EMBEDDED SYSTEMS

- Course Code : 18EC62
- CIE Marks :40
- Lecture Hours/Week : 03 + 2 (Tutorial)
- SEE marks :60
- Total Number of Lecture Hours : 50 (10 Hrs / Module)
- Exam Hours : 03
- CREDITS : 04

Module-3

Embedded System Components: Embedded Vs General computing system, Classification of Embedded systems, Major applications and purpose of ES. Elements of an Embedded System (Block diagram and explanation),Differences between RISC and CISC, Harvard and Princeton, Big and LittleEndian formats, Memory (ROM and RAM types), Sensors, Actuators, Optocoupler, Communication Interfaces (12C, SPI, IrDA, Bluetooth, Wi-Fi, Zigbee only)(Text 2: All the Topics from Ch—l and Ch-2 (Fig and explanation before 2.1)2.1.1.6 to 2.1.1.8, 2.2 to 2.2.2.3, 2.3 to 2.3.2, 2.3.3.3, selected topics of 2.4.1and 2.4.2 only).

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
1. A system which is a combination of a generic hardware and a General Purpose Operating System for executing a variety of applications	1. A system which is a combination of special purpose hardware and embedded OS for executing a specific set of applications
2. Contains a General Purpose Operating System (GPOS)	2. May or may not contain an operating system for functioning
3. Applications are alterable (programmable) by the user (It is	3. The firmware of the embedded system is pre-programmed and it is

possible for the end user to re-install the operating system, and also add or remove user applications)	non-alterable by the end-user (There may be exceptions for system supporting OS kernel image flashing through special hardware settings)
4. Performance is the key deciding factor in the selection of the system. Always, 'Faster is Better'	4. Application-specific requirements (like performance, power requirements, memory usage, etc.) are the key deciding factors
5. Less/not at all tailored towards reduced operating power requirements, options for different levels of power management	5. Highly tailored to take advantage of the power saving modes supported by the hardware and the operating system
6. Response requirements are not time-critical	6. For certain category of embedded systems like mission critical systems, the response time requirement is highly critical
7. Need not be deterministic in execution behaviour	7. Execution behaviour is deterministic for certain types of embedded systems like 'Hard Real Time' systems

History of Embedded Systems

- Embedded systems were in existence even before the IT revolution.
- Built around the old vacuum tube and transistor technologies.
- Advances in semiconductor and nanotechnology and IT revolution gave way to the development of miniature embedded systems.
- The first recognised modern embedded system is the Apollo Guidance Computer (AGC) developed by the MIT Instrumentation Laboratory for the lunar expedition.
- It had 36K words of fixed memory and 2K words of erasable memory.
- The clock frequency of was 1.024 MHz and it was derived from a 2.048 MHz crystal clock.
- The first mass-produced embedded system was the Autonetics D-17 guidance computer for the Minuteman-I missile in 1961.
- It was built using discrete transistor logic and a hard-disk for main memory.
- The first integrated circuit was produced in September 1958 and computers using them began to appear in 1963.

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

1. 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.

2. 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.

3. 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.

4.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.

5. 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

1. Small-Scale Embedded Systems
2. Medium-Scale Embedded Systems
3. Large-Scale Embedded Systems/Complex Systems

1. 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.

2. 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.

3. 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.
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.
12. Wearable Devices: Health and fitness trackers, Smartphone screen extension for notifications, etc.
13. Cloud Computing and Internet of Things (IoT)

Purpose of Embedded Systems

- Each embedded system is designed to serve the purpose of any one or a combination of the following tasks:
    1. Data Collection/Storage/Representation
    2. Data Communication

3. Data (Signal) Processing
4. Monitoring
5. Control
6. Application Specific User Interface

1. Data Collection/Storage/Representation

- Embedded systems designed for the purpose of data collection performs acquisition of data from the external world.
- Data collection is usually done for storage, analysis, manipulation and transmission.
- The term "data" refers all kinds of information, viz. text, voice, image, video, electrical signals and any other measurable quantities.
- Data can be either analog (continuous) or digital (discrete).
- The collected data may be stored or transmitted or it may be processed or it may be deleted instantly after giving a meaningful representation.

- A digital camera is a typical example of an embedded system with data collection/storage/representation of data.
- Images are captured and the captured image may be stored within the memory of the camera.
- The captured image can also be presented to the user through a graphic LCD unit.

2. Data Communication

- Embedded data communication systems are deployed in applications ranging from complex satellite communication systems to simple home networking systems.
- The transmission is achieved either by a wire- line medium or by a wireless medium.
- The data collecting embedded terminal itself can incorporate data communication units like wireless modules (Bluetooth, ZigBee, Wi-Fi, EDGE, GPRS, etc.) or wire-line modules (RS- 232C, USB, TCP/IP, PS2, etc.).

Fig 1: Fig: A wireless network router for data communication

- Network hubs, routers, switches, etc. are typical examples of dedicated data transmission embedded systems.
- They act as mediators in data communication and provide various features like data security, monitoring etc.

3. Data (Signal) Processing

- The data (voice, image, video, electrical signals and other measurable quantities) collected by embedded systems may be used for various kinds of data processing.
- Embedded systems with signal processing functionalities are employed in applications demanding signal processing like speech coding, synthesis, audio video codec, transmission applications, etc.

- A digital hearing aid is a typical example of an embedded system employing data processing.
- Digital hearing aid improves the hearing capacity of hearing impaired persons.

4. Monitoring

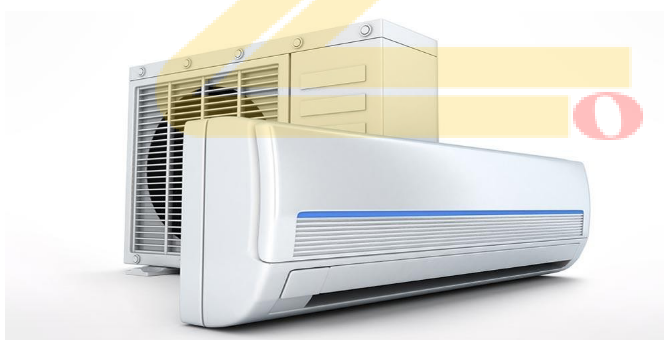
- Almost embedded products coming under the medical domain are used for monitoring.
- A very good example is the electro cardiogram (ECG) machine for monitoring the heartbeat of a patient.
- The machine is intended to do the monitoring of the heartbeat.
- It cannot impose control over the heartbeat.
- The sensors used in ECG are the different electrodes connected to the patient's body.

Fig: A patient monitoring system for monitoring heartbeat

- Some other examples of embedded systems with monitoring function are measuring instruments like digital CRO, digital multimeters, logic analyzers, etc. used in Control & Instrumentation applications.

5. Control

- Embedded systems with control functionalities impose control over some variables according to the changes in input variables.
- A system with control functionality contains both sensors and actuators.
- Sensors are connected to the input port for capturing the changes in environmental variable or measuring variable.
- The actuators connected to the output port are controlled according to the changes in input variable to put an impact on the controlling variable to bring the controlled variable to the specified range.



- An Air Conditioner System used to control the room temperature to a specified limit is a typical example for embedded system for control purpose.
- An air conditioner contains a room temperature-sensing element (sensor) which may be a thermistor and a handheld unit for setting up (feeding) the desired temperature.

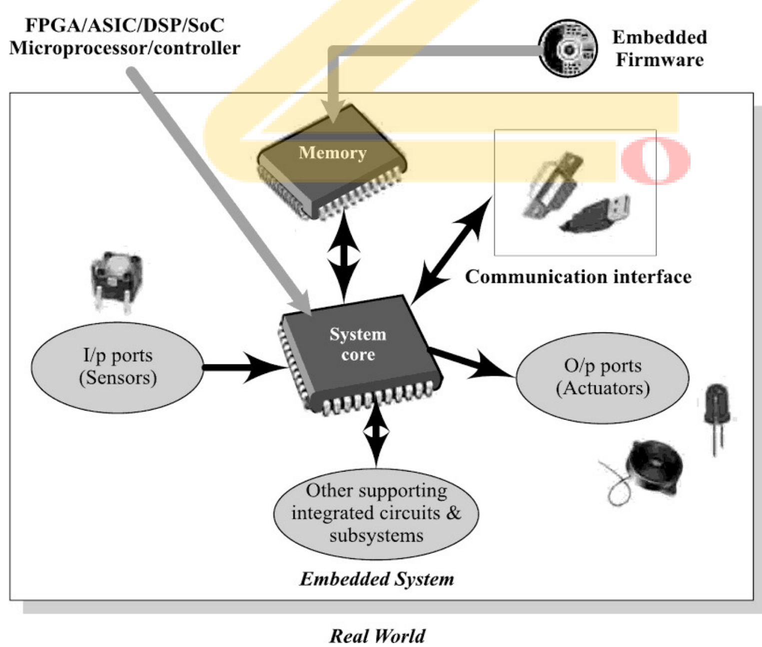
6. Application Specific User Interface

- These are embedded systems with application-specific user interfaces like buttons, switches, keypad, lights, bells, display units, etc.
- Mobile phone is an example for this.
- In mobile phone the user interface is provided through the keypad, graphic LCD module, system speaker, vibration alert, etc.

- A mobile phone is an example for embedded system with an application-specific user interfaces.

The Typical Embedded System

ELEMENTS OF AN EMBEDDED SYSTEM



Elements of an Embedded System

- 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)
- 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 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.
- 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.)

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
1. Lesser number of instructions	1. Greater number of instructions
2. Instruction pipelining and increased execution speed	2. Generally no instruction pipelining feature
3. Orthogonal instruction set (Allows each instruction to operate on any register and use any addressing mode)	3. Non-orthogonal instruction set (All instructions are not allowed to operate on any register and use any addressing mode. It is instruction-specific)
4. Operations are performed on registers only, the only memory operations are load and store	4. Operations are performed on registers or memory depending on the instruction
5. A large number of registers are available	5. Limited number of general purpose registers
6. Programmer needs to write more code to execute a task since the instructions are simpler ones	6. 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
7. Single, fixed length instructions	7. Variable length instructions
8. Less silicon usage and pin count	8. More silicon usage since more additional decoder logic is required to implement the complex instruction decoding
9. With Harvard Architecture	9. Can be Harvard or Von-Neumann Architecture

Harvard vs. Von-Neumann Processor/ControIIer Architecture

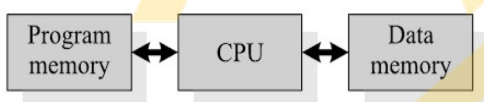
Von-Neumann Architecture

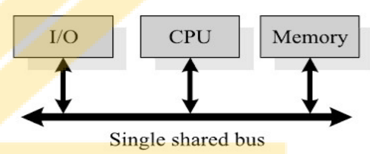
- Microprocessors/controllers based on the Von-Neumann architecture share a single common bus for fetching both instructions and data.
- Program instructions and data are stored in a common main memory.
- They first fetch an instruction and then fetch the data to support the instruction from code memory.
- 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 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 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





Big-Endian vs. Little-Endian Processors/Controllers

- Endianness specifies the order in which the data is stored in the memory by processor operations in a multi byte system.
- Suppose the word length is two byte then data can be stored in memory in two different ways:
    - Higher order of data byte at the higher memory and lower order of data byte at location just below the higher memory – Little-Endian
        - E.g.: Intel x86 Processors
    - Lower order of data byte at the higher memory and higher order of data byte at location just below the higher memory – Big-Endian
        - E.g.: Motorola 68000 Series Processors
- Little-endian means the lower-order byte of the data is stored in memory at the lowest address, and the higher-order byte at the highest address. (The little end comes first.)

- For example, a 4 byte long integer Byte3 Byte2 Byte1 Byte0 will be stored in the memory as shown below:

Base Address + 0	Byte 0	Byte 0	0×20000 (Base Address)
Base Address + 1	Byte 1	Byte 1	0×20001 (Base Address + 1)
Base Address + 2	Byte 2	Byte 2	0×20002 (Base Address + 2)
Base Address + 3	Byte 3	Byte 3	0×20003 (Base Address + 3)

- Big-endian means the higher-order byte of the data is stored in memory at the lowest address, and the lower-order byte at the highest address. (The big end comes first.)
- For example, a 4 byte long integer Byte3 Byte2 Byte1 Byte0 will be stored in the memory as shown below:

Base Address + 0	Byte 3	Byte 3	0×20000 (Base Address)
Base Address + 1	Byte 2	Byte 2	0×20001 (Base Address + 1)
Base Address + 2	Byte 1	Byte 1	0×20002 (Base Address + 2)
Base Address + 3	Byte 0	Byte 0	0×20003 (Base Address + 3)

- **<u>Consider data 43675829H</u>**

Base Address + 0	Byte 0	Byte 0	0×20000 (Base Address)
Base Address + 1	Byte 1	Byte 1	0×20001 (Base Address + 1)
Base Address + 2	Byte 2	Byte 2	0×20002 (Base Address + 2)
Base Address + 3	Byte 3	Byte 3	0×20003 (Base Address + 3)

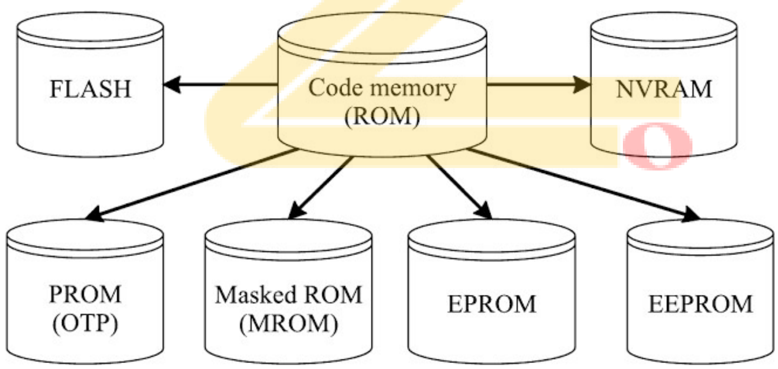
Little-Endian

Base Address + 0	Byte 3	Byte 3	0×20000 (Base Address)
Base Address + 1	Byte 2	Byte 2	0×20001 (Base Address + 1)
Base Address + 2	Byte 1	Byte 1	0×20002 (Base Address + 2)
Base Address + 3	Byte 0	Byte 0	0×20003 (Base Address + 3)

Big-Endian

Memory

- Memory is an important part of a processor/controller based embedded systems.
- Some of the processors/controllers contain built in memory and this memory is referred as on-chip memory.
- Others do not contain any memory inside the chip and requires external memory to be connected with the controller/processor to store the control algorithm. It is called off-chip memory.
- Also some working memory is required for holding data temporarily during certain operations.
- The program memory or code storage memory of an embedded system stores the program instructions.
- The code memory retains its contents even after the power is turned off. It is generally known as non-volatile storage memory.
- It can be classified into different types as shown:



1. Masked ROM (MROM)

- Masked ROM is a one-time programmable device.
- Masked ROM makes use of the hardwired technology for storing data.
- The device is factory programmed by masking and metallisation process at the time of production itself, according to the data provided by the end user.
- Advantage – low cost for high volume production.
- Limitation – inability to modify the device firmware against firmware upgrades.

- Since the MROM is permanent in bit storage, it is not possible to alter the bit information.
- Different mechanisms are used for the masking process of the ROM, like
- Creation of an enhancement or depletion mode transistor through channel implant.
- By creating the memory cell either using a standard transistor or a high threshold transistor.
- In the high threshold mode, the supply voltage required to turn ON the transistor is above the normal ROM IC operating voltage.
- This ensures that the transistor is always off and the memory cell stores always logic 0.

2. Programmable Read Only Memory (PROM) / (OTP)

- One Time Programmable Memory (OTP) or PROM is not pre-programmed by the manufacturer.
- The end user is responsible for programming these devices.
- This memory has nichrome or polysilicon wires arranged in a matrix. These wires can be functionally viewed as fuses.
- It is programmed by a PROM programmer which selectively burns the fuses according to the bit pattern to be stored.
- Fuses which are not blown/burned represents a logic “1" whereas fuses which are
- blown/burned represents a logic 0 .
- The default state is logic "1".
- OTP is widely used for commercial production of embedded systems whose prototyped versions are proven and the code is finalised.
- It is a low cost solution for commercial production.
- OTPs cannot be reprogrammed.

3. Erasable Programmable Read Only Memory (EPROM)

- Erasable Programmable Read Only Memory (EPROM) gives the flexibility to re-program the same chip.
- EPROM stores the bit information by charging the floating gate of an FET.
- Bit information is stored by using an EPROM programmer, which applies high voltage to charge the floating gate.
- EPROM contains a quartz crystal window for erasing the stored information.
- If the window is exposed to ultraviolet rays for a fixed duration, the entire memory will be erased.
- Even though the EPROM chip is flexible in terms of re-programmability, it needs to be taken out of the circuit board and put in a UV eraser device for 20 to 30 minutes.
- It is a tedious and time-consuming process.

4. Electrically Erasable Programmable Read only Memory (EEPROM)

- The information contained in the EEPROM memory can be altered by using electrical signals at the register/byte level.
- They can be erased and reprogrammed in-circuit.
- These chips include a chip erase mode and in this mode they can be erased in a few milliseconds.
- It provides greater flexibility for system design.
- The only limitation is their capacity is limited (a few kilobytes) when compared with the standard ROM.

5. FLASH

- FLASH memory is a variation of EEPROM technology – It combines the re-programmability of EEPROM and the high capacity of standard ROMs.
- FLASH is the latest ROM technology.
- Most popular ROM technology used in today's embedded designs.
- FLASH memory is organised as sectors (blocks) or pages.
- FLASH memory stores information in an array of floating gate MOSFET transistors.
- The erasing of memory can be done at sector level or page level without affecting the other sectors or pages.
- Each sector/page should be erased before re-programming.
- The typical erasable capacity of FLASH is of the order of a few thousand cycles.
- E.g.: SST39LF010 from Microchip is an example of 1Mbit (organised as 128Kx8) FLASH
- memory with typical endurance of 100,000 cycles.

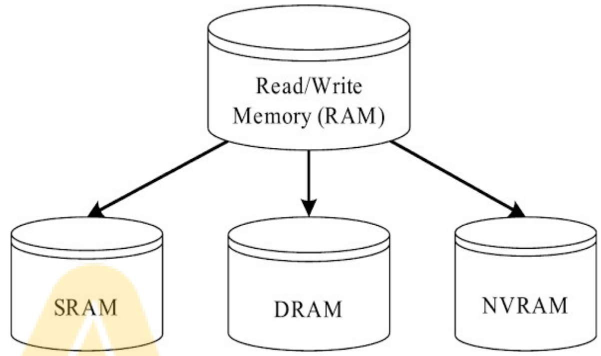
6. Non-Volatile RAM (NVRAM)

- Non-volatile RAM is a random access memory with battery backup.
- It contains static RAM based memory and a minute battery for providing supply to the memory in the absence of external power supply.
- The memory and battery are packed together in a single package.
- The life span of NVRAM is expected to be around 10 years.
- E.g.: DS1644 from Maxim/Dallas is an example of 32KB NVRAM.

7. Read-Write Memory/Random Access Memory (RAM)

- RAM is the data memory or working memory of the controller/processor.
- Controller/processor can read from it and write to it.
- RAM is volatile – when the power is turned off, all the contents are destroyed.
- RAM is a direct access memory – we can access the desired memory location directly without the need for traversing through the entire memory locations to reach the desired memory position (i.e. random access of memory location).

- This is in contrast to the Sequential Access Memory (SAM), where the desired memory location is accessed by either traversing through the entire memory or through a 'seek' method. Magnetic tapes, CD ROMs, etc. are examples of sequential access memories.
- RAM generally falls into three categories: Static RAM (SRAM), Dynamic RAM (DRAM) and Non-Volatile RAM (NVRAM).



I. Static RAM (SRAM)

- Static RAM stores data in the form of voltage.
- They are made up of flip-flops.
- Static RAM is the fastest form of RAM available.
- Fast due to its resistive networking and switching capabilities.
- In typical implementation, an SRAM cell (bit) is realised using six transistors (or 6 MOSFETs).
- Four of the transistors are used for building the latch (flip-flop) part of the memory cell and two for controlling the access.
- In its simplest representation an SRAM cell can be visualised as shown in the figure below:

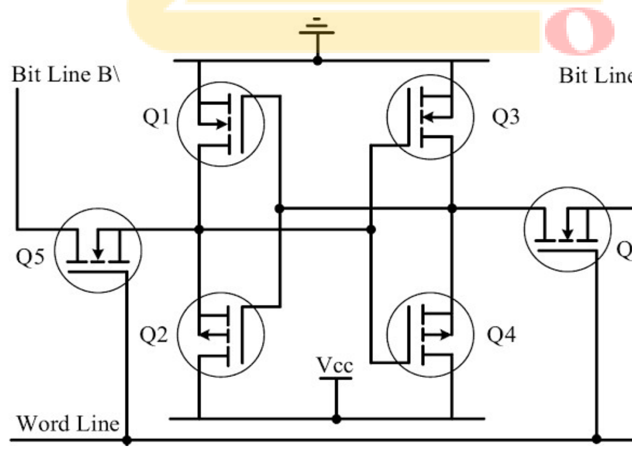


Fig: SRAM cell implementation

- This implementation in its simpler form can be visualised as two cross-coupled inverters with read/write control through transistors.

- The four transistors in the middle form the cross-coupled inverters.
- This can be visualised as shown in the figure below:

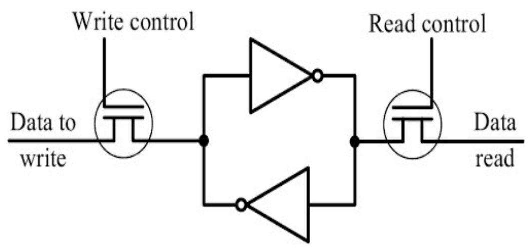


Fig: Visualisation of SRAM cell

- The access to the memory cell is controlled by Word Line, which controls the access transistors (MOSFETs) Q5 and Q6.
- The access transistors control the connection to bit lines B & B\.
- In order to write a value to the memory cell, apply the desired value to the bit control lines (For writing 1, make B = 1 and B\ = 0; For writing 0, make B = 0 and B\ = 1) and assert the Word Line (Make Word line high).
- This operation latches the bit written in the flip-flop.
- For reading the content of the memory cell, assert both B and B\ bit lines to 1 and set the Word line to 1.
- The major limitations of SRAM are low capacity and high cost.

II. Dynamic RAM (DRAM)

- Dynamic RAM stores data in the form of charge.
- They are made up of MOS transistor gates.
- Advantages – high density and low cost compared to SRAM.
- Disadvantage – since the information is stored as charge it gets leaked off with time and to prevent this they need to be refreshed periodically.
- Special circuits called DRAM controllers are used for the refreshing operation.
- The refresh operation is done periodically in milliseconds interval.
- Figure below illustrates the typical implementation of a DRAM cell.
- The MOSFET acts as the gate for the incoming and outgoing data whereas the capacitor acts as the bit storage unit.

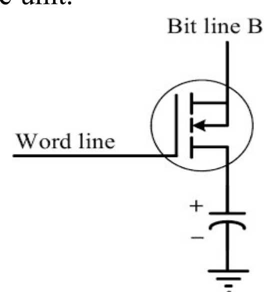


Fig: DRAM cell implementation

SRAM vs DRAM

- Table given below summarises the relative merits and demerits of SRAM and DRAM technology.

SRAM Cell	DRAM Cell
Made up of 6 CMOS transistors (MOSFET)	Made up of a MOSFET and a capacitor
Doesn't require refreshing	Requires refreshing
Low capacity (Less dense)	High capacity (Highly dense)
More expensive	Less expensive
Fast in operation. Typical access time is 10ns	Slow in operation due to refresh requirements. Typical access time is 60ns. Write operation is faster than read operation.

Non-Volatile RAM (NVRAM)

- Non-volatile RAM is a random access memory with battery backup.
- It contains static RAM based memory and a minute battery for providing supply to the memory in the absence of external power supply.
- The memory and battery are packed together in a single package.
- The life span of NVRAM is expected to be around 10 years.
- E.g.: DS1644 from Maxim/Dallas is an example of 32KB NVRAM.

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.
- 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

Optocoupler

- Optocoupler is a solid state device to isolate two parts of a circuit.
- Optocoupler combines an LED and a photo-transistor in a single housing.
- Figure illustrates the functioning of an optocoupler device.

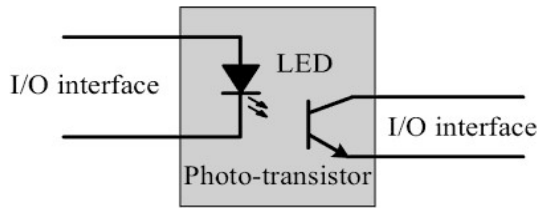


Fig: An optocoupler device

- In electronic circuits, an optocoupler is used for suppressing interference in data communication, circuit isolation, high voltage separation, simultaneous separation and signal intensification, etc.
- Optocouplers can be used in either input circuits or in output circuits.
- Optocoupler is available as ICs from different semiconductor manufacturers.
- The MCT2M IC from Fairchild semiconductor is an example for optocoupler IC.
- Figure illustrates the usage of optocoupler in input circuit and output circuit of an embedded system with a microcontroller as the system core.

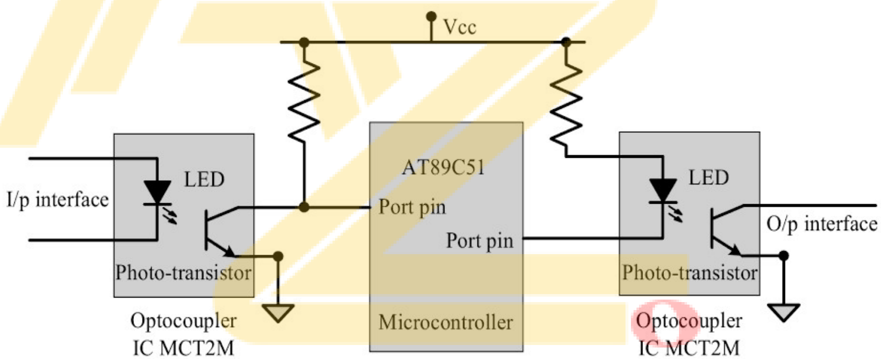


Fig: Optocoupler in Input and Output circuit

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

Inter Integrated Circuit (I2C) Bus

- The Inter Integrated Circuit Bus (I2C or I^2C Pronounced 'I square C') is a synchronous bi-directional half duplex two wire serial interface bus.
    - (Half duplex - one-directional communication at a given point of time)
- The concept of I2C bus was developed by Philips Semiconductors in the early 1980s.
- The original intention of I2C was to provide an easy way of connection between a microprocessor/microcontroller system and the peripheral chips in television sets.
- The I2C bus comprise of two bus lines:
    - Serial Clock (SCL line) – responsible for generating synchronisation clock pulses
    - Serial Data (SDA line) – responsible for transmitting the serial data across devices
- I2C bus is a shared bus system to which many number of I2C devices can be connected.
- Devices connected to the I2C bus can act as either 'Master' or 'Slave'.
    - The 'Master' device is responsible for controlling the communication by initiating/terminating data transfer, sending data and generating necessary synchronisation clock pulses.
    - 'Slave' devices wait for the commands from the master and respond upon receiving the commands.
- 'Master' and 'Slave' devices can act as either transmitter or receiver.
- Regardless whether a master is acting as transmitter or receiver, the synchronisation clock signal is generated by the 'Master' device only.
- I2C supports multi masters on the same bus.
- The following bus interface diagram illustrates the connection of master and slave devices on the I2C bus.

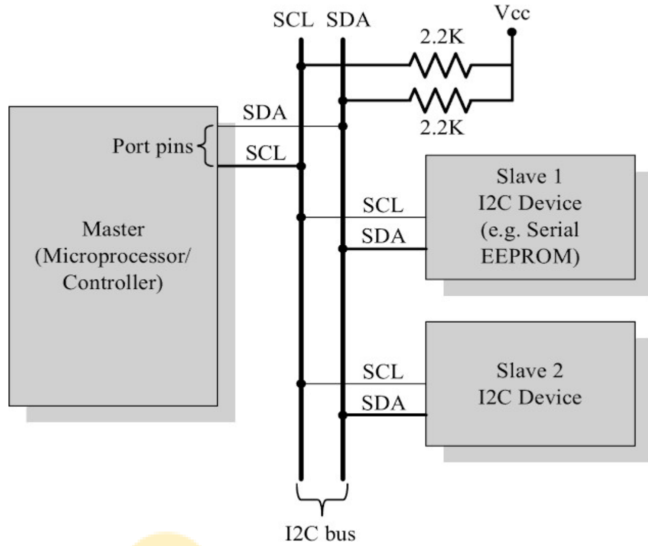


Fig: I2C Bus Interfacing

- The I2C bus interface is built around an input buffer and an open drain or collector transistor.
- When the bus is in the idle state, the open drain/collector transistor will be in the floating state and the output lines (SDA and SCL) switch to the 'High Impedance' state.
- For proper operation of the bus, both the bus lines should be pulled to the supply voltage (+5 V for TTL family and +3.3V for CMOS family devices) using pull-up resistors.
  - The typical value of resistors used in pull-up is 2.2K.
  - With pull-up resistors, the output lines of the bus in the idle state will be 'HIGH'
- The address of a I2C device is assigned by hardwiring the address lines of the
- device to the desired logic level.
  - Done at the time of designing the embedded hardware.
- The sequence of operations for communicating with an I2C slave device is listed below:

  1. The master device pulls the clock line (SCL) of the bus to 'HIGH'
  2. The master device pulls the data line (SDA) 'LOW', when the SCL line is at logic 'HIGH' (This is the 'Start' condition for data transfer)
  3. The master device sends the address (7 bit or 10 bit wide) of the 'slave' device to which it wants to communicate, over the SDA line.
     - Clock pulses are generated at the SCL line for synchronising the bit reception by the slave device.
     - The MSB of the data is always transmitted first.

- The data in the bus is valid during the 'HIGH' period of the clock signal

4. The master device sends the Read or Write bit (Bit value = 1 Read operation; Bit value = 0 Write operation) according to the requirement
5. The master device waits for the acknowledgement bit from the slave device whose address is sent on the bus along with the Read/ Write operation command.
    - Slave devices connected to the bus compares the address received with the address assigned to them
6. The slave device with the address requested by the master device responds by sending an acknowledge bit (Bit value 1) over the SDA line
7. Upon receiving the acknowledge bit, the Master device sends the 8 bit data to the slave device over SDA line, if the requested operation is 'Write to device'.
    - If the requested operation is 'Read from device', the slave device sends data to the master over the SDA line
8. The master device waits for the acknowledgement bit from the device upon byte transfer complete for a write operation and sends an acknowledge bit to the Slave device for a read operation
9. The master device terminates the transfer by pulling the SDA line 'HIGH' when the clock line SCL is at logic 'HIGH' (Indicating the 'STOP' condition)

- First generation I2C devices were designed to support data rates only up to 100kbps.
- Over time, there have been several additions to the specification so that there are now five operating speed categories:
    - Standard mode (Sm – Data rate up to 100kbit/sec)
    - Fast mode (Fm – Data rate up to 400kbit/sec)
    - Fast mode Plus (Fm+ – Data rate up to 1Mbit/sec)
    - High speed mode (Hsm – Data rate up to 3.4Mbit/sec)
    - Ultra Fast mode (UFm – Data rate up to 5Mbit/sec)

Serial Peripheral Interface (SPI) Bus

- The Serial Peripheral Interface Bus (SPI) is a synchronous bi-directional full duplex four-wire serial interface bus.
- The concept of SPI was introduced by Motorola.
- SPI is a single master multi-slave system.

  - There can be more than one masters, but only one master device can be active at any given point of time.
- SPI requires four signal lines for communication. They are:
  - Master Out Slave In (MOSI) – Signal line carrying the data from master to slave device. It is also known as Slave Input/SIave Data In (SI/SDI)
  - Master In Slave Out (MISO) – Signal line carrying the data from slave to master device. It is also known as Slave Output (SO/SDO)
  - Serial Clock (SCLK) – Signal line carrying the clock signals
  - Slave Select (SS) – Signal line for slave device select. It is an active low signal
- The bus interface diagram shown in the figure illustrates the connection of master and slave devices on the SPI bus.

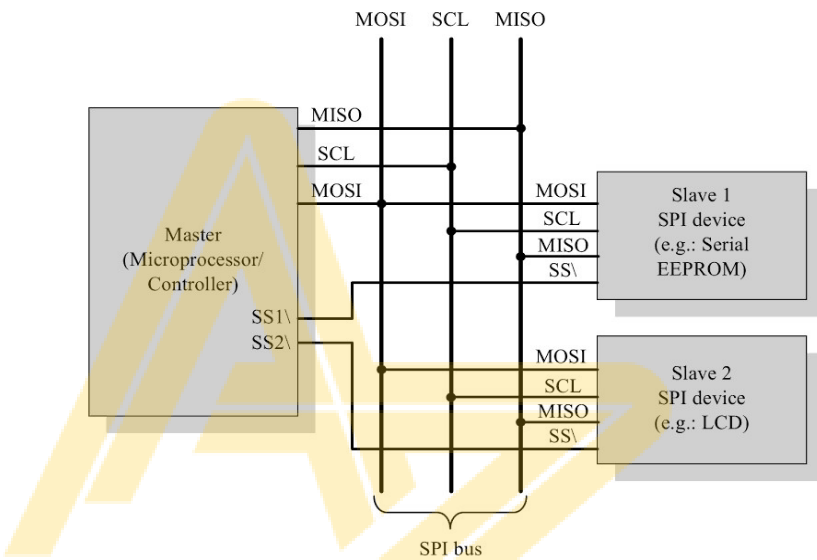


Fig: SPI Bus Interfacing

- The master device is responsible for generating the clock signal.
- It selects the required slave device by asserting the corresponding slave
- device's slave select signal 'LOW'.
- The data out line (MISO) of all the slave devices when not selected floats at high impedance state.
- The serial data transmission through SPI bus is fully configurable.
  - SPI devices contain a certain set of registers for holding these configurations.
  - The control register holds the various configuration parameters like master/slave selection for the device, baud rate selection for communication, clock signal control, etc.
  - The status register holds the status of various conditions for transmission and reception.
- SPI works on the principle of 'Shift Register'.
- The master and slave devices contain a special shift register for the data to transmit or receive.

  - The size of the shift register is device dependent. Normally it is a multiple of 8.

- During transmission from the master to slave, the data in the master's shift register is shifted out to the MOSI pin and it enters the shift register of the slave device through the MOSI pin of the slave device.
- At the same time the shifted out data bit from the slave device's shift register enters the shift register of the master device through MISO pin.
- In summary, the shift registers of 'master' and 'slave' devices form a circular buffer.
- When compared to I2C, SPI bus is most suitable for applications requiring transfer of data in 'streams'.
- The only limitation is SPI doesn't support an acknowledgement mechanism.

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.

Infrared (IrDA)

- Infrared (IrDA) is a serial, half duplex, line of sight based wireless technology
- for data communication between devices.
- It is in use from the olden days of communication and you may be very familiar with it.
- E.g.: The remote control of TV, VCD player, etc. works on Infrared.
- Infrared communication technique uses infrared waves of the electromagnetic
- spectrum for transmitting the data.
- It supports point-point and point-to-multipoint communication, provided all
- devices involved in the communication are within the line of sight.
- The typical communication range for IrDA lies in the range 10 cm to 1 m.
- The range can be increased by increasing the transmitting power of the IR device.
- IR supports data rates ranging from 9600bits/second to 16Mbps.

- Depending on the speed of data transmission IR is classified into:
    - Serial IR (SIR) – supports data rates ranging from 9600bps to 115.2kbps.
    - Medium IR (MIR) – supports data rates of 0.576Mbps and 1.152Mbps.
    - Fast IR (FIR) – supports data rates up to 4Mbps.
    - Very Fast IR (VFIR) – supports data rates up to 16Mbps.
    - Ultra Fast IR (UFIR) – supports data rates up to 96Mbps.
    - GigaIR – supports data rates 512 Mbps to 1 Gbps.
- IrDA communication involves a transmitter unit for transmitting the data over IR and a receiver for receiving the data.
- Infrared Light Emitting Diode (LED) is the IR source for transmitter and at the receiving end a photodiode acts as the receiver.
- Both transmitter and receiver unit will be present in each device supporting IrDA communication for bidirectional data transfer. Such IR units are known as 'Transceiver'.
- Certain devices like a TV remote control always require unidirectional communication and so they contain either the transmitter or receiver unit.
- The remote control unit contains the transmitter unit and TV contains the receiver unit.
- Infrared Data Association (IrDA) is the regulatory body responsible for defining and licensing the specifications for IR data communication.
- IR communication has two essential parts: a physical link part and a protocol part.
- The physical link is responsible for the physical transmission of data between devices supporting IR communication
- Protocol part is responsible for defining the rules of communication.
- The physical link works on the wireless principle making use of Infrared for communication.
- The IrDA specifications include the standard for both physical link and protocol layer.
- The IrDA control protocol contains implementations for Physical Layer (PHY), Media Access Control (MAC) and Logical Link Control (LLC).
- IrDA is a popular interface for file exchange and data transfer in low cost devices.

- IrDA was the prominent communication channel in mobile phones before Bluetooth's existence.

Bluetooth (BT)

- Bluetooth is a low cost, low power, short range wireless technology for data and voice communication.
- Bluetooth was first proposed by Ericsson in 1994.
- Bluetooth operates at 2.4GHz of the Radio Frequency spectrum and uses the Frequency Hopping Spread Spectrum (FHSS) technique for communication.
- Bluetooth supports a data rate of up to 1Mbps to 24Mbps and a range of approximately 30 to 100 feet for data communication (depending on the version)
  - v1.2 supports data rate up to 1Mbps
  - v2.0 + EDR supports data rate up to 3Mbps
  - v3.0 + HS and v4.0 supports data rate up to 24Mbps
- Bluetooth communication has two essential parts – a physical link part and a protocol part.
  - The physical link is responsible for the physical transmission of data between devices supporting Bluetooth communication
  - The protocol part is responsible for defining the rules of communication.
- The physical link works on the wireless principle making use of RF waves for communication.
- Bluetooth enabled devices essentially contain a Bluetooth wireless radio for the transmission and reception of data.
- The rules governing the Bluetooth communication is implemented in the 'Bluetooth protocol stack'.
- The Bluetooth communication IC holds the stack.
- Each Bluetooth device will have a 48 bit unique identification number.
- Bluetooth communication follows packet based data transfer.
- Bluetooth supports point-to-point (device to device) and point-to-multipoint (device to multiple device broadcasting) wireless communication.
- The point-to-point communication follows the master-slave relationship.
- A Bluetooth device can function as either master or slave.
- When a network is formed with one Bluetooth device as master and more than one device as slaves, it is called a Piconet.
- A Piconet supports a maximum of seven slave devices.
- Bluetooth is the favourite choice for short range data communication in handheld embedded devices.
- Bluetooth technology is very popular among cell phone users as they are the easiest communication channel for transferring ringtones, music files, pictures, media files, etc. between neighbouring Bluetooth enabled phones.
- Bluetooth Low Energy (BLE)/Bluetooth Smart is a latest addition to the Bluetooth technology.

- BLE allows devices to use much less power compared to the standard Bluetooth connections, while offering most of the connectivity of Bluetooth and maintaining a similar communication range.
- Bluetooth 4.2 specification enables IoT support through low-power IP connectivity, with support for flexible Internet connectivity options (IPv6/6LoWPAN or Bluetooth Smart Gateways) and implements industry-leading privacy, power efficiency and industry standard security.
- The Bluetooth standard specifies the minimum requirements that a Bluetooth device must support for a specific usage scenario.
- The Generic Access Profile (GAP) defines the requirements for detecting a Bluetooth device and establishing a connection with it.
- All other specific usage profiles are based on GAP.
- Serial Port Profile (SPP) for serial data communication, File Transfer Profile (FTP) for file transfer between devices, Human Interface Device (HID) for supporting human interface devices like keyboard and mouse are examples for Bluetooth profiles.
- BLE implements various application specific profiles for communicating with low power Bluetooth peripherals like fitness devices, blood pressure and heart rate monitors, etc.
    - Examples:
        - Healthcare Profiles
            - HTP – Health Thermometer Profile, GLP – Glucose Profile, BLP – Blood Pressure Profile, etc.
        - Sports and Fitness Profiles
            - HRP – Heart Rate Profile, LNP – Location and Navigation Profile, RSCP – Running Speed and Cadence Profile, etc.
- The specifications for Bluetooth communication is defined and licensed by the standards body 'Bluetooth Special Interest Group (SIG)'.

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

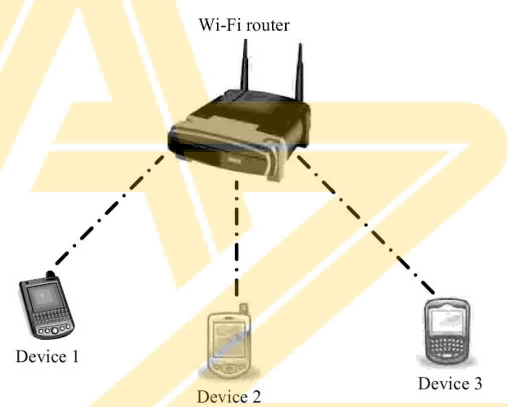


Fig: Wi-Fi Network

- 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.

ZigBee

- ZigBee is a low power, low cost, wireless network communication protocol based on the IEEE 802.15.4-2006 standard.
- ZigBee is targeted for low power, low data rate and secure applications for Wireless Personal Area Networking (WPAN).
- The ZigBee specifications support a robust mesh network containing multiple nodes.
- This networking strategy makes the network reliable by permitting messages to travel through a number of different paths to get from one node to another.
- ZigBee operates worldwide at the unlicensed bands of Radio spectrum, mainly at 2.400 to 2.484 GHz, 902 to 928 MHz and 868.0 to 868.6 MHz.
- ZigBee supports an operating distance of up to 100 metres and a data rate of 20 to 250 Kbps.
- In the ZigBee terminology, each ZigBee device falls under any one of the following ZigBee device category:
- ZigBee Coordinator (ZC)/Network Coordinator
    - The ZigBee coordinator acts as the root of the ZigBee network.
    - The ZC is responsible for initiating the ZigBee network and it has the capability to store information about the network.
- ZigBee Router (ZR)/Full function Device (FFD)
    - Responsible for passing information from device to another device or to another ZR.
- ZigBee End Device (ZED)/Reduced Function Device (RFD):
    - End device containing ZigBee functionality for data communication.
    - It can talk only with a ZR or ZC and doesn't have the capability to act as a mediator for transferring data from one device to another.
- The diagram shown in figure gives an overview of ZC, ZED and ZR in a ZigBee network.

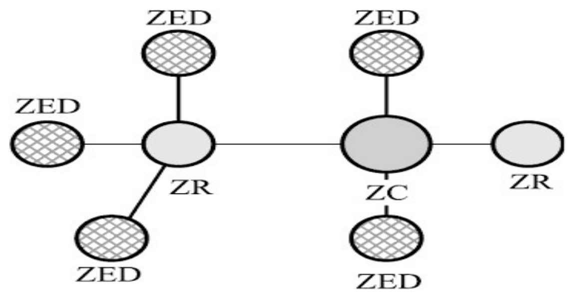


Fig: A ZigBee network model

- ZigBee is primarily targeting application areas like home & industrial automation, energy management, home control/security, medical/patient tracking, logistics & asset tracking and sensor networks & active RFID.
- Automatic Meter Reading (AMR), smoke detectors, wireless telemetry, HVAC control, heating control, lighting controls, environmental controls, etc. are examples for applications which can make use of the ZigBee technology.
- ZigBee PRO offers full wireless mesh, low-power networking capable of supporting more than 64,000 devices on a single network.
    - It provides standardized networking designed to connect the widest range of devices, in any industry, into a single control network.
    - ZigBee PRO offers an optional new and innovative feature, 'Green Power' to connect energy harvesting for self-powered devices into ZigBee PRO networks.
    - The 'Green Power' feature of ZigBee PRO is the most eco-friendly way to power battery-less devices such as sensors, switches, dimmers and many other devices and allows them to securely join ZigBee PRO networks.
- ZigBee 3.0 delivers all the features of ZigBee while unifying the ZigBee application standards found in ZigBee devices.
    - ZigBee 3.0 standard enables communication and interoperability among devices for smart homes, connected lighting and other markets.
- The specifications for ZigBee is developed and managed by the ZigBee Alliance, a non-profit consortium of leading semiconductor manufacturers, technology providers, OEMs and end-users worldwide.

References

1. Shibu K V, ***"Introduction to Embedded Systems"***, Tata McGraw Hill, 2009.
2. Raj Kamal, ***"Embedded Systems: Architecture and Programming",*** Tata McGraw Hill, 2008.