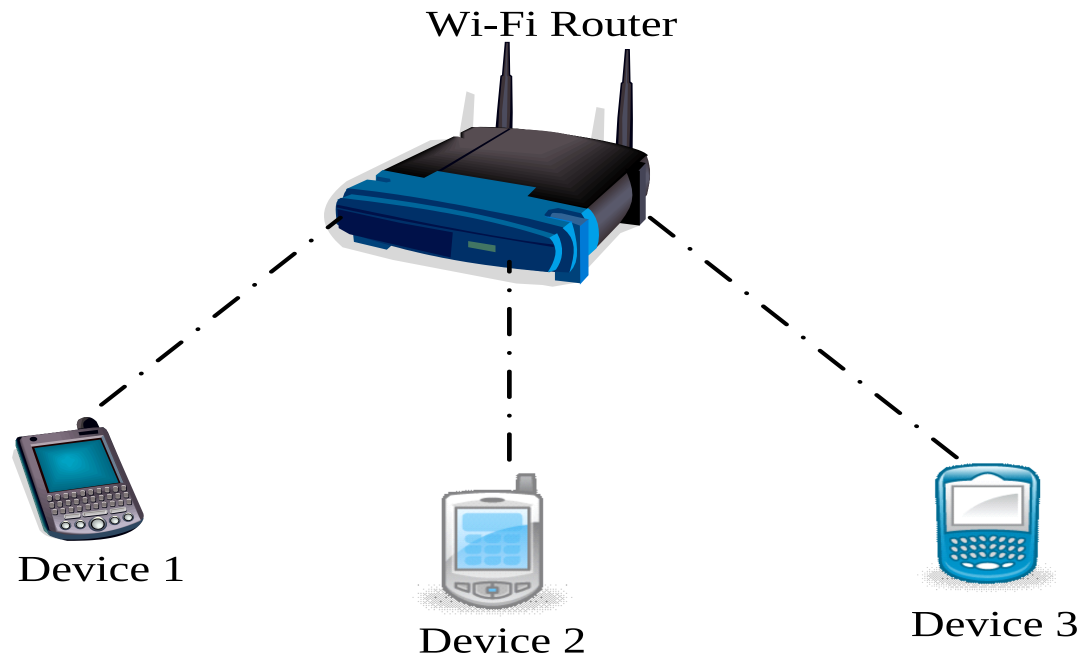


Fig: Wi-Fi Network

Wi-Fi (continued)

- 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.
- Wi-Fi employs different security mechanisms like Wired Equivalency Privacy (WEP), Wireless Protected Access (WPA), etc. for securing the data communication.
- Wi-Fi supports data rates ranging from 1 Mbps to 1.73 Gbps 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.

General Packet Radio Service (GPRS)

- General Packet Radio Service (GPRS) is a communication technique for transferring data over a mobile communication network like GSM.
- Data is sent as packets in GPRS communication.
- 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 171.2 Kbps.
- In GPRS communication, the radio channel is concurrently shared between several users instead of dedicating a radio channel to a cell phone user.

General Packet Radio Service (GPRS) (continued)

- 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.
- To accomplish GPRS based communication, the carrier network also should have support for GPRS communication.

General Packet Radio Service (GPRS) (continued)

- GPRS is an old technology and it is being replaced by new generation data communication techniques like EDGE, High Speed Downlink Packet Access (HSDPA), 4G, Long Term Evolution (LTE), 5G, etc. which offers higher bandwidths for communication.
- 3G offers data rates ranging from 144Kbps to 2 Mbps or 14.4 Mbps with High Speed Packet Access (HSPA).
- 4G gives a practical data rate of 2 to 100+ Mbps depending on the network and underlying technology.
- 5G is aimed at being as fast as 35.46 Gbps.