GANESH COLLEGE OF ENGINEERING, SALEM

BM3551-EMBEDDED SYSTEM AND IOMT NOTES

V- SEMESTER-BME (R-2021)

DEPARTMENT OF BIOMEDICAL ENGINEERING

Prepared By

Mr.B.VINOD B.E.,M.E.,(Ph.D)

Assistant Professor & BME

UNIT I: INTRODUCTION TO EMBEDDED SYSTEM DESIGN

1.1 Overview of Embedded Application Architecture

Embedded systems, an emerging area of computer technology, combine multiple technologies, such as computers, semiconductors, microelectronics, and the Internet, and as a result, are finding ever-increasing application in our modern world. With the rapid development of computer and communications technologies and the growing use of the Internet, embedded systems have brought immediate success and widespread application in the post-PC era, especially as the core components of the Internet of Things. They penetrate into every corner of modern life from the mundane, such as an automated home thermostat, to industrial production, such as in robotic automation in manufacturing. Embedded systems can be found in military and national defense, healthcare, science, education, and commercial services, and from mobile phones, MP3 players, and PDAs to cars, planes, and missiles.

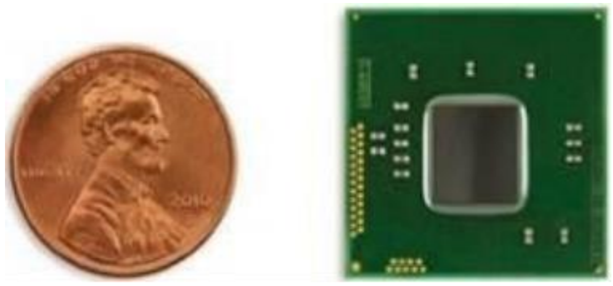
This chapter provides the concepts, structure, and other basic information about embedded systems and lays a theoretical foundation for embedded application development, of which application development for Android OS is becoming the top interest of developers.

1.2 Introduction to Embedded Systems

Since the advent of the first computer, the ENIAC, in 1946, the computer manufacturing process has gone from vacuum tubes, transistors, integrated circuits, and large-scale integration (LSI), to very-large-scale integration (VLSI), resulting in computers that are more compact, powerful, and energy efficient but less expensive (per unit of computing power).

After the advent of microprocessors in the 1970s, the computer-using world witnessed revolutionary change. Microprocessors are the basis of microcomputers, and personal computers (PCs) made them more affordable and practical, allowing many private users to own them. At this stage, computers met a variety of needs: they were sufficiently versatile to satisfy various demands such as computing, entertainment, information sharing, and office automation. As the adoption of microcomputers was occurring, more people wanted to embed them into specific systems to intelligently control the environment. For example, microcomputers were used in machine tools in factories. They were used to control signals and monitor the operating state through the configuration of peripheral sensors. When microcomputers were embedded into such environments, they were prototypes of embedded systems.

As the technology advanced, more industries demanded special computer systems. As a result, the development direction and goals of specialized computer systems for specific environments and general-purpose computer systems grew apart. The technical requirement of general-purpose computer systems is fast, massive, and diversified computing, whereas the goal of technical development is faster computing speed and larger storage capacity. However, the technical requirement of embedded computer systems is targeted more toward the intelligent control of targets, whereas the goal of technical development is embedded performance, control, and reliability closely related to the target system.



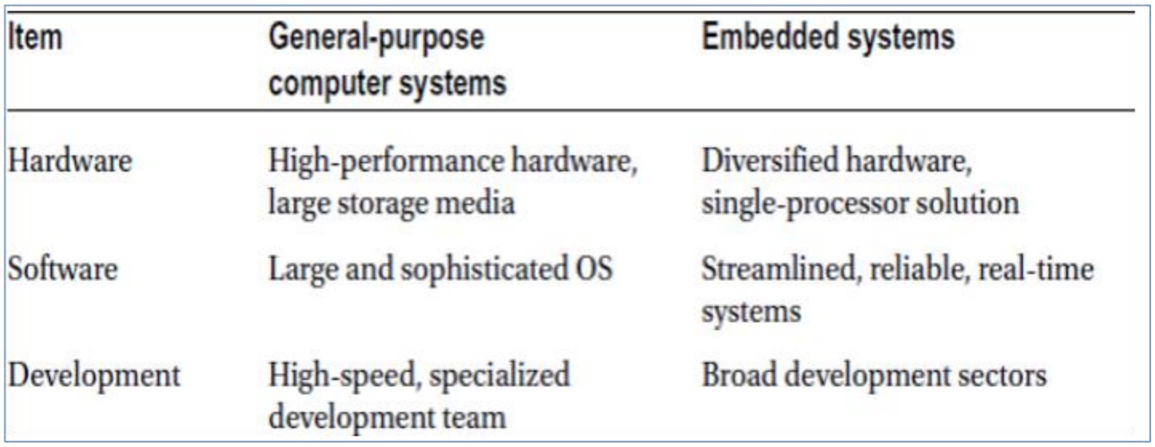
Embedded computing systems evolved in a completely different way. By emphasizing the characteristics of a particular processor, they turned traditional electronic systems into modern intelligent electronic systems. Figure 1-1 shows an embedded computer processor, the Intel Atom N2600 processor, which is 2.2 × 2.2 cm, alongside a penny.

Figure 1.1: Comparison of an embedded computer chip to a US penny.

The emergence of embedded computer systems alongside general-purpose computer systems is a milestone of modern computer technologies. The comparison of general-purpose computers and embedded systems is shown in Table 1-1.

Today, embedded systems are an integral part of people's lives due to their mobility. As mentioned earlier, they are used everywhere in modern life. Smartphones are a great example of embedded systems.

Table 1-1. Comparison of General-Purpose Computers and Embedded Systems



Item	General-purpose computer systems	Embedded systems
Hardware	High-performance hardware, large storage media	Diversified hardware, single-processor solution
Software	Large and sophisticated OS	Streamlined, reliable, real-time systems
Development	High-speed, specialized development team	Broad development sectors

1.2.1 Mobile Phones

Mobile equipment, especially smartphones, is the fastest growing embedded sector in recent years. Many new terms such as *extensive embedded development* and *mobile development* have been derived from mobile software development. Mobile phones not only arepervasive but also have powerful functions, affordable prices, and diversified applications. In addition to basic telephone functions, they include, but are not limited to, integrated PDAs, digital cameras, game consoles, music players, and wearables.

1.2.2 Consumer Electronics and Information Appliances

Consumer electronics and information appliances are additional big application sectors for embedded systems. Devices that fall into this category include personal mobile devices and home/entertainment/audiovisual devices. Personal mobile devices usually include smart handsets such as PDAs, as well as wireless Internet access equipment like mobile Internet devices (MIDs). In theory, smartphones are also in this class; but due to their large number, theyare listed as a single sector.

Home/entertainment/audiovisual devices mainly include network television like interactive television; digital imaging equipment such as digital cameras, digital photo frames, and video players; digital audio and video devices such as MP3 players and other portable audio players; and electronic entertainment devices such as handheld game consoles, PS2 consoles, and so on. Tablet PCs (tablets), one of the newer types of embedded devices, have become favorites of consumers since Apple released the iPad in 2010.

1.3 General Architecture of an Embedded System

Figure 1-2 shows a configuration diagram of a typical embedded system consisting of two main parts: embedded hardware and embedded software. The embedded hardware primarily includes the processor, memory, bus, peripheral devices, I/O ports, and various controllers. The embedded software usually contains the embedded operating system and various applications. Input and output are characteristics of any open system, and the embedded system is no exception. In the embedded system, the hardware and software often collaborate to deal with various input signals from the outside and output the processing results through some form.

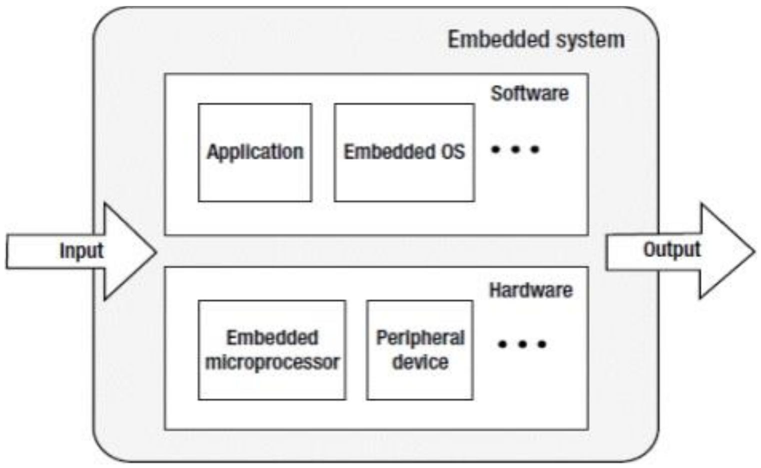


Figure 1.2: Basic architecture of an embedded system

The input signal may be an ergonomic device (such as a keyboard, mouse, or touch screen) or the output of a sensor circuit in another embedded system. The output may be in the form of sound, light, electricity, or another analog signal, or a record or file for a database.

The basic computer system components—microprocessor, memory, and input and output modules are interconnected by a system bus in order for all the parts to communicate and execute a program (see Figure 1-3).

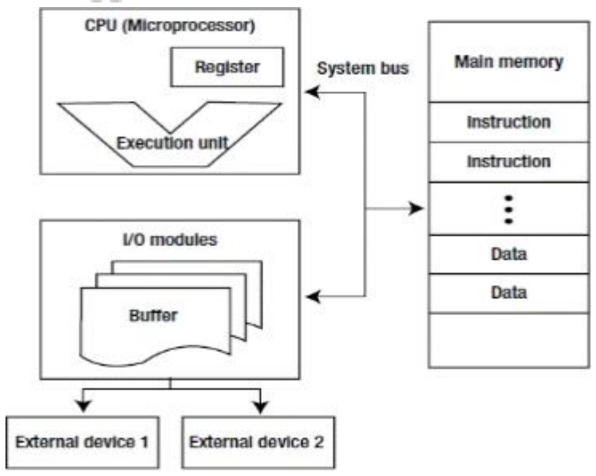


Figure 1.3: Hardware architecture of Embedded System

In embedded systems, the microprocessor's role and function are usually the same as those of the CPU in a general-purpose computer: control computer operation, execute instructions, and process data. In many cases, the microprocessor in an embedded system is also called the CPU. Memory is used to store instructions and data. I/O modules are responsible for the data exchange between the processor, memory, and external devices.

External devices include secondary storage devices (such as flash and hard disk), communications equipment, and terminal equipment. The system bus provides data and controls signal communication and transmission for the processor, memory, and I/O modules.

There are basically two types of architecture that apply to embedded systems: Von Neumann architecture and Harvard architecture. In a Von-Neumann architecture, the same memory and bus are used to store both data and instructions that run the program. Since you cannot access program memory and data memory simultaneously, the Von Neumann architecture is susceptible to bottlenecks and system performance is affected.

1.3.1 Von Neumann Architecture

Von Neumann architecture (also known as Princeton architecture) was first proposed by John von Neumann. The most important feature of this architecture is that the software and data use the same memory: that is, "The program is data, and the data is the program" (as shown in Figure 1-4).

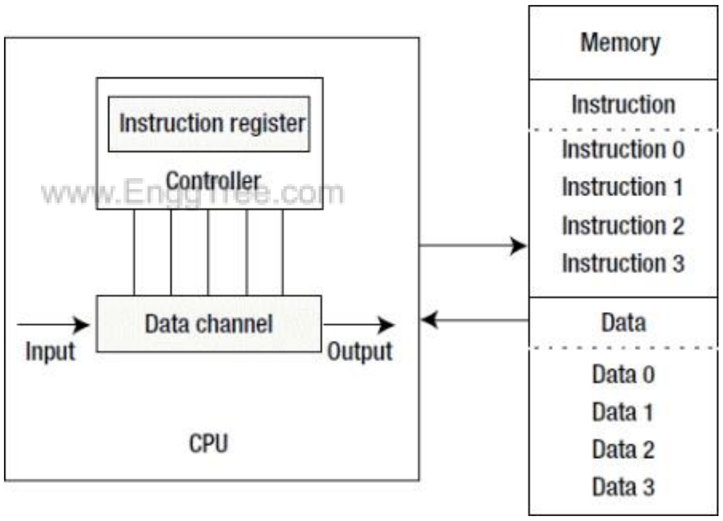


Figure 1.4: Von Neumann architecture

In the Von Neumann architecture, an instruction and data share the same bus. In this architecture, the transmission of information becomes the bottleneck of computer performance and affects the speed of data processing; so, it is often called the *Von Neumann bottleneck*. In reality, cache and branch-prediction technology can effectively solve this issue.

1.3.2 Harvard Architecture

The Harvard architecture was first named after the Harvard Mark I computer. Compared with the Von Neumann architecture, a Harvard architecture processor has two outstanding features. First, instructions and data are stored in two separate memory modules; instructions and data do not coexist in the same module. Second, two independent buses are used as dedicated communication paths between the CPU and memory; there is no connection between the two buses. The Harvard architecture is shown in Figure 1-5.

To efficiently perform memory reads/writes, the processor is not directly connected to the main memory, but to the cache. Commonly, the only difference between the Harvard architecture and the Von Neumann architecture is single or dual L1 cache. In the Harvard architecture, the L1 cache is often divided into an instruction cache (I cache) and a data cache (D cache), but the Von-Neumann architecture has a single cache.

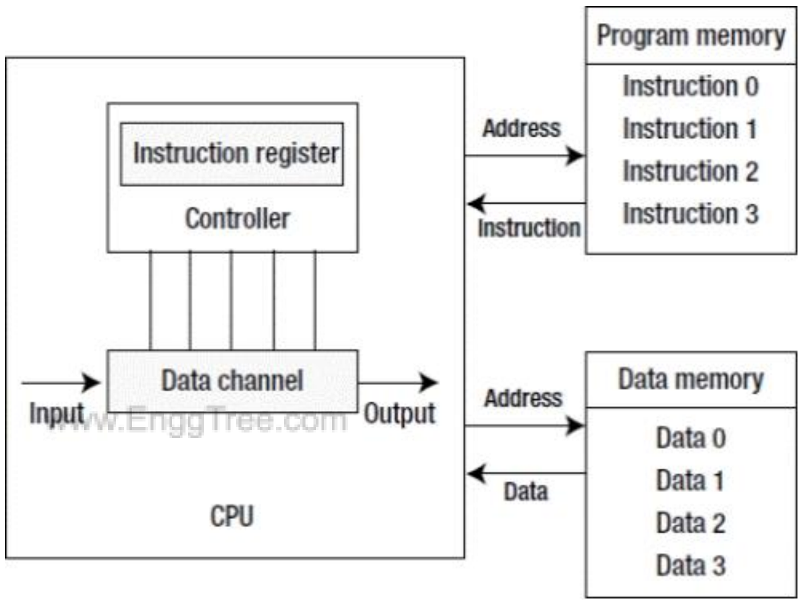


Figure 1.5: Harvard architecture

Because the Harvard architecture has separate program memory and data memory, it can provide greater data-memory bandwidth, making it the ideal choice for digital signal processing. Most systems designed for digital signal processing (DSP) adopt the Harvard architecture. The Von Neumann architecture features simple hardware design and flexible program and data storage and is usually the one chosen for general-purpose and most embedded systems.

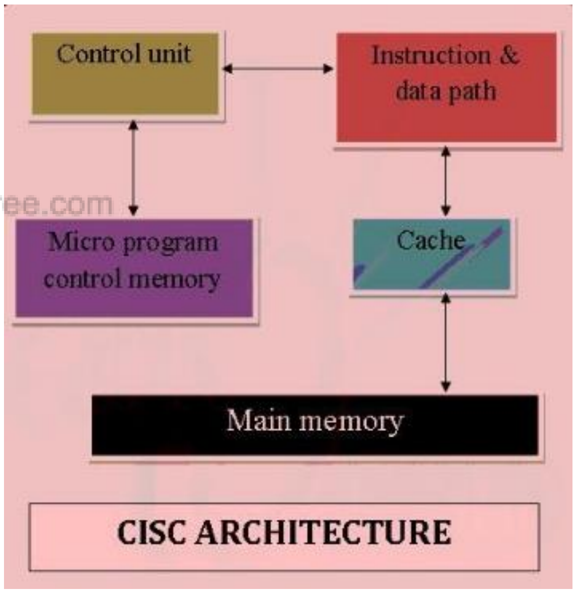
1.4. Microprocessor Architecture for Embedded Systems

A microprocessor is the CPU of the computer fabricated on a single chip. The microprocessor is the core in embedded systems. By installing a microprocessor into a special circuit board and adding the necessary peripheral circuits and expansion circuits, a practical embedded system can be created. The microprocessor architecture determines the instructions, supporting peripheral circuits, and expansion circuits. There are wide ranges of microprocessors: 8-bit, 16-bit, 32-bit and 64-bit, with clock performance from MHz to GHz, and ranging from a few pins to thousands of pins.

In general, there are two types of embedded microprocessor architecture: reduced instruction set computer (RISC) and complex instruction set computer (CISC). The RISC Nprocessor uses a small, limited, simple instruction set. Each instruction uses a standard word length and has a short execution time, which facilitates the optimization of the instruction pipeline. To compensate for the command functions, the CPU is often equipped with a large number of general-purpose registers. The CISC processor features a powerful instruction set and different instruction lengths, which facilitates the pipelined execution of instructions.

Currently, microprocessors used in most embedded systems have five architectures: RISC, CISC, MIPS, PowerPC, and SuperH. The details follow.

1.4.1 CISC Architecture



The CISC approach attempts to minimize the number of instructions per program, sacrificing the number of cycles per instruction. Computers based on the CISC architecture are designed to decrease the memory cost (figure 1.6).

Figure 1.6 CISC Architecture

Because, the large programs need more storage, thus increasing the memory cost and large memory becomes more expensive. To solve these problems, the number of instructions per program can be reduced by embedding the number of operations in a single instruction, thereby making the instructions more complex.

Characteristics of CISC processor

- MUL loads two values from the memory into separate registers in CISC.
- CISC uses minimum possible instructions by implementing hardware and executes operations.
- Instruction-decoding logic will be Complex.
- One instruction is required to support multiple addressing modes.
- Less chip space is enough for general purpose registers for the instructions that are operated directly on memory.
- Various CISC designs are set up two special registers for the stack pointer, handling interrupts, etc.
- MUL is referred to as a "complex instruction" and requires the programmer for storing functions.

Note: Instruction Set Architecture is a medium to permit communication between the programmer and the hardware. Data execution part, copying of data, deleting or editing is the user commands used in the microprocessor and with this microprocessor the Instruction set architecture is operated.

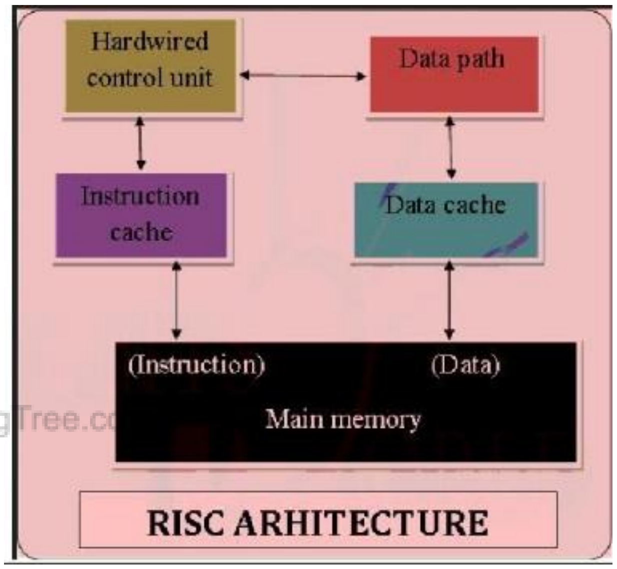
Examples of CISC PROCESSORS

- **IBM 370/168** – It was introduced in the year 1970. CISC design is a 32 bit processor and four 64-bit floating point registers.
- **VAX 11/780** – CISC design is a 32-bit processor and it supports many numbers of addressing modes and machine instructions which is from Digital Equipment Corporation.
- **Intel 80486** – It was launched in the year 1989 and it is a CISC processor, which has instructions varying lengths from 1 to 11 and it will have 235 instructions.

1.4.2 RISC Architecture

RISC (Reduced Instruction Set Computer) processors take simple instructions and are executed within a clock cycle. The first RISC projects came from IBM, Stanford, and UC-Berkeley in the late 70s and early 80s. The IBM 801, Stanford MIPS, and Berkeley RISC 1 and 2 were all designed with a similar philosophy which has become known as RISC. Certain design features have been characteristic of most RISC processors:

- *one cycle execution time*: RISC processors have a CPI (clock per instruction) of one cycle. This is due to the optimization of each instruction on the CPU and a technique called pipelining.
- *pipelining*: A techique that allows for simultaneous execution of parts, or stages, of instructions to more efficiently process instructions;
- *large number of registers*: the RISC design philosophy generally incorporates a larger number of registers to prevent in large amounts of interactions with memory



RISC is used in portable devices due to its power efficiency. For Example, Apple iPod and Nintendo DS. RISC is a type of microprocessor architecture that uses highly-optimized set of instructions. RISC does the opposite, reducing the cycles per instruction at the cost of the number of instructions per program Pipelining is one of the unique feature of RISC. It is performed by overlapping the execution of several instructions in a pipeline fashion. It has a high performance advantage over CISC.

Figure 1.7: RISC Architecture

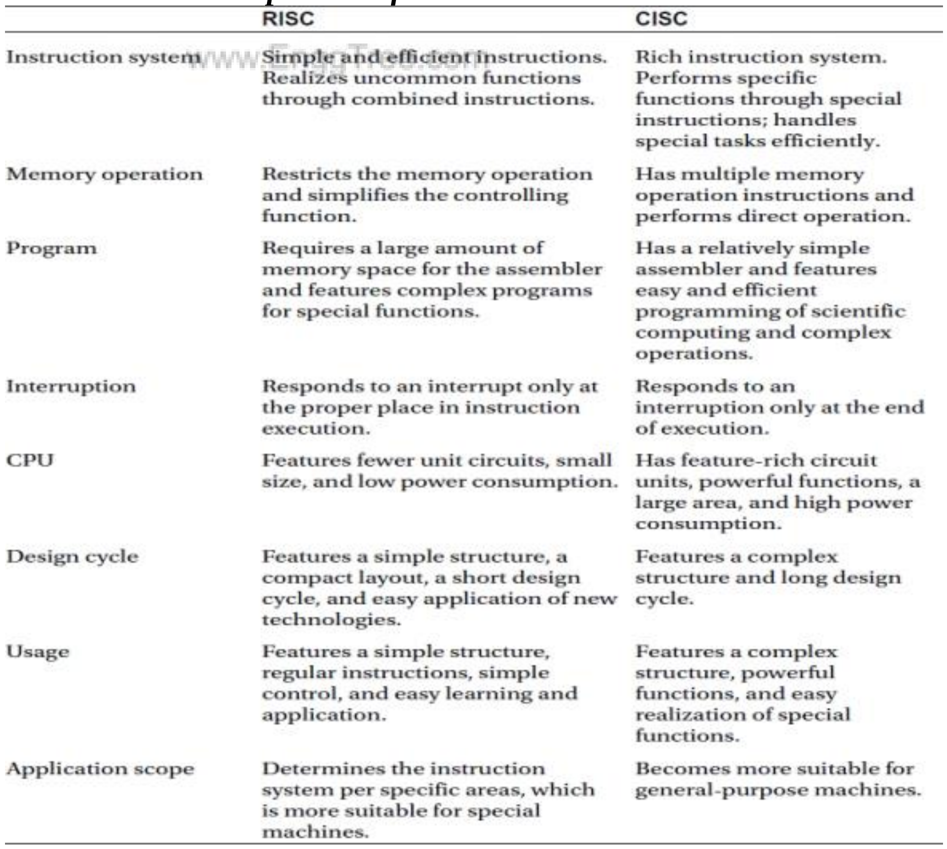
RISC Architecture Characteristics

- Simple Instructions are used in RISC architecture.
- RISC helps and supports few simple data types and synthesize complex data types.
- RISC utilizes simple addressing modes and fixed length instructions for pipelining.
- RISC permits any register to use in any context.

- One Cycle Execution Time
- The amount of work that a computer can perform is reduced by separating “LOAD” and “STORE” instructions.
- RISC contains Large Number of Registers in order to prevent various number of interactions with memory.
- In RISC, Pipelining is easy as the execution of all instructions will be done in a uniform interval of time i.e. one click.
- In RISC, more RAM is required to store assembly level instructions.
- Reduced instructions need a less number of transistors in RISC.
- RISC uses Harvard memory model means it is Harvard Architecture.
- A compiler is used to perform the conversion operation means to convert a high-level language statement into the code of its form.

A comparison of RISC and CISC is given in Table 1-2.

Table 1-2. Comparison of RISC and CISC



	RISC	CISC
Instruction system	Simple and efficient instructions. Realizes uncommon functions through combined instructions.	Rich instruction system. Performs specific functions through special instructions; handles special tasks efficiently.
Memory operation	Restricts the memory operation and simplifies the controlling function.	Has multiple memory operation instructions and performs direct operation.
Program	Requires a large amount of memory space for the assembler and features complex programs for special functions.	Has a relatively simple assembler and features easy and efficient programming of scientific computing and complex operations.
Interruption	Responds to an interrupt only at the proper place in instruction execution.	Responds to an interruption only at the end of execution.
CPU	Features fewer unit circuits, small size, and low power consumption.	Has feature-rich circuit units, powerful functions, a large area, and high power consumption.
Design cycle	Features a simple structure, a compact layout, a short design cycle, and easy application of new technologies.	Features a complex structure and long design cycle.
Usage	Features a simple structure, regular instructions, simple control, and easy learning and application.	Features a complex structure, powerful functions, and easy realization of special functions.
Application scope	Determines the instruction system per specific areas, which is more suitable for special machines.	Becomes more suitable for general-purpose machines.

RISC and CISC have distinct characteristics and advantages, but the boundaries between RISC and CISC begin to blur in the microprocessor sector. Many traditional CISCs absorb RISC advantages and use a RISC-like design. Intel x86 processors are typical of them. They are considered as CISC architecture. These processors translate x86 instructions into RISC-like instructions through a decoder and comply with the RISC design and operation to obtain the benefits of RISC architecture and improve internal operation efficiency.

1.5 System on Chip (SoC) Processor

With the development of integrated circuit design and manufacturing technology, integrated circuit design has gone from transistor integration, to logic-gate integration, to the current IP integration or system on chip (SoC). The SoC design technology integrates popular circuit modules on a single chip. SoC usually contains a large number of peripheral function modules such as microprocessor/microcontroller, memory, USB controller, universal asynchronous receiver/transmitter (UART) controller, A/D and D/A conversion, I2C, and Serial Peripheral Interface (SPI). Figure 1-8 is an example structure of SoC-based hardware for embedded systems.

A System on Chip or an SoC is an integrated circuit that incorporates a majority of components present on a computer. As the name suggests, it is an entire system fabricated on a silicon chip. SoC also includes software and an interconnection structure for integration. The hardware-software integration approach makes the SoC smaller in size, allows for less power consumption, and more reliable than a standard multi-chip system.

1.5.1 Components of an SoC

SoCs can be identified as the following types: built around a microcontroller, build around a microprocessor, built for specific applications, and programmable SoCs (PSoC). The integral parts of an SoC include a processor, primary and secondary memory storage and input/output ports. The other vital components include a graphics processor unit (GPU), a WiFi module, Digital Signal Processor (DSP), and various peripherals such as USB, Ethernet, SPI (Serial Peripheral Interface), ADC, DAC, and even FPGAs. Usually, it has multiple cores. Depending on various deciding factors and preferences, the core can be a microcontroller, microprocessor, DSP, or even an ASIP (Application Specific Instruction- set Processor). ASIPs have instruction sets based on a particular application. Usually, SoCs use ARM architecture, which is a family of RISC (Reduced Instruction Set Computing), which requires less digital design, thereby making it compatible for embedded system use. The ARM architecture is much more power-efficient than processors like the 8051 because, in contrast to processors using the

CISC architecture, processors with RISC architecture require fewer transistors. This also reduces heat dissipation and the cost.

The following diagram shows an example of an SoC block diagram.

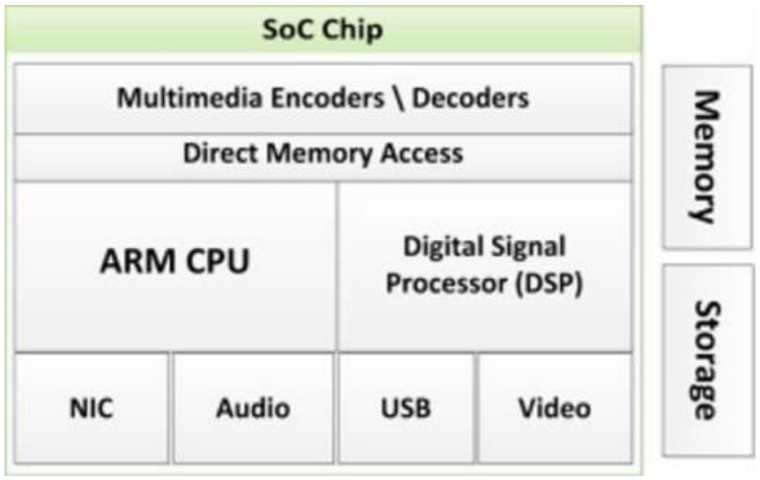


Figure 1.8: Example of an SoC block diagram.

1.5.2 Processor architecture/Models for SoC

At the heart of the SoC is its Processor. It usually has multiple processor cores. Multiple cores allow different processes to run at the same time, which increases the speed of the system as it enables your computer to perform multiple operations at the same time. The operating system sees the multiple cores as multiple CPUs, which increases performance. As multiple cores are fitted onto the same chip, there is less latency, which is because of faster communication between the cores.

1.5.2.1 Simple Sequential Processor

Sequential processors directly implement the sequential execution model. These processors process instructions sequentially from the instruction stream. The next instruction is not processed until all execution for the current instruction is complete and its results have been committed. The semantics of the instruction determines the sequence of actions that must be performed to produce the specified result. These actions include

1. fetching the instruction into the instruction register (IF),
2. decoding the opcode of the instruction (ID),
3. generating the address in memory of any data item residing there (AG),

4. fetching data operands into executable registers (DF),

5. executing the specified operation (EX), and

6. writing back the result to the register file (WB).

A simple sequential processor model is shown in Figure 1.9. During execution, a sequential processor executes one or more operations per clock cycle from the instruction stream. An instruction is a container that represents the smallest execution packet managed explicitly by the processor. One or more operations are contained within an instruction. The distinction between instructions and operations is crucial to distinguish between processor behaviors. Scalar and superscalar processors consume one or more instructions per cycle, where each instruction contains a single operation. Although conceptually simple, executing each instruction sequentially has significant performance drawbacks: A considerable amount of time is spent on overhead and not on actual execution. Thus, the simplicity of directly implementing the sequential execution model has significant performance costs.

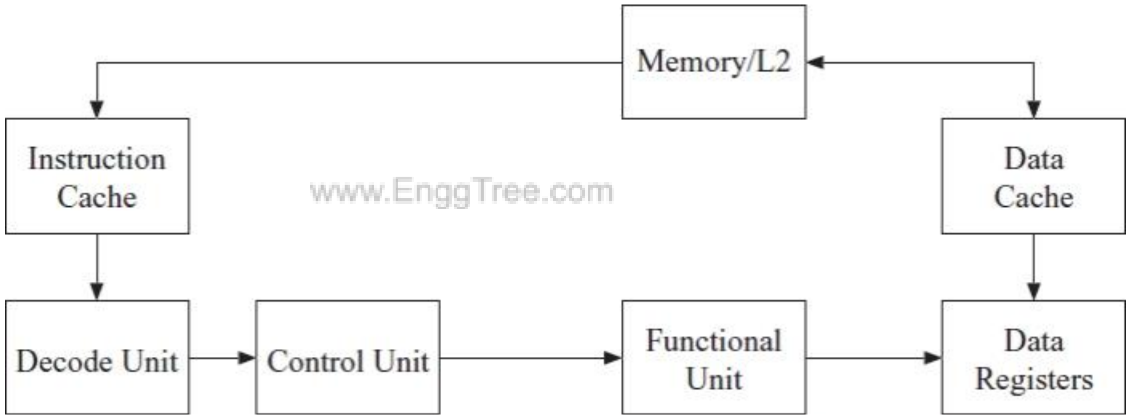


Figure1.9: Sequential Processor Model

1.5.2.2 Pipelined Processor

Pipelining is a straightforward approach to exploiting parallelism that is based on concurrently performing different phases (instruction fetch, decode, execution, etc.) of processing an instruction. Pipelining assumes that these phases are independent between different operations and can be overlapped — when this condition does not hold, the processor stalls the downstream phases to enforce the dependency. Thus, multiple operations can be processed simultaneously with each operation at a different phase of its processing. Figure 1.10 illustrates the instruction timing in a pipelined processor, assuming that the instructions are independent.

For a simple pipelined machine, there is only one operation in each phase at any given time; thus, one operation is being fetched (IF); one operation is being decoded (ID); one operation is generating an address (AG); one operation is accessing operands (DF); one operation is in execution (EX); and one operation is storing results (WB). Figure 1.10 illustrates the general form of a pipelined processor.

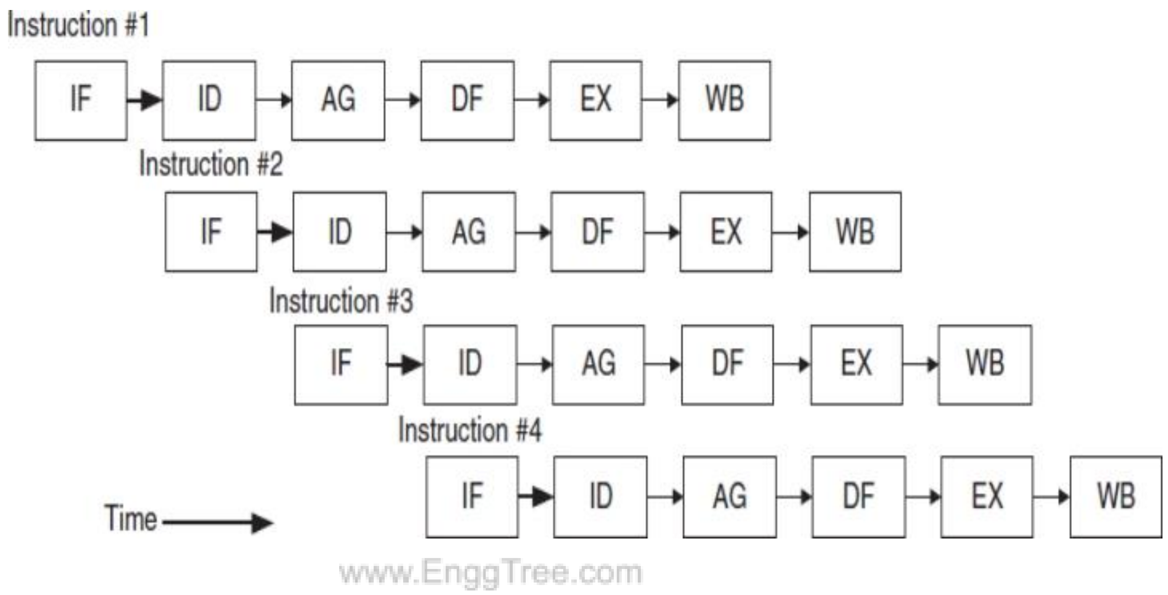


Figure 1.10: Instruction Execution in a Pipelined Processor

The most rigid form of a pipeline, sometimes called the static pipeline, requires the processor to go through all stages or phases of the pipeline whether required by a particular instruction or not. A dynamic pipeline allows the bypassing of one or more pipeline stages, depending on the requirements of the instruction. The more complex dynamic pipelines allow instructions to complete out of (sequential) order, or even to initiate out of order. The out - of - order processors must ensure that the sequential consistency of the program is preserved.

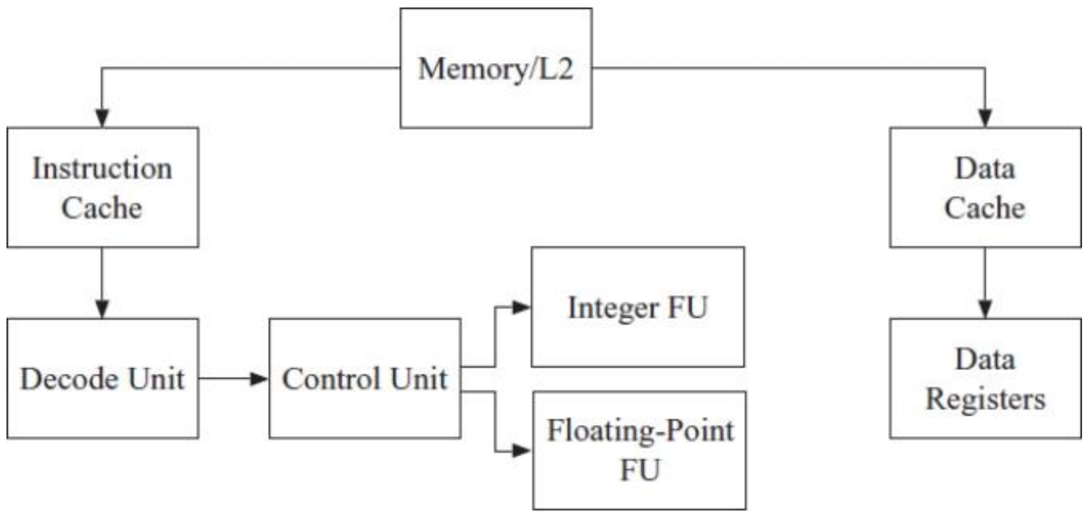


Figure 1.11 : Pipelined processor model.

Two architectures that exploit ILP (Instruction level parallelism) are *superscalar* and *VLIW* processors. They use different techniques to achieve execution rates greater than one operation per cycle. A superscalar processor dynamically examines the instruction stream to determine which operations are independent and can be executed. A VLIW processor relies on the compiler to analyze the available operations (OP) and to schedule independent operations into wide instruction words, which then execute these operations in parallel with no further analysis.

1.5.2.3 Superscalar Processors

Dynamic pipelined processors remain limited to executing a single operation per cycle by virtue of their scalar nature. This limitation can be avoided with the addition of multiple functional units and a dynamic scheduler to process more than one instruction per cycle (Figure 1.12). These superscalar processors can achieve execution rates of several instructions per cycle (usually limited to two, but more is possible depending on the application). The most significant advantage of a superscalar processor is that processing multiple instructions per cycle is done transparently to the user, and that it can provide binary code compatibility while achieving better performance.

Compared to a dynamic pipelined processor, a superscalar processor adds a scheduling instruction window that analyses multiple instructions from the instruction stream in each cycle. Although processed in parallel, these instructions are treated in the same manner as in a pipelined processor. Before an instruction is issued for execution, dependencies between the instruction and its prior instructions must be checked by hardware.

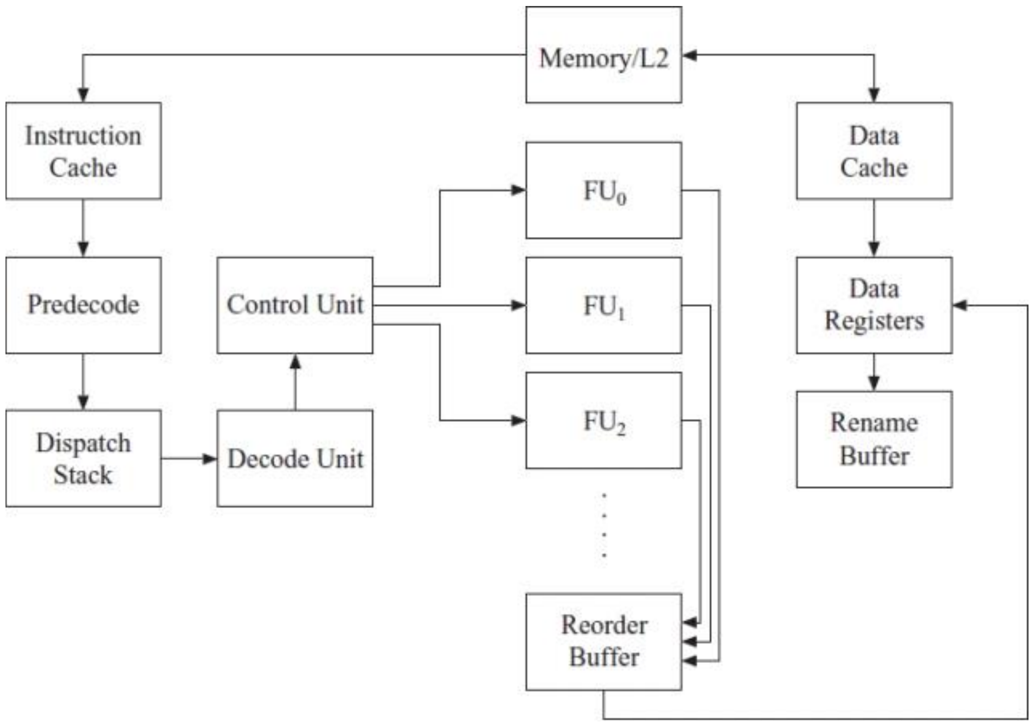


Figure 1.12 Superscalar processor model.

Because of the complexity of the dynamic scheduling logic, high – performance superscalar processors are limited to processing four to six instructions per cycle. Although superscalar processors can exploit ILP from the dynamic instruction stream, exploiting higher degrees of parallelism requires other approaches.

1.5.2.4 VLIW Processors

In contrast to dynamic analyses in hardware to determine which operations can be executed in parallel, VLIW processors (Figure 1.13) rely on static analyses in the compiler. VLIW processors are thus less complex than superscalar processors and have the potential for higher performance. A VLIW processor executes operations from statically scheduled instructions that contain multiple independent operations. Because the control complexity of a VLIW processor is not significantly greater than that of a scalar processor, the improved performance comes without the complexity penalties. VLIW processors rely on the static analyses performed by the compiler and are unable to take advantage of any dynamic execution characteristics. For applications that can be scheduled statically to use the processor resources effectively, a simple VLIW implementation results in high performance. Unfortunately, not all applications can be effectively scheduled statically. In many applications, execution does not proceed exactly along the path defined by the code scheduler in the compiler.

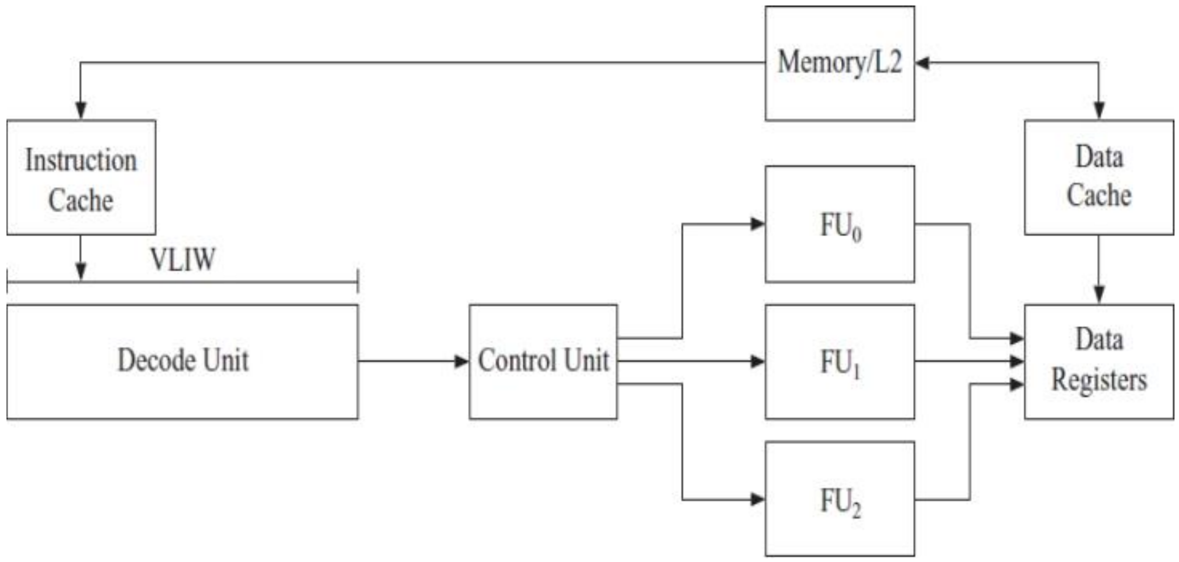


Figure 1.13 VLIW processor model.

Two classes of execution variations can arise and affect the scheduled execution behavior:

1. delayed results from operations whose latency differs from the assumedlatency scheduled by the compiler and
2. interruptions from exceptions or interrupts, which change the executionpath to a completely different and unanticipated code schedule.

Although stalling the processor can control a delayed result, this solution can result in significant performance penalties. The most common execution delay is a data cache miss. Many VLIW processors avoid all situations that can result in a delay by avoiding data caches and by assuming worst - case latencies for operations. However, when there is insufficient parallelism to hide the exposed worst - case operation latency, the instruction schedule has many incompletely filled or empty instructions, resulting in poor performance.

1.5.3 Digital Signal Processor (DSP)

Digital Signal Processor (DSP) is a chip optimized for operations for digital signal processing. This includes operations for sensors, actuators, data processing, and data analysis. It can be used for image decoding. The use of DSP saves CPU cycles for other processing tasks, which increases performance. Dedicated DSPs are more power-efficient, which makes them befitting for use in SoCs. The instruction set used for DSP cores is SIMD (Single Instruction, Multiple Data) and VLIW (Very Long Instruction Word). The use of this architecture allows for parallel processing of instructions and superscalar execution. DSPs are used to perform operations like Fast Fourier Transform, convolution, multiply-accumulate.

1.5.4 Memories on SoC

SoCs have memories based on the application. The memories are semiconductor memory blocks for computation purposes. Semiconductor memory usually refers to Metal Oxide Semiconductor memory cells, which are fabricated on a single silicon chip. The types of memories are:

- Volatile memories: Memories that lose data after power off. In other words, they need a constant power source to retain information. Volatile memories are faster and cheaper, which is why they are chosen frequently.

RAM is a type of volatile memory. The most common RAM used are SRAM (Static RAM) and DRAM (Dynamic RAM). SRAM is made of memory cells which consist of either 1,3 or 6 transistors (MOSFETs). In contrast, DRAM has only one MOSFET and a capacitor which is charged and discharged according to the state of the FET. However, DRAM is prone to capacitor leakage currents. One significant advantage of DRAM is that its cheaper than SRAM. If an SoC has a cache hierarchy, SRAM is used for cache and DRAM is used for the main memory. This is because cache requires a faster type of memory as compared to the main memory.

There are RAM types designed for non-volatile function as well. These are FRAM (Ferroelectric RAM), MRAM (Magneto-resistive random-access memory), which stores data in magnetic states, PRAM (Parameter Random Access Memory), which is used in Macintosh computers to store system settings including the display and time-zone settings. Other than these, there is RRAM (Resistive Random Access Memory), which has a component called memristor. A memristor is a resistor whose voltage varies as per the applied voltage.

- Non-volatile memories: Memories that retain information even in the absence of a power source. ROM (Read Only Memory) is a kind of non-volatile memory. Types of ROM include EPROM (Erasable Programmable Read-Only Memory), which is an array of floating-gate transistors. UVROM (Ultra-Violet Erasable Programmable Read-Only Memory), which is erased using UV light and reprogrammed with data, EEPROM (Electrically Erasable Programmable ROM) and flash.

The type of memory selected depends upon the design specifications and application.

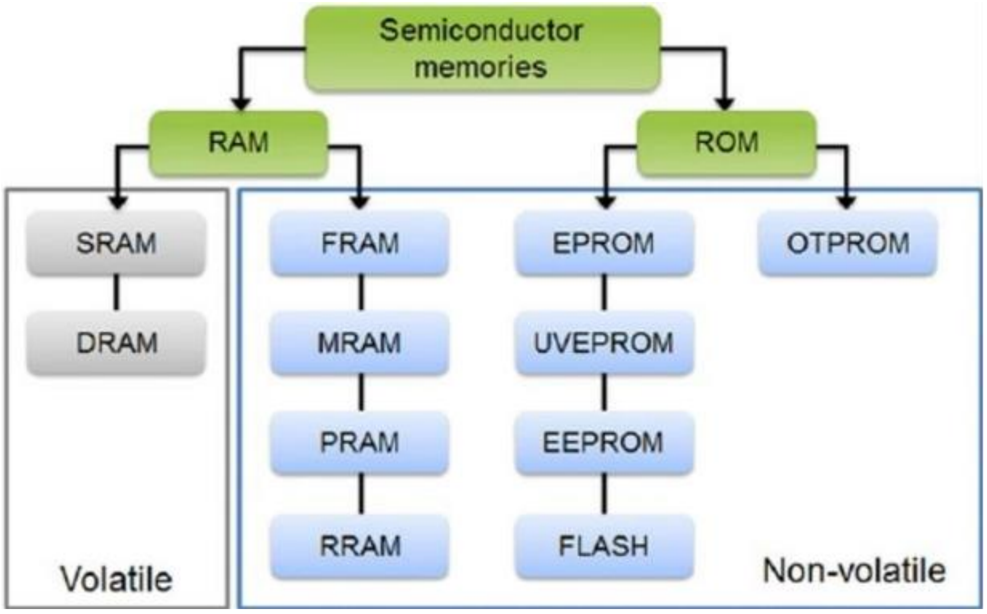


Figure 1.14: Classification of semiconductor memories used SoC.

1.5.5 SYSTEM - LEVEL INTERCONNECTION

SOC technology typically relies on the interconnection of predesigned circuit modules (known as intellectual property [IP] blocks) to form a complete system, which can be integrated onto a single chip. In this way, the design task is raised from a circuit level to a system level. Central to the system – level performance and the reliability of the finished product is the method of interconnection used. A well - designed interconnection scheme should have vigorous and efficient communication protocols, unambiguously defined as a published standard. This facilitates interoperability between IP blocks designed by different people from different organizations and encourages design reuse. It should provide efficient communication between different modules maximizing the degree of parallelism achieved. SOC interconnect methods can be classified into two main approaches:

- buses and
- network - on - chip

1.5.5.1 Bus - Based Approach

With the bus - based approach, IP blocks are designed to conform to published bus standards such as ARM ' s Advanced Microcontroller Bus Architecture (AMBA)

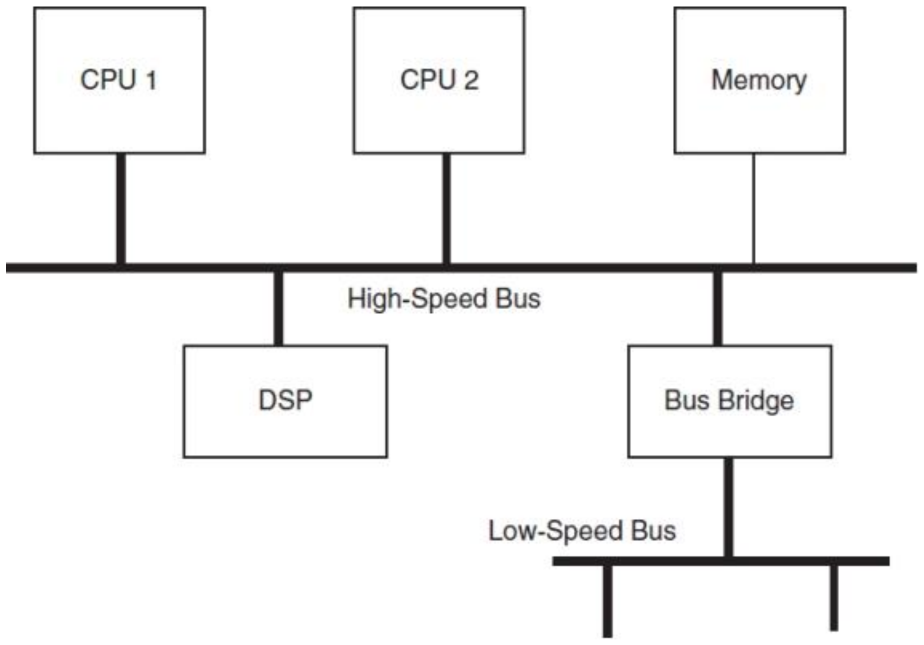


Figure 1.15: System - level interconnection: bus - based approach.

or IBM's CoreConnect. Communication between modules is achieved through the sharing of the physical connections of address, data, and control bus signals. This is a common method used for SOC system – level interconnect. Usually, two or more buses are employed in a system, organized in a hierarchical fashion. To optimize system - level performance and cost, the bus closest to the CPU has the highest bandwidth, and the bus farthest from the CPU has the lowest bandwidth.

1.5.5.2 Network - on - Chip Approach

A network - on - chip system consists of an array of switches, either dynamically switched as in a crossbar or statically switched as in a mesh. The crossbar approach uses asynchronous channels to connect synchronous modules that can operate at different clock frequencies. This approach has the advantage of higher throughput than a bus - based system while making integration of a system with multiple clock domains easier. In a simple statically switched network (Figure 1.16), each node contains processing logic forming the core, and its own routing logic. The interconnect scheme is based on a two - dimensional mesh topology. All communications between switches are conducted through data packets, routed through the router interface circuit within each node. Since the interconnections between switches have a fixed distance, interconnect - related problems such as wire delay and cross talk noise are much reduced.

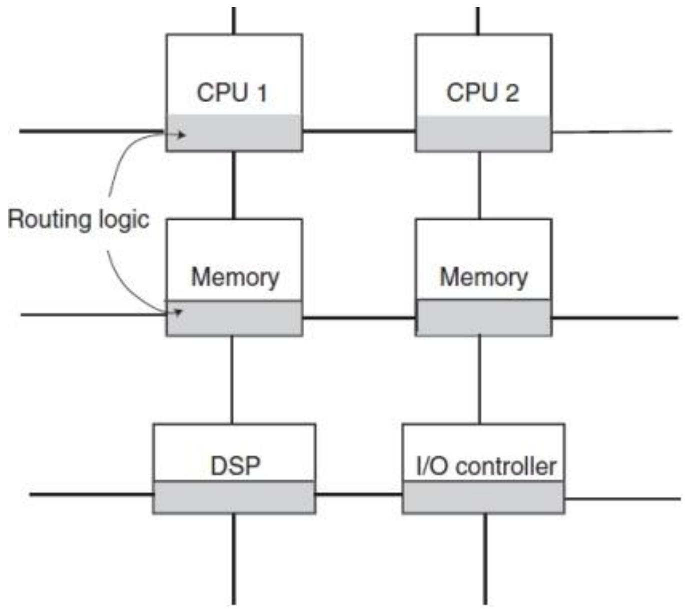


Figure 1.16: SOC interconnection: Network - on - Chip approach.

The Network-On-Chip employs system-level network techniques for on-chip traffic management. The NOC is a homogeneous, scalable switch fabric network that is used to transport multi-purpose data packets. This architecture is layered in nature with user-defined technology. The communication takes place over a three-layer communication scheme, namely Transaction, Transport and Physical.

The aim of a NOC interconnect fabric is to reduce the wire routing congestion on-chip, better timing closure, a standardized way to make changes various IPs to the SOC design. NOC architectures have proven to be more power-efficient and can match throughput requirements.

1.5.6 External interfaces

SOC interfaces defer as per the intended application. The external interfaces are commonly based on communication protocols such as WiFi, USB, Ethernet, I2C, SPI, HDMI. If required, analog interfaces may be added for interfacing with sensors and actuators.

1.5.7 Other components

Other components necessary for a fully functioning SOC are timing sources like clocks, timers, oscillators, phase lock loop systems, voltage regulators, and power management units.

1.5.6 Advantages & disadvantages of SoC

The main aim of an SoC is to minimize external components. Hence, it has the following advantages over a Single Board Computer:

- Size: The SoC is the size of a coin. Due to the rapidly decreasing size of MOS technology, SOCs can be made very small while being able to perform complex tasks. The size does not impact the features of the chip.
- Decreased power consumption: An SoC is optimized for low-power devices like cell-phones. Low power consumption results in higher battery capacity in cell-phones.
- Flexibility: SoCs are easily reprogrammable, which makes them flexible. They so allow the reuse of IPs.
- Reliability: SoCs offer high circuit security and reduced design complexity.
- Cost Efficient: Mainly due to fewer physical components and design reuse
- Faster circuit

operation SoCs pose some

disadvantages as well:

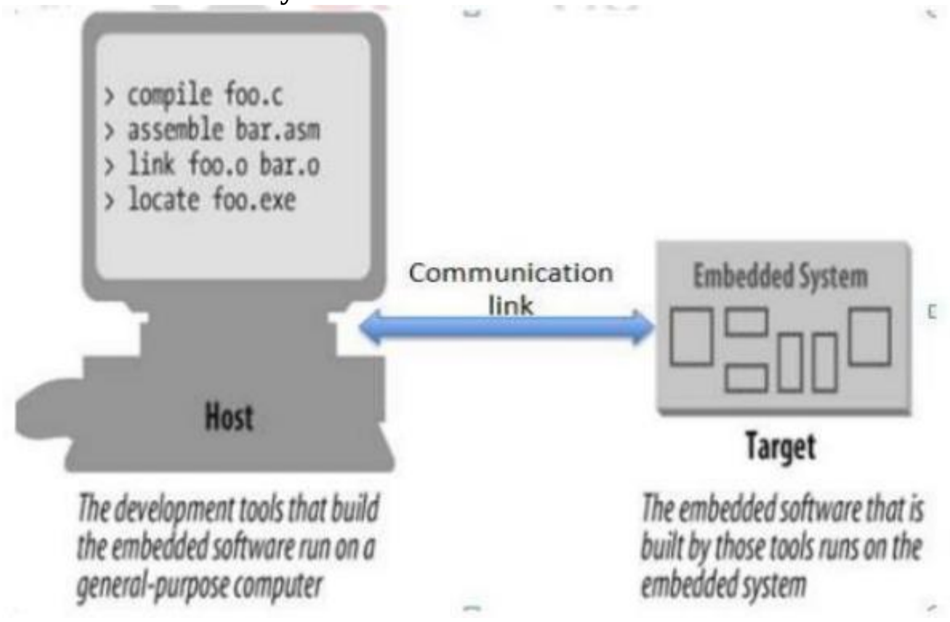
1. **Time Consuming:** The entire process from design to fabrication can take between 6 months to 1 year. Hence, the time to market demand is very high.
2. Design Verification requirements are very high and consume 70% of the total time. DV is tedious due to the increasing complexity of SoC design.
3. Availability and compatibility of IPs play a very significant role, which can add to the time to market.
4. Exponentially increasing fabrication costs.
5. For low volume products, SoC may not be the best option.

1.5.7 Applications

The most common application of SOCs today is in mobile applications, including smartphones, smartwatches, tablets. Other applications include signal speech processing, PC interfaces, data communication. SoCs are being applied to personal computers as well due to the integration of communication modules like LTE and wireless networks onto the chip.

The most popular SoCs in the market today are manufactured by Qualcomm Technologies for smartphones, smartwatches, and the upcoming 5G network compatibility. Other manufacturers include Intel Technology, Samsung Inc, Apple Inc., among many others.

1.6 Software Development process for embedded system


> compile foo.c
> assemble bar.asm
> link foo.o bar.o
> locate foo.exe
Communication link
Embedded System
Host
Target
The development tools that build the embedded software run on a general-purpose computer
The embedded software that is built by those tools runs on the embedded system

Because machine code is the only language the hardware can directly execute, all other languages need some type of mechanism to generate the corresponding machine code. This mechanism usually includes one or some combination of *preprocessing*, *translation*, and *interpretation*. Depending on the language, these mechanisms exist on the programmer's ***host*** system (typically a non-embedded development system, such as a PC), or the ***target*** system (the embedded system being developed). See Figure 1.17.

Figure 1.17: Host and target system diagram

Preprocessing is an optional step that occurs before either the translation or interpretation of source code, and whose functionality is commonly implemented by a ***preprocessor***. The preprocessor's role is to organize and restructure the source code to make translation or interpretation of this code easier. As an example, in languages like C and C++, it is a preprocessor that allows the use of named code fragments, such as *macros*, that simplify code development by allowing the use of the macro's name in the code to replace fragments of code. The preprocessor then replaces the macro name with the contents of the macro during preprocessing. The preprocessor can exist as a separate entity, or can be integrated within the translation or interpretation unit.

1.6.1 Compiler

Many languages convert source code, either directly or after having been preprocessed through use of a ***compiler***, a program that generates a particular target language such as machine code and Java byte code from the source language as depicted in Figure 1.18. A compiler typically “translates” all of the source code to some target code at one time. As is usually the case in embedded systems, compilers are located on the programmer’s host machine and generate target code for hardware platforms that differ from the platform the compiler is actually running on. These compilers are commonly referred to as ***cross-compilers***. In the case of assembly language, the compiler is simply a specialized cross-compiler referred to as an ***assembler***, and it always generates machine code. The language name plus the term “compiler, ”such as“ Java compiler and C compiler, commonly refer to other high-level language compilers.

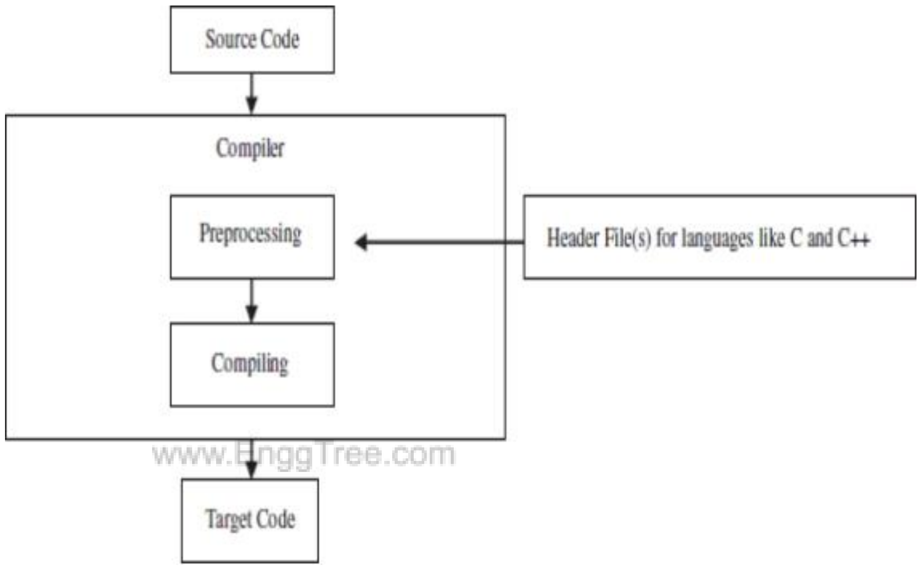


Figure 1.18 General functions of an Embedded software

High-level language compilers vary widely in terms of what is generated. Some generate machine code, while others generate other high-level code, which then requires what is produced to be run through at least one more compiler or interpreter, as discussed later in this section. Other compilers generate assembly code, which then must be run through an assembler. After all the compilation on the programmer’s host machine is completed, the remaining target code file is commonly referred to as an ***object file***, and can contain anything from machine code to Java byte code (discussed later in this section), depending on the programming language used. As shown in Figure 1.13, after linking this object file to any system libraries required, the object file, commonly referred to as an ***executable***, is then ready to be transferred to the target embedded system’s memory.

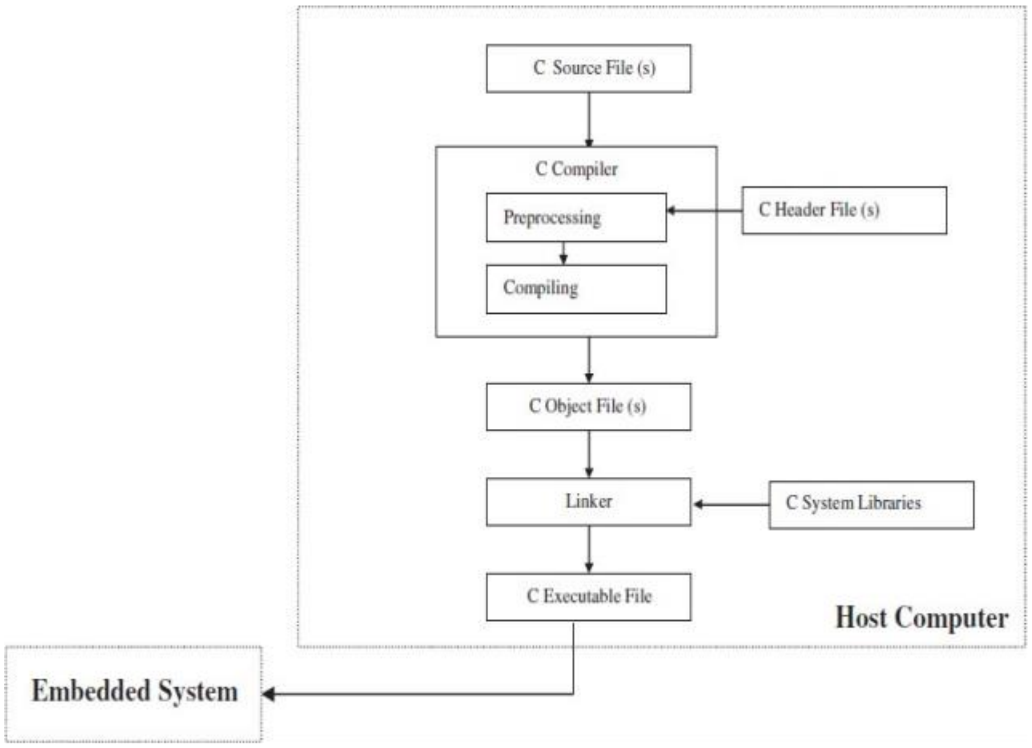


Figure 1.19: C Example compilation/linking steps and object file results

References

[1] F. Vahid and T. Givargis, “Embedded System Design: A Unified Hardware/Software Introduction”, Wiley India Pvt. Ltd., 2002.

[2] Michael J. Flynn and Wayne Luk, “Computer System Design System-on-Chip”, Wiley India Pvt. Ltd.

[3] Steve Furber, “ARM System on Chip Architecture “, 2nd Edition, 2000, Addison Wesley Professional.AAAXZX

[4] Pascricha and N. Dutt, Morgan Kaufmann, On-Chip Communication Architectures, System on Chip Interconnect, -Elsevier Publishers 2008

Exercise Questions

1. List the important considerations when selecting a processor for embedded systemdesign.
2. Categorize the different types of computing devices used to design embedded systems.
3. List the merits and de-merits of Von Neumann processor architecture.
4. Mention the key characteristics of RISC processors.
5. Identify the scenarios that creates a bottleneck for pipelined instruction execution.
6. Contrast superscalar and VLIW processor architectures with respect to compiler design.
7. Outline the reasons for using CISC architecture based processors for desktop computers.
8. Compare SoC processor and application specific integrated circuits.
9. Mention the reason for the widespread use of Dynamic RAMs for main memory in spiteof being slower than Static RAMs.
10. Distinguish scratch pads and cache memory.
11. Recall the two types of interconnect architectures used in SoC processors.
12. Give examples of commercial embedded processors with RISC architectures.

13. Illustrate the key aspects of Von-Neumann and Harvard architectures used in the designof computers.
14. Explain with suitable examples, the process of instruction execution in CISC and RISCprocessors.
15. Illustrate the basic architecture of a System on Chip processor and summarize theimportance of each functional unit.
16. Classify the types of on-chip memories used in SoC processors.
17. Demonstrate with suitable examples that a superscalar processor can improve theefficiency of instruction level of parallelism.
18. Examine the architecture of VLIW processor model and give your opinion on how itleads to lower hardware complexity compared to superscalar model.

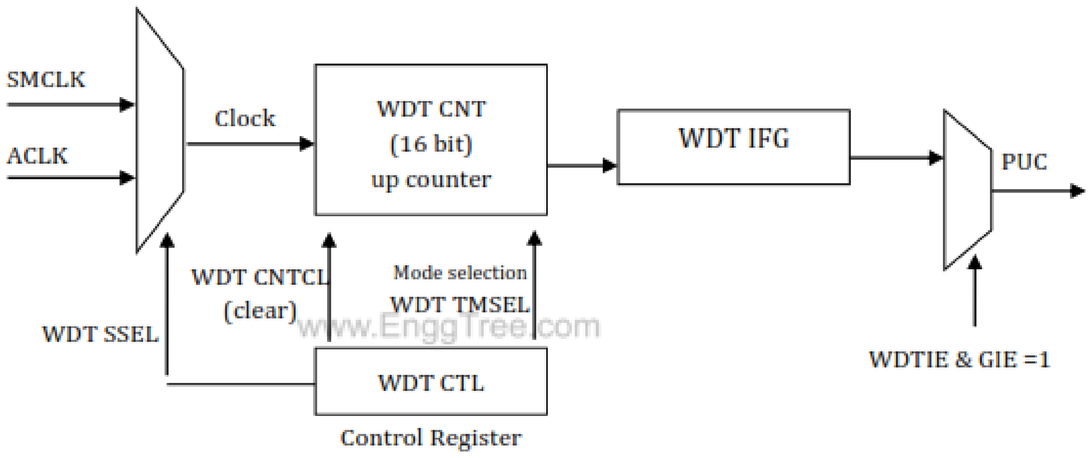
UNIT II

PERIPHERAL INTERFACING

WATCHDOGTIMERS

The main purpose of the watchdogtimer is to protect the system against failure of the software,such as the program becoming trapped in an unintended ,infiniteloop.

Watchdogcountsupand resets the MSP430 when it reaches its limit. The code must there for keep clearing the counter before the limit is reached to prevent a reset. The operation of the watchdog is controlled by the 16-bitregister
WDTCTL



The watchdog counter is a 16-bit register WDTCNT, which is not visible to the user. It is clocked from either SMCLK (default) or ACLK, according to the WDTSSEL bit. The watchdog is always active after theMSP430 has been reset.

By default the clock is SMCLK, which is in turn derived from the DCO at about1MHz.Thedefaultperiodofthewatchdogisthemaximum value of 32,768 counts, which is therefore around 32ms. We must clear, stop, or reconfigure the watchdog before this time has elapsed. If the watchdog is left running, the counter must be repeatedly cleared to prevent it counting up as far as its limit. This is done by setting the WDTCNTCL bit in WDTCTL.

Thewatchdogtimer sets theWDTIFGflaginthespecialfunctionregisterIFG1.Thisisclearedbyapower-onresetbutits valueispreservedduringaPUC.Thusaprogram canchecktthisbittofindoutwhetheraresetarose fromthewatchdog.

Applications of watchdog timer

Following are the applications of watchdog timer:
• An application in mobile phone is that display is off in case no GUI interaction takes place within a watched time interval. This will save good amout of battery power.
• An application in temperature controller is that if controller takes no action to switch off the current within preset watched time interval, the current is switched off and warning signal is raised as indication of controller failure. Failure to switch off current may burst a boiler in which water is heated.
• If Software hangs due to some bug/issue, it helps to reset the system automatically without any human interactions.

Interfacing Protocols-SPI, I^{2}C, USB, CAN, Ethernet/WiFi,Bluetooth

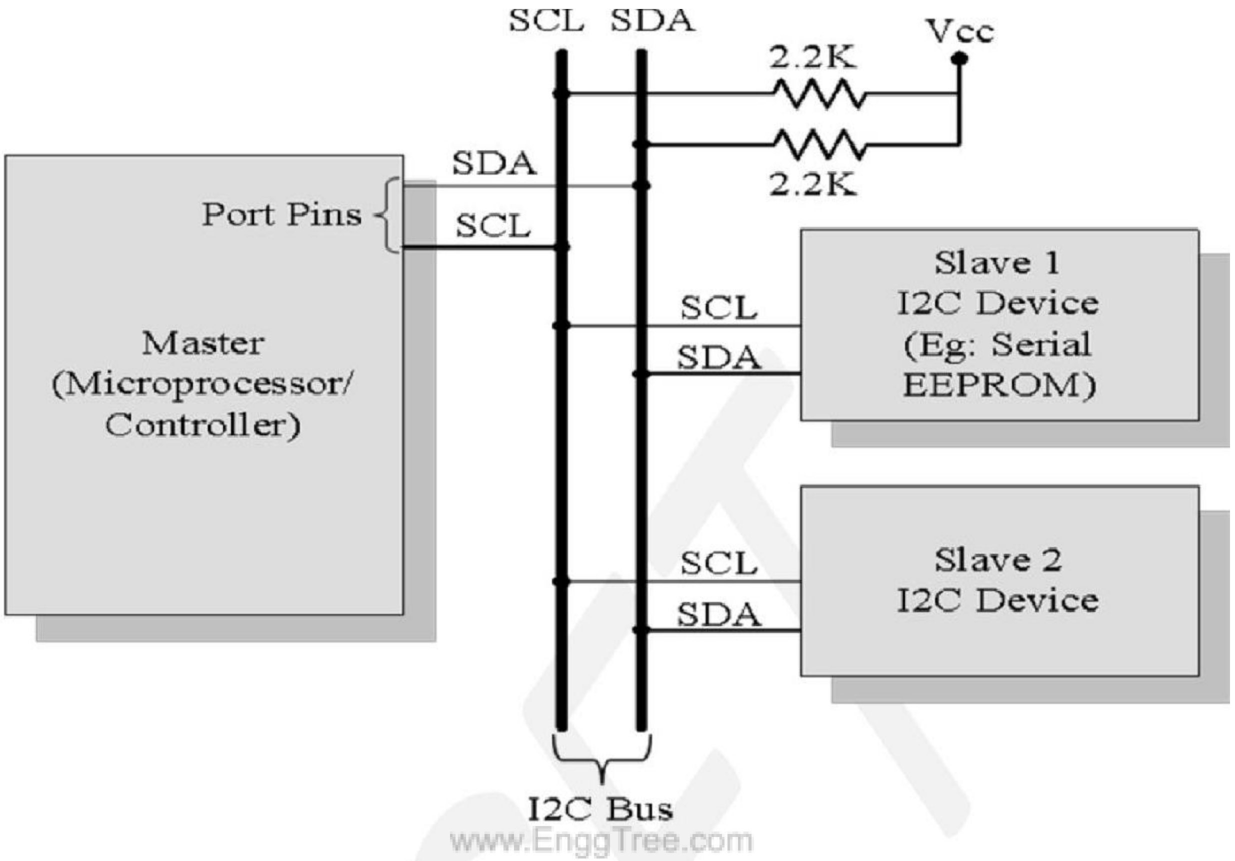
I2C (Inter Integrated Circuit) Bus:

Inter Integrated Circuit Bus (I2C - Pronounced „I square C‟) is a synchronous bi-directional half duplex (one-directional communication at a given point of time) two wire serial interface bus.The concept of I2C bus was developed by „Philips Semiconductors‟ in the early 1980‟s. 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.

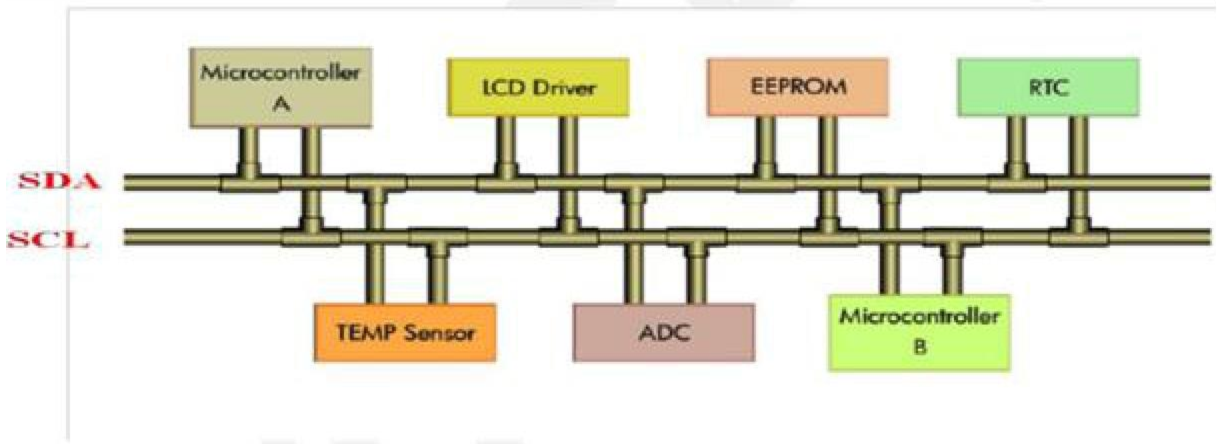
The I2C bus is comprised of two bus lines, namely; Serial Clock – SCL and Serial Data – SDA.

SCL line is responsible for generating synchronization clock pulses and SDA is 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‟ device or „Slave‟ device.

The „Master‟ device is responsible for controlling the communication by initiating/terminating data transfer, sending data and generating necessary synchronization clock pulses.



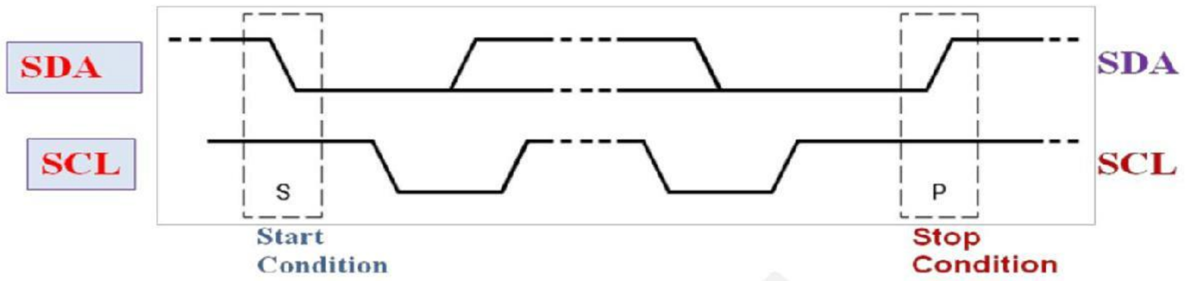
I2C Bus Interfacing



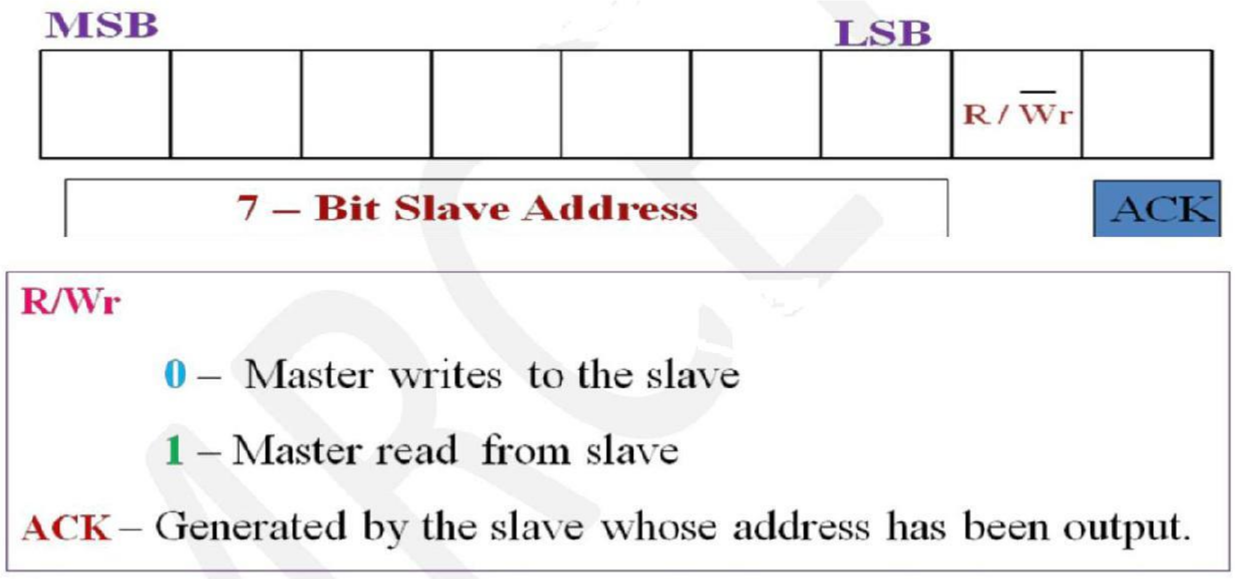
Slave devices wait for the commands from the master and respond upon receiving the commands. Master and „Slave" devices can act as either transmitter or receiver. Regardless whether a master is acting as transmitter or receiver, the synchronization clock signal is generated by the „Master" device only.I2C supports multi masters on the same bus.

The sequence of operation for communicating with an I2C slave device is:

1. Master device pulls the clock line (SCL) of the bus to „HIGH"
2. 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. Master sends the address (7 bit or 10 bit wide) of the „Slave" device to which it wants to communicate, over the SDA line.
4. Clock pulses are generated at the SCL line for synchronizing the bit reception by the slave device.
5. The MSB of the data is always transmitted first.
6. The data in the bus is valid during the „HIGH" period of the clock signal
7. In normal data transfer, the data line only changes state when the clock is low.


MSB
LSB
$R/\overline{Wr}$
7 – Bit Slave Address
ACK
R/Wr
0 – Master writes to the slave
1 – Master read from slave
ACK – Generated by the slave whose address has been output.

8. Master waits for the acknowledgement bit from the slave device whose address is sent on the bus along with the Read/Write operation command.

9. Slave devices connected to the bus compares the address received with the address assigned to them

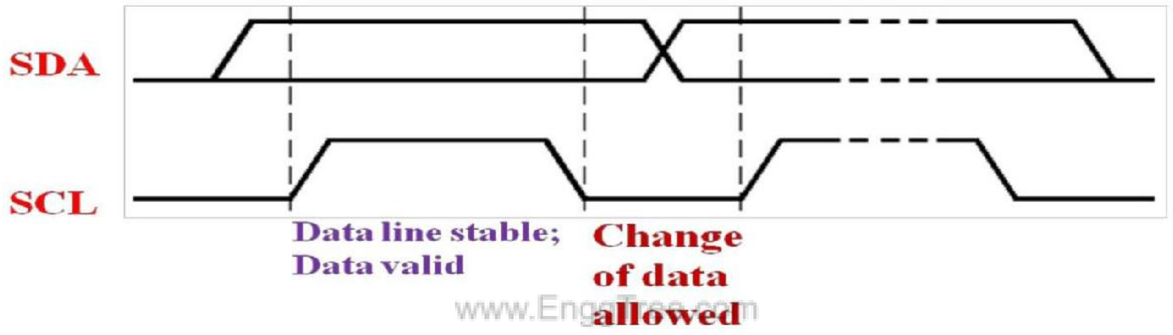
10. The Slave device with the address requested by the master device responds by sending an acknowledge bit (Bit value =1) over the SDA line

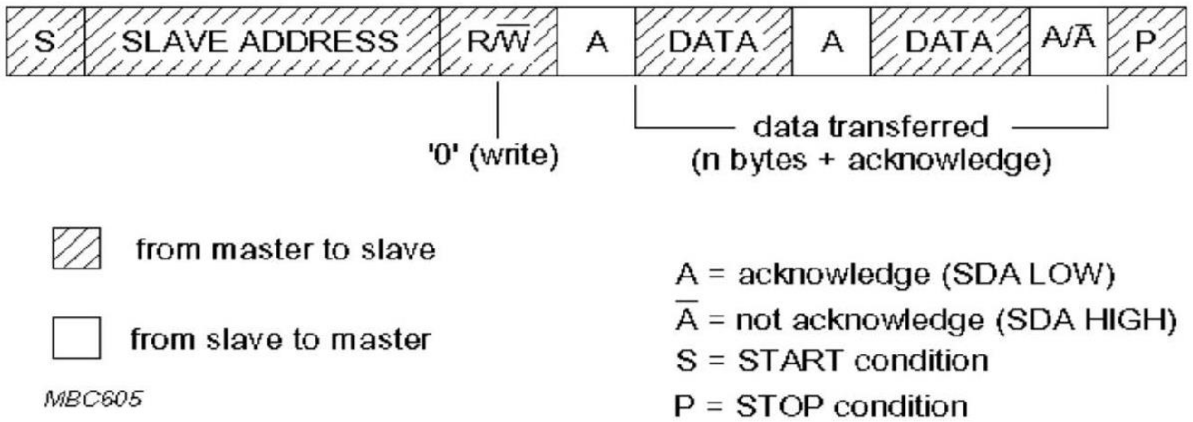
11. Upon receiving the acknowledge bit, master sends the 8bit data to the slave device over SDA line, if the requested operation is „Write to device".

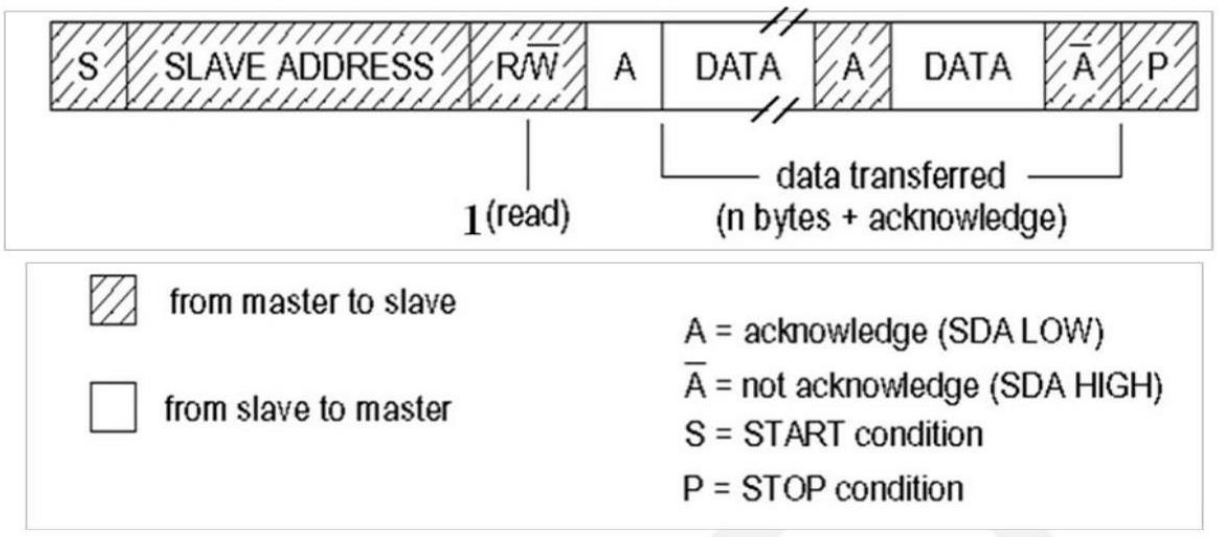
12. If the requested operation is „Read from device", the slave device sends data to the master over the SDA line.

13. Master 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

14. Master terminates the transfer by pulling the SDA line „HIGH" when the clock line SCL is at logic „HIGH" (Indicating the „STOP" condition).



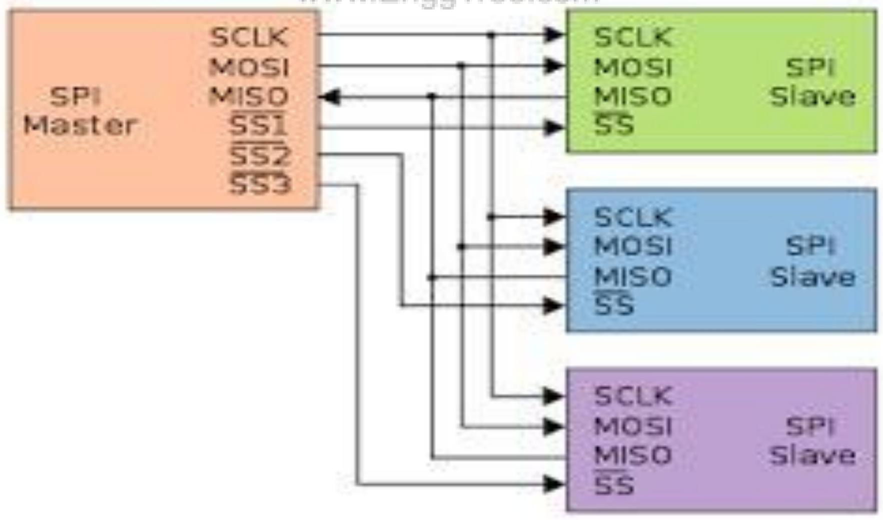




1.2 Serial Peripheral Interface (SPI) Bus:

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

- It is possible to have a system where more than one SPI device can be master, provided the condition only one master device is active at any given point of time, is satisfied.
- SPI is used to send data between Microcontrollers and small peripherals such as shift registers, sensors, and SD cards.



SPI requires four signal lines for communication. They are:

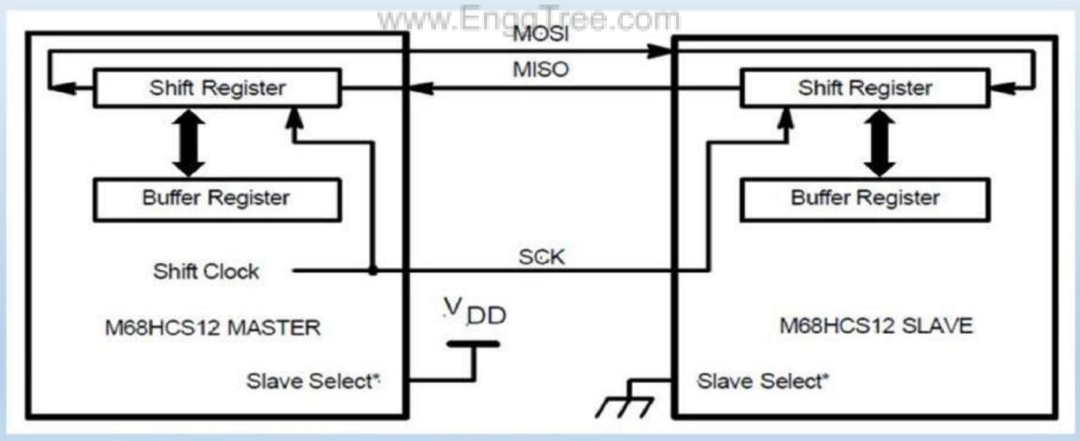
Master Out Slave In (MOSI): Signal line carrying the data from master to slave device. It isalso known as Slave Input/Slave Data In (SI/SDI)

Master In Slave Out (MISO): Signal line carrying the data from slave to master device. It isalso 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 master device is responsible for generating the clock signal.
Master device selects the required slave device by asserting the corresponding slave devices slave select signal „LOW‟.

- The data out line (MISO) of all the slave devices when not selected floats at high impedancestate
- The serial data transmission through SPI Bus is fully configurable.
- SPI devices contain certain set of registers for holding these configurations.
- The Serial Peripheral Control Register holds the various configuration parameters like master/slave selection for the device, baudrate 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.

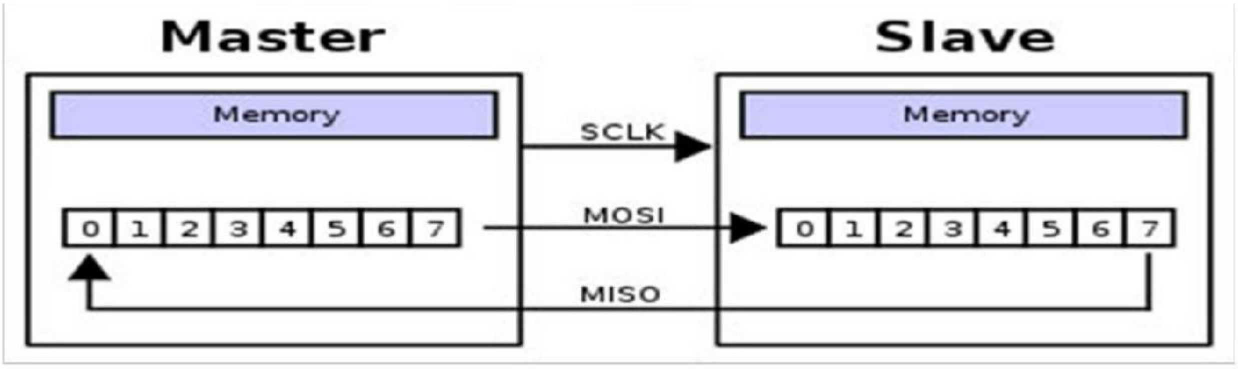
Operation of SPI protocol



Master/slave serial peripheral interface.

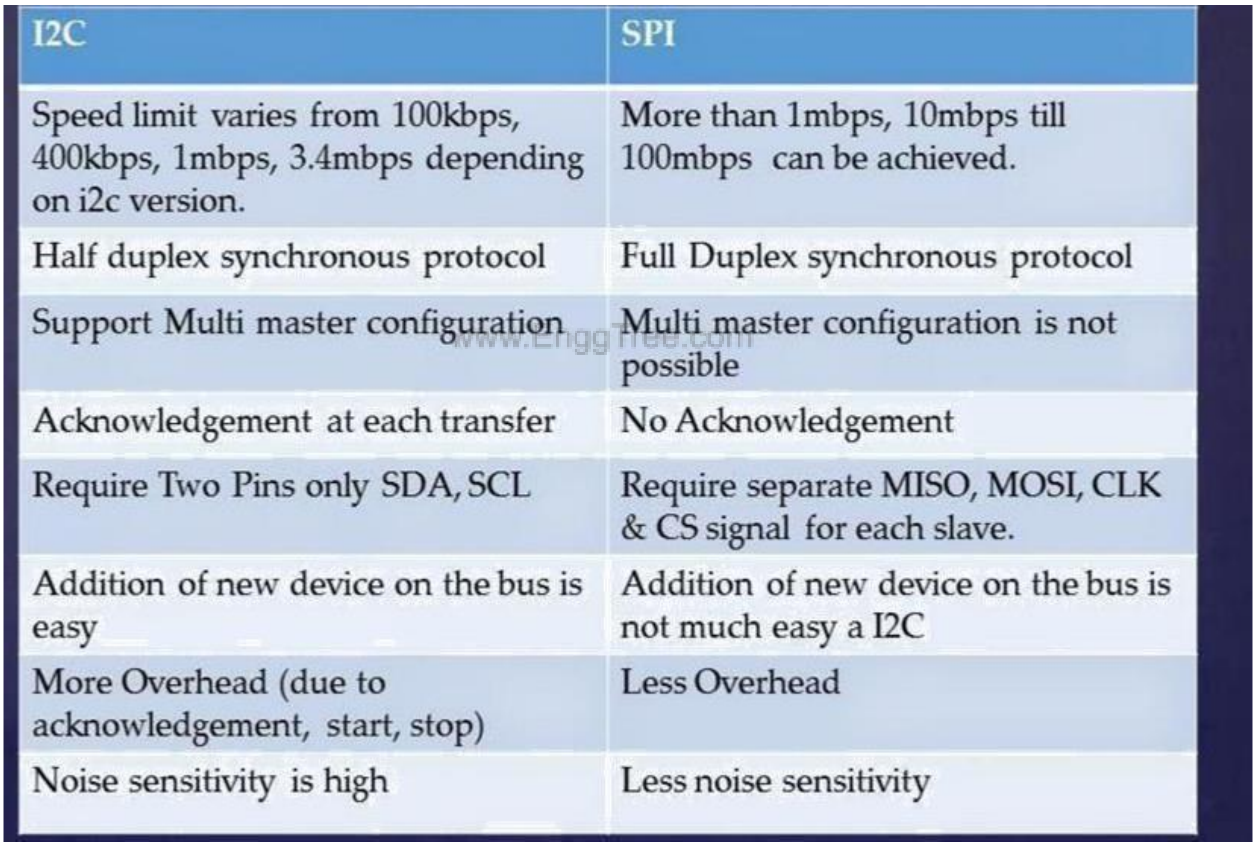
- During transmission from the master to slave, the data in the master‟s shift register is shifted out to the MOSI pin and it enters the shift register of the slave device through the MOSI pin of the slave device.

- 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



Master shifts out data to Slave, and shift in data from Slave

I2C V/S SPI:



I2C	SPI
Speed limit varies from 100kbps, 400kbps, 1mbps, 3.4mbps depending on i2c version.	More than 1mbps, 10mbps till 100mbps can be achieved.
Half duplex synchronous protocol	Full Duplex synchronous protocol
Support Multi master configuration	Multi master configuration is not possible
Acknowledgement at each transfer	No Acknowledgement
Require Two Pins only SDA, SCL	Require separate MISO, MOSI, CLK & CS signal for each slave.
Addition of new device on the bus is easy	Addition of new device on the bus is not much easy a I2C
More Overhead (due to acknowledgement, start, stop)	Less Overhead
Noise sensitivity is high	Less noise sensitivity

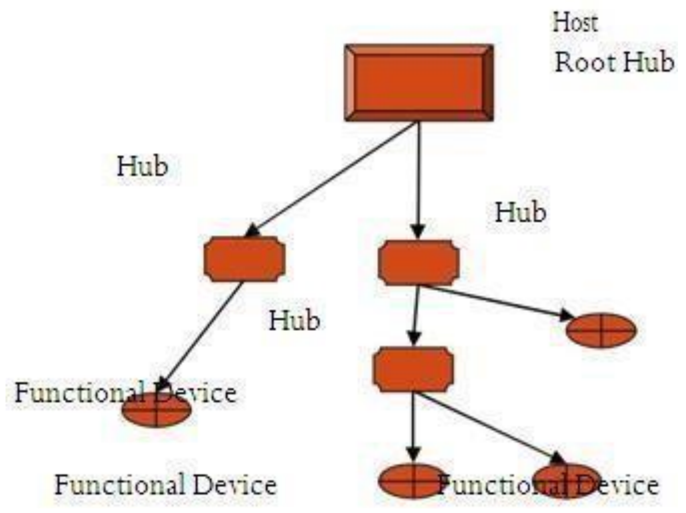
USB (UNIVERSAL SERIAL BUS):

- External Bus Standard.
- Allows connection of peripheral devices.
- Connects Devices such as keyboards, mice, scanners, printers, joysticks, audio devices, disks.

- Facilitates transfers of data at 480 (USB 2.0 only), 12 or 1.5 Mb/s (mega-bits/second).
- Developed by a Special Interest Group including Intel, Microsoft, Compact, DEC, IBM, Northern Telecom and NEC originally in 1994.
- Low-Speed: 10 – 100 kb/s
- 1.5 Mb/s signaling bit rate
- Full-Speed: 500 kb/s – 10 Mb/s 12 Mb/s signaling bit rate
- High-Speed: 400 Mb/s
- 480 Mb/s signaling bit rate
- NRZI with bit stuffing used
- SYNC field present for every packet
- There exist two pre-defined connectors in any USB system - Series "A" and Series "B" Connectors.
- Series "A" cable: Connects USB devices to a hub port.
- Series "B" cable: Connects detachable devices (hot- swappable)

Bus Topology:

- Connects computer to peripheral devices.
- Ultimately intended to replace parallel and serial ports
- Tiered Star Topology
- All devices are linked to a common point referred to as the root hub.
- Specification allows for up to 127 (2^7 -1) different devices.



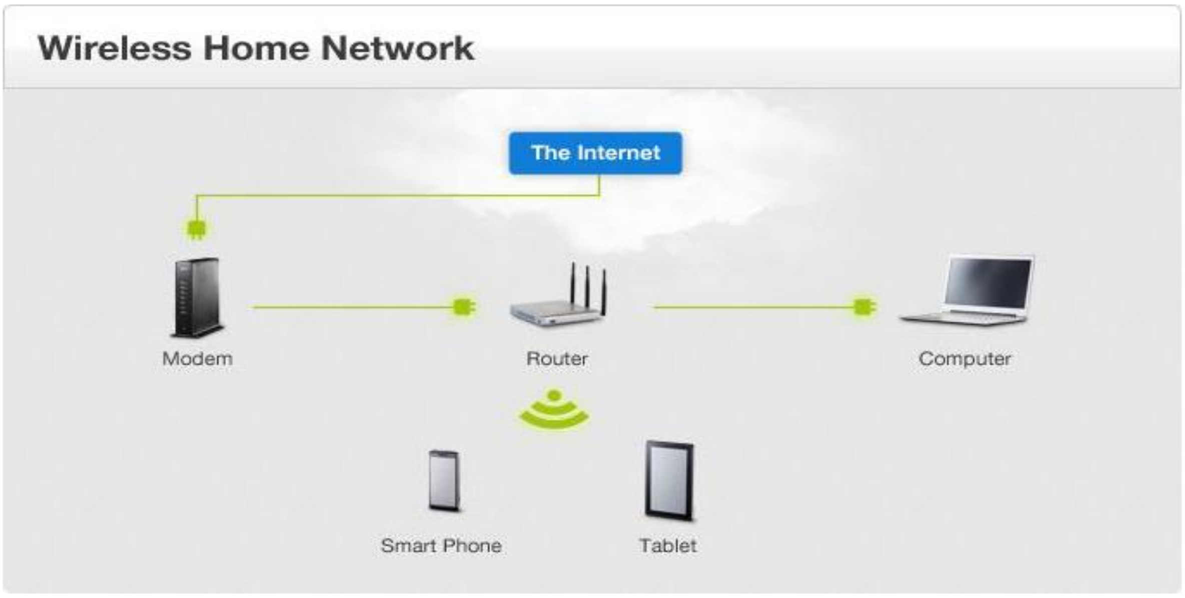
- Four wire cable serves as interconnect of system - power, ground and two differential signaling lines.
- USB is a polled bus-all transactions are initiated by host.

USB HOST: Device that controls entire system usually a PC of some form. Processes dataarriving to and from the USB port.

USB HUB: Tests for new devices and maintains status information of child devices.Serve as repeaters, boosting strength of up and downstream signals. Electrically isolates devices from one another - allowing an expanded number of devices.

Wi-Fi:

- Wi-Fi is the name of a popular wireless networking technology that uses radio waves to provide wireless high-speed Internet and network connections
- Wi-Fi follows the IEEE 802.11 standard
- Wi-Fi is intended for network communication and it supports Internet Protocol (IP) based communication
- 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.

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

BLUETOOTH:

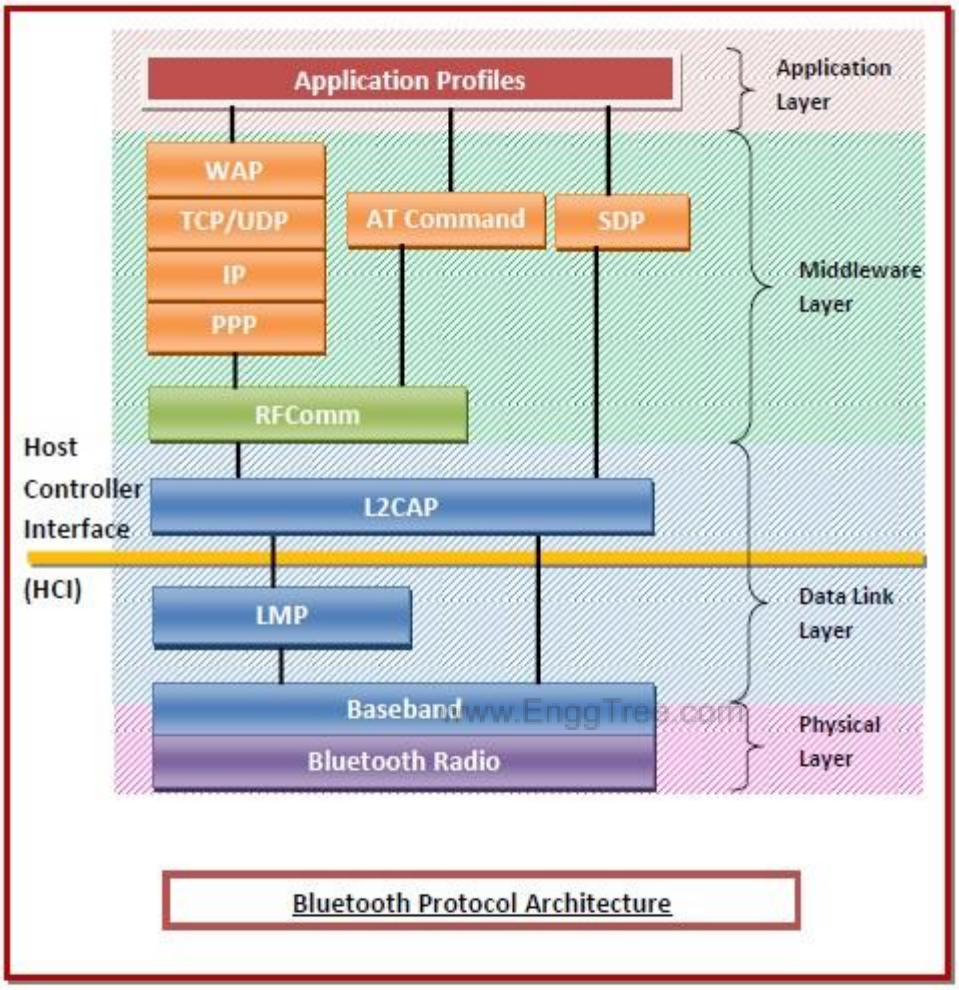
Bluetooth is a wireless technology standard for short distances (using short-wavelength UHFband from 2.4 to 2.485 GHz)for exchanging data over radio waves in the ISM and mobile devices, and building personal area networks (PANs).Invented by telecom vendor Ericsson in1994, it was originally conceived as a wireless alternative to RS- 232 data cables

Bluetooth uses a radio technology called frequency- hopping spread spectrum. Bluetooth divides transmitted data into packets, and transmits each packet on one of 79 designated Bluetooth channels. Each channel has a bandwidth of 1 MHz. It usually performs 800 hops per second, with Adaptive Frequency-Hopping (AFH) enabled

Originally, Gaussian frequency-shift keying (GFSK) modulation was the only modulation scheme available. Since the introduction of Bluetooth 2.0+EDR, π/4-DQPSK (Differential Quadrature Phase Shift Keying) and 8DPSK modulation may also be used between compatible devices. Bluetooth is a packet-based protocol with a master- slave structure. One master may communicate with up to seven slaves in a piconet. All devices share the master's clock. Packet exchange is based on the basic clock, defined by the master, which ticks at312.5 µs intervals.

A master BR/EDR Bluetooth device can communicate with a maximum of seven devices in a piconet (an ad-hoc computer network using Bluetooth technology), though not all devices reach this maximum. The devices can switch roles, by agreement, and the slave can become the master (for example, a headset initiating a connection to a phone necessarily begins as master— as initiator of the connection—but may subsequently operate as slave).

Bluetooth network technology connects mobile devices wirelessly over a short-range to form a personal area network (PAN). The Bluetooth architecture has its own independent model with a stack of protocols, instead of following the standard OSI model or TCP/IP model.
The protocols in the Bluetooth standard can be loosely grouped into the physical layer, data link layer, middleware layer, and application layer as shown in the following diagram −



Protocols in the Bluetooth Protocol Architecture

- **Physical Layer** − This includes Bluetooth radio and Baseband (also in the data link layer.
    - **Radio** − This is a physical layer equivalent protocol that lays down the physical structure and specifications for transmission of radio waves. It defines air interface, frequency bands, frequency hopping specifications, and modulation techniques.
    - **Baseband** − This protocol takes the services of radio protocol. It defines the addressing scheme, packet frame format, timing, and power control algorithms.
- **Data Link Layer** − This includes Baseband, Link Manager Protocol (LMP), and Logical Link Control and Adaptation Protocol (L2CAP).
    - **Link Manager Protocol (LMP)** − LMP establishes logical links between Bluetooth devices and maintains the links for enabling communications. The other main functions of LMP are device authentication, message encryption, and negotiation of packet sizes.

    - **Logical Link Control and Adaptation Protocol (L2CAP)** − L2CAP provides adaption between upper layer frame and baseband layer frame format. L2CAP provides support for both connection-oriented as well as connectionless services.
- **Middleware Layer** − This includes Radio Frequency Communications (RFComm) protocol, adopted protocols, SDP, and AT commands.
    - **RFComm** − It is short for Radio Frontend Component. It provides a serial interface with WAP.
    - **Adopted Protocols** − These are the protocols that are adopted from standard models. The commonly adopted protocols used in Bluetooth are Point-to-Point Protocol (PPP), Internet Protocol (IP), User Datagram Protocol (UDP), Transmission Control Protocol (TCP), and Wireless Application Protocol (WAP).
    - **Service Discovery Protocol (SDP)**− SDP takes care of service-related queries like device information so as to establish a connection between contending Bluetooth devices.
    - **AT Commands** − ATtention command set.
- **Applications Layer** − This includes the application profiles that allow the user to interact with the Bluetooth applications.

Ethernet

Ethernet was developed at Xerox Palo Alto Research center in 1973. The pilot Ethernet was the first high-speed local network. It lets computers to connect at a high speed and low error rates and without switching delays. An Ethernet is a simple bus like connection line using a transmission line and Hubs plus repeaters are helping to transmit the same signals across the network. Linked ethernet appears as a single network to higher protocol networks. The ARP protocol can resolve the IP address to the Ethernet addresses across the network.

There are several methods of communication in Ethernet networks.

Packet broadcasting

This method id broadcasting data on the transmission medium. Every station is listening to the medium for the packets addressed to them. Data transmission proceeds to 10 Mbps or at higher speeds are from 100 to 1000 Mbps. The address of the destination station is normally referred to as a single network interface. The Ethernet network protocol is implemented in the Ethernet hardware interface; protocol software id required for the transport layer and above it.

Packet Collision

Even it is short time is taken to send messages there a probability to have a collision between messages sent. If there is no check when sending messages the collision happens a lot. The Ethernet has multiple mechanisms to deal with collisions.

First, is called (*carrier sensing)*. In carrier sensing the interface, the hardware is listening to the presence of the signal in the medium but the technique does not prevent all collisions due to the finite time for a signal inserted at a point in a medium.

Another called *(collision detection)*.

Whenever a packet transmitted through a hardware output port it also listens to its input port and then two signals are compared. If they differ then a collision happened. If this occurs then all stations stop transmission to detect the collision but is the packets that they transmit take more time to broadcast then the collision will not be notified since each sending station will not see the other packet transmitted until it finishes transmitting its own whereas the stations in the middle will receive the packets causing a collision.

Ethernet is a way of connecting computers together and it was the most widely used method of connecting.

FLOW CHART

UNIT III

EMBEDDED SYSTEM SOFTWARE DESIGN

1. APPLICATION SOFTWARE

Application software is a computer program that performs specialized duties. Application software can carry out a variety of tasks, including personal, professional, and academic ones. Application software is frequently referred to as productivity software or end-user software. Each piece of software is designed to help users with a certain productivity, efficiency, or communication process.

Processing and spreadsheet programs like Microsoft Word and Excel, as well as web browsers like Firefox and Google Chrome. Besides games like Candy Crush Saga and Ludo, it also contains smartphone apps like WhatsApp and Telegram. Some applications connect customers with their businesses and popular services that people use daily, including weather or transportation information.

In contrast to System software, application software is focused on its functionality and completes the goal for which it was created. Most of the apps we use on our smartphones are examples of application software. Every piece of application software's main objective is to make a task easier for users to complete.

Need of Application Software

Application software allows end users to carry out a variety of single and multiple actions. Here are a few factors that make application software (App) necessary for your computer:

Assists the user in carrying out particular tasks: The end user should be kept in mind when developing any application program. Users can create, edit, remove, and carry out other Word document operations using Microsoft Word, a widely used application software. They will primarily help the end-user by enabling them to carry out specific tasks in various fields, including education, business, and entertainment.

Manages and modifies data: Organizations utilize application software to manage and modify employee, customer, and other databases. Examples of application software include customer relationship management and enterprise resource management systems.

Allows users to arrange information efficiently: Individual users can produce and manage data with application software. For instance, Microsoft Excel is a widely popular program in businesses and enables users to handle datasheets.

Functions of Application Software

These software or mobile applications enhance users' functions like creativity, communication, productivity, efficiency, and amusement. Furthermore, they can help with data analysis, calculation, and resource coordination. The range of utility for such programs can be broad, with each one allowing end users to carry out particular tasks. Different apps may simultaneously offer help for various tasks or in a few areas depending on their functionality and nature

A few of the functions of application software include:

FLOW CHART

management of information and data

data management and analysis

document management (document exchange systems)

creation of images and videos

Several choices include texting, conferencing over audio and video, and working together.

Accounting, financial, and payroll management

the management of resources (ERP and CRM systems)

managing a project

management over corporate operations

software for use in schools (LMS and e-learning systems)

software for use in healthcare

Types of Application Software

Every company sector needs application software because we live in a digital age. Every industry, including banking, healthcare, education, retail, travel, logistics, etc., is heavily utilizing the program. Choosing the best application software for your unique needs enhances performance and effectiveness. You may cut costs, save time, and use fewer resources if you are aware of the many kinds of application software. You can also increase productivity and make better decisions. General applications, business applications, and specially developed applications can all be broadly categorized as application software.

Types of Application Software are divided into two categories:

General Application Software

Customized Application Software

1. General Application Software

Numerous fundamental tasks can be accomplished by general software. They finish all of the standard tasks a user must carry out on the system. The user must complete several applications. This category includes a wide range of frequently used applications. These software programs can be purchased alone or as a suite of related programs.

Further divisions of general application software include:

Word Processing Software

Text can be edited and formatted; documents, including memos, letters, faxes, and documents, are created using word processing software. The text is formatted and made beautiful using processing software. Software for word processors enables the creation, enhancement, and manipulation of text. They offer a wide range of capabilities that enable efficient text consolidation and editing. The user can input, edit, format, and output text with the aid of this software. There is a list of features provided.

FLOW CHART

Additionally, by giving users access to a thesaurus, synonyms, and other tools, this kind of software creates an amazing visual experience. The software also features font sizes, colors, and styles. For instance, Corel WordPerfect, Lotus Word Pro, Word pad, and Microsoft Word.

Business Application Software

This business application software was created to automate corporate processes and meet users' demands for increased operational efficiency and accuracy. The business application software's output is expected to increase productivity and profitability. Business application software is divided into different categories based on its unique requirements.

Database Software

A program called a database management system, or DBMS, is designed for retrieving, storing, and searching data from a single database. This program, also called database management software, aids in efficient data management. This program makes it simple to organize data and access it. Software development services automatically create and save data when they create an application.

The database management system (DBMS) helps the program get the necessary data and store it after it has been finished using it. This program modifies data to gain access to it while searching DBMS. A few database management solutions assist in managing the system's diverse data, including email addresses, phone numbers, catalogs, etc. MS Access, Oracle, and MySQL are a few DBMS types.

Presentation Software

Presentation software lets you present your thoughts and ideas as visual data. After that, you can use slides to display the information. You may add videos, words, charts, graphs, and photographs to your presentations to make them more engaging and educational.

Key components of presentation software include:

Text editor: The text editor feature of presentation software allows users to input and design their text.

Multimedia files: The ability to add and animate graphics, texts, multimedia, and movies is provided by presentation software.

Slideshow: Presentation software can also assist in creating a slideshow for the presentation.

Web Browsers

These sorts of application software help consumers browse websites and conduct research. They support users' information retrieval and web exploration. Internet Explorer, Chrome, Firefox, MS Edge, Safari, and more popular web browsers are available. Users with these browsers can utilize search engines like Google, Bing, Yahoo, and others to explore the web and conduct searches.

Web browsers are software programs that make it simple to browse the internet. These can be used to find information on the web quickly.

FLOW CHART

Education Software

Education software refers to any program that improves the educational process. In classroom settings, teachers use this software to transfer knowledge to students. Education software, for instance, makes it simpler to deliver lectures with visual and aural experiences to make learning enjoyable and simple. ProProfs, Schoology, Google Classroom, TalentLMS, Litmos, and other industry standards for educational software are available.

Graphic Design Software

Graphics software is used to make changes to visual data, images, and animation. It includes various editorial software. Digital photos or films can be quickly edited using graphic design software. This software includes tools for creating and altering images. Software for graphic design includes Adobe Photoshop, Clip Studio, and Adobe Illustrator, to name a few.

Simulation Software

System behavior can be predicted using simulation software. Using simulation software, you may assess new designs, identify issues with old designs, and test a system in challenging environments like an orbiting satellite. This is mostly useful in video games or training exercises for the workplace. Software development businesses can create simulation software to acquaint learners with the surroundings before handling the actual machinery when using genuine machinery is only possible with training. Examples of simulation software include SOLIDWORKS and Teamcenter.

2. Customized Application Software

Software specifically tailored for a given user group or organization is created with business challenges in mind. Either an internal development team or an outside company creates custom solutions. However, the technique and development process is the same as other software development. Take a look at Uber, a popular on-demand service that uses specialized software to offer drivers and passengers a seamless experience.

This application software is nowhere to be found online. Therefore, we are unable to download directly from anywhere. Additionally, if you employ software developers to create a computer application, they must deliver it to you under specific conditions. As a result, it's also known as a bespoke application or tailor-made software. This category includes things like the salon's computer application or application software.

Office Software

Microsoft Word: Microsoft Word is one of the most widely used word-processing applications. This program, a component of the Microsoft Office Suite, was created by Microsoft. This tool can also be used to create visuals. Even though it lacks the strength of a graphic tool, most people with little to no experience in graphic design find it simple to use. Microsoft Word includes a text editor, paragraph and font formatting, grammar and spell-checking HTML support, and picture support.

Microsoft Excel: Microsoft Excel is an additional application software from the company. Utilize this spreadsheet application to perform computations. Time, date, text, and numeric fields are all defined by different cells in MS Excel. You can carry out your computations and supply the formulas and functions. Additionally, you can create analyses and visuals with this program.

Microsoft PowerPoint: Microsoft PowerPoint is a presentation application software that allows you to graphically present your thoughts to your audience. You can present information in slide format with this application. It has built-in templates, clip art, audio, and video capabilities to improve your presentations. You can use MS PowerPoint as part of the Microsoft Office Suite for personal and professional purposes.

2) Graphics and Design Software

Adobe Photoshop: Adobe Photoshop is a software tool for graphic design and picture manipulation. You can edit videos, produce, modify, and organize digital images and graphic work, develop designs and take photographs. You may create any concept art using this program from the comfort of your computer.

AutoCAD: AutoCAD is the most popular CAD drawing application program. With this program, you can document your workflow more effectively and easily. It also performs quite well. Its built-in tools may connect with computer-aided manufacturing software to create machine tool settings for various daily tasks. Both Mac and Windows operating systems are compatible.

3) Open-Source Software

MySQL: The open-source database management application software MySQL is very popular. This utility anchors the back end of various applications because of its strong capabilities. This software's broad compatibility is its best feature. It integrates seamlessly with cPanel and other server management interfaces. Compatibility with shared data from other databases is another feature it offers.

4) Communication and Multimedia Software

Skype: You can make video conversations using the software program Skype with your friends, coworkers, family, business partners, clients, and more. You can talk with folks to discuss projects or other topics if you don't want to video call. Skype makes it easier for people and gadgets to communicate with one another.

Spotify: Spotify is an audio player application for Windows 10. You can use the tool to find the podcasts or music you're looking for. There are several different tracks and episodes included. The program lets you choose your preferred music filter and share your podcast and music. Windows, Mac, Linux, and iOS are all supported.

5) Document Viewer Software

Windows Photo Viewer: Another application software for seeing photographs is Windows Photo Viewer. It has become simple to capture several images thanks to the development of webcams, smartphone cameras, and other digital cameras. Slideshows of your photos can be played, and favorites can be noted, and more.

6) Enterprise Software

FLOW CHART

Salesforce: Businesses utilize the application software Salesforce to keep up with customer interactions. Your company needs to keep open lines of communication between the executives, departments, staff, and clients. This tool streamlines procedures by drawing on prior knowledge and user comments.

Forecast: Forecast is a whole suite of application software for companies that provide professional services. The forecasting process aims to make projects more profitable and predictable. Utilizing project automation aids in resource planning and financial and operational tuning.

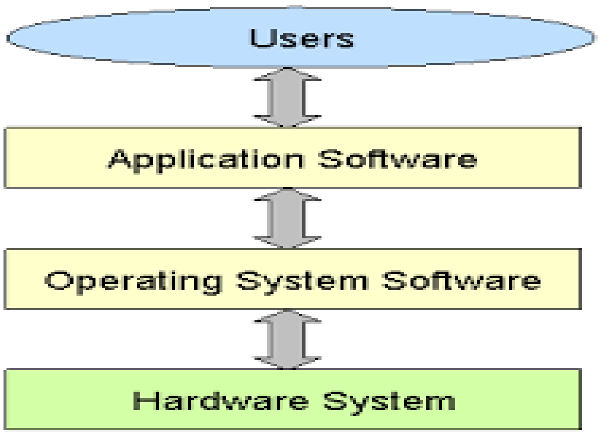
7) Other Popular Application Software

Google Chrome: Google Chrome is application software that also allows you to access social media networks and send/receive emails. You can search on any subject with this browser, including technology, travel, politics, medicine, and a wide range of other subjects. You can retrieve data from across the web with Google Chrome.

WinZip: WinZip is a file compression and management program. Using this program, you may compress files, including folders, PDFs, MP3s, videos, and more. Unzipping directories also allow you to decompress them. Additionally, you can extract various files from Zip files and repair them.

2.SYSTEM SOFTWARE

System software is a category of computer software designed to manage and control the hardware and provide a platform for running application software. It serves as an intermediary between the hardware components of a computer system and the user-facing application software. System software includes a variety of essential components that enable the computer system to function properly and efficiently. Here are some key components of system software:



1. Operating System (OS):

The operating system is a crucial piece of system software that manages hardware resources, provides a user interface, and facilitates communication between hardware and application software. Popular operating systems include Microsoft Windows, macOS, Linux, and various flavors of Unix.

FLOW CHART

2. Device Drivers:

Device drivers are software modules that enable communication between the operating system and hardware devices, such as printers, graphics cards, network adapters, and storage devices. They allow the OS and application software to control and utilize hardware resources effectively.

3. Firmware:

Firmware is a type of software that is embedded in hardware devices. It provides low-level control and initialization routines for hardware components, ensuring they operate correctly and consistently. Examples include BIOS (Basic Input/Output System) and firmware in embedded systems like routers and embedded controllers.

4. Utility Software:

Utility software includes tools designed to perform specific system-related tasks, such as system optimization, data backup, antivirus scanning, disk maintenance, and file management. Examples include disk defragmentation tools, backup software, and antivirus programs.

5. Language Translators:

Compilers, interpreters, and assemblers are language translators that convert high-level programming code into machine code that the computer's hardware can execute. Compilers are used for languages like C and C++, while interpreters are used for languages like Python and JavaScript.

6. System Libraries:

System libraries are collections of pre-written code that provide essential functions and services to application software. Programmers can use these libraries to simplify and speed up software development by leveraging existing code.

7. Virtualization Software:

Virtualization software allows multiple operating systems to run on a single physical machine. This technology is used to create virtual environments that can host multiple virtual machines (VMs), each running its own operating system and applications.

8. Middleware:

Middleware is software that acts as an intermediary between different software applications or components. It facilitates communication, data exchange, and integration between different software systems.

9. System Tools:

System tools are software applications that help users manage and configure their computer systems. These tools include task managers, performance monitors, system information utilities, and network configuration tools.

10. Boot Loaders:

FLOW CHART

Boot loaders are programs that initiate the computer's boot process. They load the operating system into memory and start its execution. GRUB (Grand Unified Bootloader) is an example of a boot loader commonly used with Linux.

3. MODEL-BASED SYSTEM ENGINEERING (MBSE)

MBSE in a digital-modeling environment provides advantages that document-based systems engineering cannot provide. For example, in a document-based approach, many documents are generated by different authors to capture the system's design from various stakeholder views, such as system behavior, software, hardware, safety, security, or other disciplines. Using a digital-modeling approach, a single source of truth for the system is built in which discipline-specific views of the system are created using the same model elements.

A digital-modeling environment also creates a common standards-based approach to documenting the system that can be programmatically validated to remove inconsistencies within the models and enforce the use of a standard by all stakeholders. This common modeling environment improves the analysis of the system and reduces the number of defects that are commonly injected in a traditional document-based approach. The availability of digitalized system data for analysis across disciplines provides consistent propagation of corrections and incorporation of new information and design decisions (i.e., state it once and automatically propagate to various views of the data) to all stakeholders. When MBSE is done properly, the result is an overall reduction of development risks.

MBSE brings together three concepts: model, systems thinking, and systems engineering:

- A **model** is a simplified version of something--a graphical, mathematical, or physical representation that abstracts reality to eliminate some complexity. This definition implies formality or rules in simplifying, representing, or abstracting. To model a system, a systems architect must represent the system with less detail so that its structure and behavior are apparent and its complexity is manageable. In other words, models should sufficiently represent the system, and the system should confirm the models.
- **Systems thinking** is a way of looking at a system under consideration not as a self-sufficient entity, but as part of a larger system. Systems thinking is not the same as a systematic adherence to following good plans, collecting statistics, or being methodical. The systems engineer observes the system from a distance; explores its boundaries, context, and lifecycle; notes its behavior; and identifies patterns. This method can help the engineer to identify issues (e.g., missing interaction, a missing step in a process, duplication of effort, missed opportunity for automation) and manage a system's complexity. Although systems engineers must break down and analyze the system in the beginning--identify parts and describe connections between them--with systems thinking, they later synthesize the parts back into a coherent whole. Parts are not just connected to other parts, they depend on each other to work properly. Systems thinking emphasizes this interconnectedness. The behavior of the system emerges from the activities of the system's subparts. Observing the system's interconnections, the systems engineer identifies feedback loops and causality patterns that may not be apparent at first. Systems thinking can help make issues more apparent and easier to identify, balance the system, and manage the system's complexity.

FLOW CHART

- **Systems engineering** is a transdisciplinary and integrative approach to enable the successful realization, use, and retirement of engineered systems, using systems principles and concepts, and scientific, technological, and management methods. It brings together a number of techniques to make sure that all requirements are satisfied by the designed system. It concentrates on architecture, implementation, integration, analysis, and management of a system during its lifecycle. It also considers software, hardware, personnel, processes, and procedural aspects of the system.

If an organization has decided to adopt MBSE as an internal systems-engineering approach and chosen one of the four or five existing products for digital modeling that are on the market, the organization's systems engineers should consider whether it is going to follow any architectural frameworks. Although a comprehensive discussion of this topic is beyond the scope for this blog post, the choice of a particular architectural framework will provide additional guidance and structure to the modeling activities, especially if the systems engineers are already familiar with the framework.

MBSE is a multidisciplinary and multifaceted endeavor. It requires its own actors, processes, environment, and information flows. To create a successful model of a complex system or system of systems, an organization must support the modeling process. The support needed is not much different from what is required for an organization to successfully develop and deliver a complex system or system of systems. MBSE can be effectively integrated into a development process, but the organization must commit to the effort that will be required to model the system.

Applying systems thinking, we can recognize that there are three systems involved in the modeling process: the designed system, the designed system's context, and the modeling organization for the designed system. The designed system operates in the context of a larger system, and the modeling organization must understand both the designed system and the designed system's context. The organization must also be aware of its own behavior, successes, and failures.

Modeling

We have all seen, used, or created models throughout our lives, ranging from toys that represent cars or planes to mathematical formulas that describe and explain physical phenomena such as thermodynamics or gravity. While fundamentally different, those models all connect an idea to a reality and provide sufficient abstraction for the purpose. When modeling a system, the systems engineer decides what aspects of the production system are most important, such as structure, energy or matter flow, internal communication, or safety and security. Those types of aspects will become the focus of the model. The top objective of the modeling activity is to model the salient aspects on which the model is focused as closely to the real system as is possible and feasible.

Modeling as a technique uses four instruments:

- language
- structure
- argumentation
- presentation

FLOW CHART

A modeling language is a common terminology for clearly communicating an abstract idea that the model captures. The modeling language can be formal, with strict syntax and rules. A few system-modeling languages exist, including general-purpose languages such as the Systems Modeling Language (SysML) and Unified Modeling Language (UML), as well as specialized languages such as Architecture Analysis Design Language (AADL). Although SysML and UML are not mathematically formal, a valid model requires that the modeling language's rules for entities and relationships be followed. SysML has strict syntax and rules for relationships and connections between elements, which helps to avoid ambiguity. If a model is well built, several types of standard SysML diagrams can be dynamically simulated, and at least one type of SysML diagram can be mathematically simulated. UML is semi-formal; SysML is similar to UML, but more formal.

A model must have a structure. A well-structured model can make the model understandable, usable, and maintainable, which is particularly important for complex systems. The goal of a model is to show stakeholders that the presented design satisfies the system's requirements. The model should demonstrate, in an easily comprehensible way, how the system must be built to be successful. Visualization is a key way to ensure comprehensibility. Visualizing abstract ideas enables people to take the leap of imagination that is needed to "see" the system.

Modeling Domains

Even though MBSE does not dictate any specific process, essentially any process chosen should cover four systems-engineering domains:

- requirements/capabilities
- behavior
- architecture/structure
- verification and validation

Descriptions of these domains are well documented and discussed by, among others, Defense Acquisition University (DAU), NASA, and Avi Sharma. The difference that MBSE makes is that these fundamental systems-engineering domains are defined not as a set of documents, but in the model itself, i.e., in a formal way using a modeling language. The model represents an argument for how the system must be designed for it to be successful.

MBSE also fosters communication among stakeholders, systems engineers, and developers. Since system design is performed in the integrated modeling environment, all systems engineers, managers, and other stakeholders can have access to the generated information--such as requirements, behavior flows, and architecture--as soon as necessary.

The most common modeling activity is the creation of diagrams representing some part of the system--a view. This activity is so common that some engineers mistakenly equate creating a view with creating a model. This mistake is so pervasive that there is even an emerging term for it: *zombie model*. This term refers to a model that is full of diagrams, but with no interconnectivity and dependencies identified among the elements.

Anyone who is about to start modeling must realize that a set of views is not a model. Although a view or even a set of views can represent a part of the system's design and can be useful for documenting and communicating some aspects of the system, views are only facets

FLOW CHART

or portions of the true system model. A real model can produce many views and matrices, perform analyses, and run simulations.

Language of System Modeling

While a system-modeling language such as SysML is a formal syntactic language, it is still based on elements of human language. Its formality adds clarity and discipline that are critical for describing the design of a system. Such a language is easy to read and understand. Terms of MBSE's language simply map to parts of speech:

- noun: actors, blocks, components, requirements
- verb: operational activities, functions, use cases
- adjective: attributes
- adverb: relationships, needlines, exchanges, interfaces

This view of the modeling language helps its users to mentally map real-life concepts to abstract ideas, and eases the formalization of the modeling process.

Four Quadrants of the MBSE Model

Now that I have described the basics of a model's language and domains, I will describe the modeling approach. A model must describe both a problem that the designed system solves, and the designed system itself (the solution). The model must have these two sides, the problem side and the solution side. These are sometimes referred to as the *operational* and *system* points of view.

The operational point of view is the perspective of users, operators, and business people. It should represent business processes, objectives, organizational structure, use cases, and information flows. The operational side of the model can contain the description of "the world as-is" and the future state.

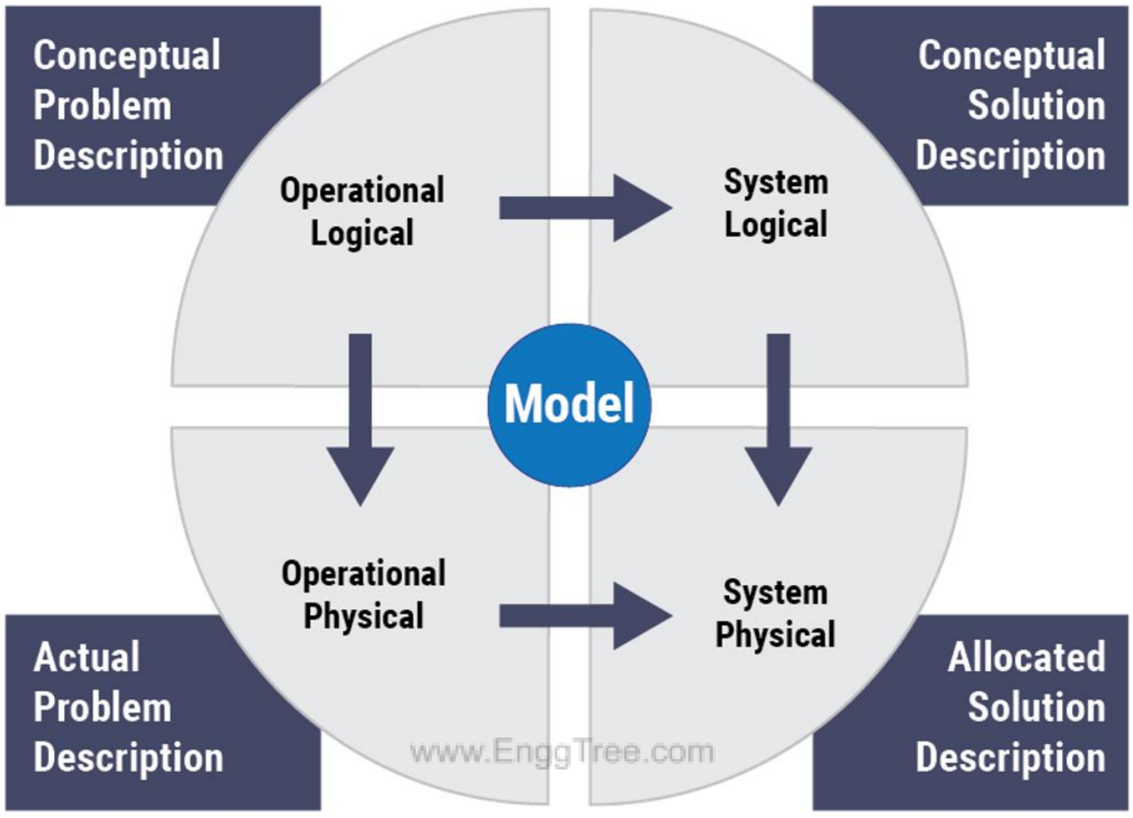
The system point of view is the solution, the architecture of the system that solves the problem posed in the operational side of the model. It should describe the behavior of the system, its structure, dataflows between components, and allocation of functionality. It should describe how the system will be deployed in the real world. It can contain solution alternatives and analyses of them.

Each of these points of view has two parts, logical and physical. Separating logical and physical aspects of the model is a way to manage a system's complexity. Logical parts of the model usually change little over time, while physical changes are often initiated by technology advances.

If the model is built properly, all four quadrants should be tightly connected, as shown in Figure 1 below. Statements of the problem should be traced to elements of the solution, and logical elements allocated to physical structures. The user of the model should be able to see clearly how the top-level concepts and components decompose to the lower level features. Users should be able to perform system analysis, create dependency matrices, run simulations, and produce a view of the system for every stakeholder. If the physical part of

FLOW CHART

the system must change, the logical side of the model identifies exactly what functionality will be affected. If a requirement or business process must be changed, the model will easily discover the impact on the solutions.


Conceptual Problem Description
Conceptual Solution Description
Operational Logical
System Logical
Model
Operational Physical
System Physical
Actual Problem Description
Allocated Solution Description
www.EnggTree.com

Components Of a Model

4. Use of High-Level Languages- embedded C / C++ Programming

Use of High-Level Languages- embedded C

Embedded C is a variant of the C programming language that is specifically tailored for developing software for embedded systems. It retains many features of the standard C language but includes certain extensions and conventions to accommodate the unique characteristics of embedded systems. Here's how Embedded C is used in the context of embedded systems development:

1. Low-Level Hardware Interaction:Embedded C allows developers to interact directly with hardware peripherals, such as timers, GPIO pins, and communication interfaces (SPI, I2C, UART), using memory-mapped registers. This level of control is essential for configuring and controlling hardware components efficiently.

2. Memory Management:Embedded C gives developers the ability to manage memory manually, which is crucial in resource-constrained environments. They can allocate memory

FLOW CHART

dynamically using functions like `malloc` and free it with `free`, though careful management is needed to avoid memory leaks and fragmentation.

3. Efficient Code Execution:Since embedded systems often have limited processing power and memory, Embedded C allows developers to write efficient code that maximizes performance. Developers can control aspects like loop unrolling, inline assembly, and compiler optimizations to squeeze the most out of the hardware.

4. Real-Time Applications:Embedded C is well-suited for real-time applications where timing constraints are critical. Developers can write time-sensitive code and manage interrupts effectively to meet real-time requirements.

5. Interrupt Handling:Embedded C enables developers to write interrupt service routines (ISRs) that respond quickly and accurately to hardware-generated interrupts. This is essential for handling events like button presses, sensor readings, and communication events.

6. Custom Data Structures:Embedded C supports the creation of custom data structures, which allows developers to organize and manage data efficiently. This is especially important when dealing with complex sensor data or communication protocols.

7. Efficient Algorithms:Developing efficient algorithms is important in embedded systems due to limited processing power. Embedded C enables developers to implement algorithms that are tailored to the hardware's capabilities.

8. Portability:While not as portable as higher-level languages, Embedded C still offers a degree of portability across architectures. Developers can often reuse and adapt code for different microcontrollers with minimal changes.

9. Compiler-Specific Features:Different microcontroller manufacturers often provide their own compilers and toolchains with specific extensions. Embedded C allows developers to utilize these extensions to interact with the hardware in a more optimized manner.

10. Custom Hardware Control:Embedded C allows developers to create custom device drivers to control specific hardware components. This level of control is necessary when dealing with unique peripherals or communication interfaces.

11. Small Memory Footprint:Embedded C code can be optimized to minimize the memory footprint, which is crucial in devices with limited memory. This includes techniques like using bitfields to save memory space.

12.Bare-Metal Programming:Embedded C allows developers to write code that runs directly on the hardware without the overhead of an operating system. This is common in resource-constrained embedded systems.

Overall, Embedded C strikes a balance between low-level control and higher-level programming, making it a versatile choice for developing software for a wide range of embedded systems, from microcontrollers in consumer electronics to automotive control units and industrial automation systems.

FLOW CHART

4.1 Use of High-Level Languages- embedded C++ Programming

Embedded C++ programming combines the features of the C++ programming language with the demands of embedded systems development. This approach allows developers to harness the power of object-oriented programming and modern C++ features while addressing the constraints of resource-constrained embedded environments. Here's how embedded C++ programming is used:

1. Object-Oriented Programming (OOP):C++ supports object-oriented programming, allowing developers to create modular and reusable code through classes and objects. This is particularly valuable for building complex embedded systems with multiple components and interactions.

2. Abstraction and Encapsulation: Embedded C++ encourages the use of abstraction and encapsulation, which helps isolate hardware-specific details from higher-level application logic. This promotes code maintainability and simplifies future updates.

3. Code Reusability: In embedded C++, developers can create libraries and classes that abstract common functionalities, making it easier to reuse code across different projects and platforms. This enhances development efficiency.

4. Resource Management: Embedded C++ offers features like constructors and destructors, which can be used to manage resources efficiently. Smart pointers and RAII (Resource Acquisition Is Initialization) can help prevent resource leaks in complex applications.

5. Template Metaprogramming: While careful usage is necessary due to potential code bloat, template metaprogramming in C++ can lead to code optimization through compile-time computations. This is beneficial in embedded systems where runtime efficiency is crucial.

6. STL and Data Structures: The C++ Standard Template Library (STL) provides a range of data structures and algorithms. These can be particularly useful for managing data and optimizing code for various embedded applications.

7. Real-Time and Multithreading: C++ supports multithreading and real-time applications, allowing developers to create responsive systems with multiple concurrent tasks. The `<thread>` and `<mutex>` libraries can help manage parallel execution.

8. Operator Overloading: Embedded C++ enables developers to overload operators to work with custom data types. This can simplify complex calculations and data manipulations, making code more readable.

9. Hardware Abstraction: Embedded C++ allows developers to create hardware abstraction layers (HALs) using classes and functions. This makes it easier to adapt code to different hardware platforms without rewriting the entire application.

10. Low-Level Access: Embedded C++ supports low-level operations, such as direct memory access and pointer manipulation, which can be essential when interfacing with hardware peripherals.

11. Custom Memory Management: Developers can implement custom memory management strategies using C++ features like operator overloading, placement `new`, and memory pools. This is useful for optimizing memory usage in resource-constrained systems.

FLOW CHART

12. Compiler and Toolchain Support:Many modern embedded systems toolchains provide support for C++ programming. Developers can take advantage of compiler optimizations and extensions for better performance.

13. Safety and Reliability:With proper usage, C++ features like exceptions and the type system can enhance code safety and reliability. However, careful consideration is needed to ensure minimal impact on performance and memory usage.

14. Mixed-Language Development: In some cases, developers might use a mix of C and C++ in an embedded project, combining the strengths of both languages. This can be particularly useful when interfacing with legacy code or hardware-specific code.

5. INTEGRATED DEVELOPMENT ENVIRONMENT TOOLS

Embedded C++ programming is suitable for a range of applications, from consumer electronics to automotive systems, medical devices, and industrial automation. However, developers must be mindful of code size, runtime efficiency, and memory consumption to ensure optimal performance in resource-limited environments.

Certainly, let's break down the key components and tools commonly found in an Integrated Development Environment (IDE) for embedded systems development:

1. Editor:

The code editor is where you write and edit your source code. It often includes features like syntax highlighting, code completion, code folding, and code navigation to help you write code more efficiently.

2. Compiler:

The compiler translates your source code written in high-level programming languages (like C or C++) into machine code that can be executed by the target microcontroller or processor. It checks for syntax errors and performs various optimizations to generate efficient code.

3. Linker:

The linker takes the output of the compiler (object files) and combines them into a single executable file. It resolves references between different parts of the code and generates the final binary image that can be loaded onto the target hardware.

4. Automatic Code Generators:

Some IDEs provide tools that generate code automatically based on specific configurations or requirements. These generators can help create initialization code, drivers, or application frameworks based on user inputs.

5. Debugger:

The debugger is a crucial tool for identifying and fixing issues in your code. It allows you to set breakpoints, step through code execution, inspect variables, and track program flow. Embedded debuggers might also support features like real-time memory inspection and peripheral register visualization.

FLOW CHART

6. Board Support Library (BSL):

A Board Support Library is a collection of code and drivers specific to a particular development board or hardware platform. It provides an abstraction layer that simplifies interacting with the board's peripherals and hardware features.

7. Chip Support Library (CSL):

A Chip Support Library provides a collection of low-level drivers and APIs for a specific microcontroller or processor family. It offers an interface to interact with the microcontroller's hardware features, such as GPIO, timers, UARTs, and more.

8. IDE-Specific Tools:

Many IDEs offer additional tools and features to enhance development:

- Version Control Integration:Integration with version control systems like Git allows teams to collaborate on projects more effectively.

- Project Management:Tools for organizing and managing project files, configurations, and build settings.

- Code Generation Tools: Some IDEs include code generators for specific tasks, like generating state machines or communication protocols.

- Simulators and Emulators:Simulators and emulators allow you to test your code on the development machine before deploying it to the target hardware.

- Profiling and Optimization:Tools for analyzing code performance and optimizing it for memory and execution speed.

These components work together to provide a unified environment for developing, testing, and debugging embedded systems software. When choosing an IDE, it's important to consider factors such as hardware compatibility, toolchain support, available libraries, debugging capabilities, and personal preferences.

6. Analysis and Optimization-Execution Time- Energy & Power

In embedded systems development, analysis and optimization of execution time, energy consumption, and power usage are critical to creating efficient and reliable software for resource-constrained environments. Here's how these factors are addressed:

1. Execution Time Analysis and Optimization:

- Profiling: Profiling tools help identify which parts of your code consume the most execution time. This information allows you to focus your optimization efforts on critical sections.

- Loop Optimization: Optimizing loops by reducing iterations, minimizing branching, and using efficient algorithms can significantly improve execution time.

- Compiler Optimization Flags:Modern compilers offer various optimization levels and flags that can improve code execution speed. These include options for loop unrolling, inlining functions, and optimizing for size or speed.

FLOW CHART

- Parallelism and Multithreading: If the hardware supports it, using multiple cores or threads can help distribute tasks and improve overall execution speed.

- Cache Optimization: Optimizing memory access patterns to take advantage of cache hierarchies can significantly reduce memory access latency.

- Reducing Code Size:Smaller code requires less time to fetch from memory, leading to faster execution. Removing dead code and using smaller data types can help in this aspect.

2. Energy and Power Analysis and Optimization:

- Power Profiling:Tools can measure the energy consumption of different parts of the system. This profiling helps identify power-hungry components and optimize their usage.

- Clock and Voltage Scaling:Many processors and microcontrollers offer clock and voltage scaling options that allow you to adjust the CPU frequency and voltage levels based on workload, saving power during periods of lower demand.

- Idle Modes: Utilizing low-power modes during periods of inactivity can significantly reduce energy consumption.

- Peripheral Management: Turning off or reducing the frequency of unused or less critical peripherals can lead to substantial power savings.

- Efficient Algorithms: Choosing algorithms that require fewer calculations or memory accesses can reduce CPU usage and subsequently lower energy consumption.

- Duty Cycling:For wireless communication, duty cycling involves turning off the radio module when it's not actively transmitting or receiving data.

- Dynamic Voltage and Frequency Scaling (DVFS): Adjusting the CPU's voltage and frequency in real-time based on workload can lead to energy savings.

- Power Gating:Power gating involves completely shutting down certain parts of the chip when they're not needed, further reducing power consumption.

3. Trade-offs and Considerations:

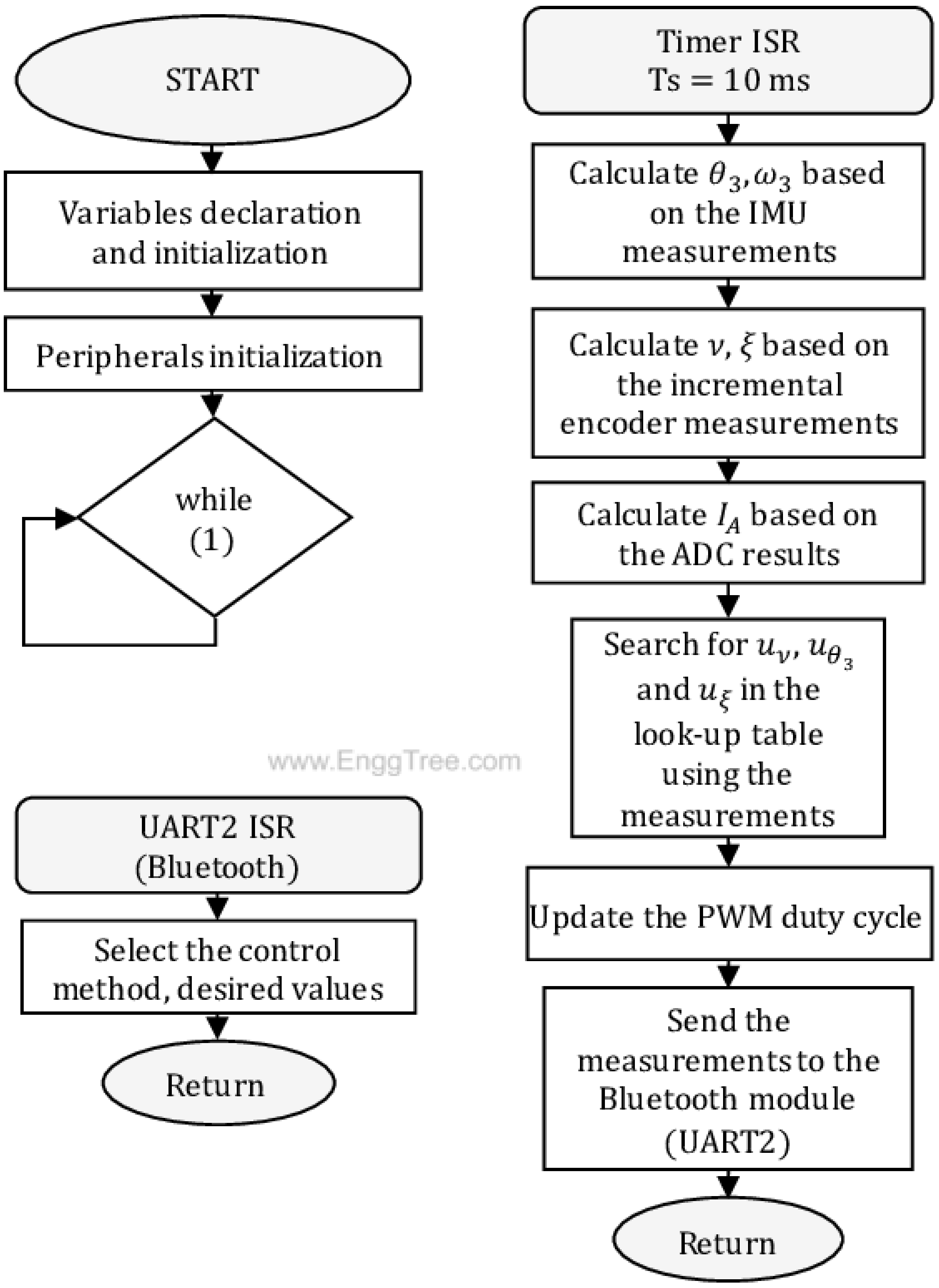
- Real-Time Requirements:Optimizations should not compromise real-time performance if the system has time-sensitive tasks.

- Hardware Limitations:Different hardware platforms might have different power-saving mechanisms and limitations. Optimizations should consider the capabilities of the target hardware.

- Accuracy vs. Efficiency: Energy and power measurement tools might introduce overhead, impacting accuracy. Striking a balance between accuracy and overall efficiency is important.

Effective analysis and optimization of execution time, energy consumption, and power usage require a deep understanding of the hardware, software, and the specific requirements of the embedded system. It's a careful balancing act to achieve the desired performance and energy efficiency without sacrificing reliability or functionality.

FLOW CHART


START
Variables declaration and initialization
Peripherals initialization
while (1)
UARTZ ISR (Bluetooth)
Select the control method, desired values
Return
Timer ISR Ts = 10 ms
Calculate θ_3, ω_3 based on the IMU measurements
Calculate v, ξ based on the incremental encoder measurements
Calculate I_A based on the ADC results
Search for u_v, u_{θ_3} and u_ξ in the look-up table using the measurements
Update the PWM duty cycle
Send the measurements to the Bluetooth module (UART2)
Return

UNIT 4

DESIGN AND DEVELOPMENT OF IoT

4.1 INTRODUCTION

The Internet of Things (IoT) describes the network of physical objects—"things"—that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the internet.

Internet of Things (IoT) is the networking of physical objects that contain electronics embedded within their architecture in order to communicate and sense interactions amongst each other or with respect to the external environment. In the upcoming years, IoT-based technology will offer advanced levels of services and practically change the way people lead their daily lives. Advancements in medicine, power, gene therapies, agriculture, smart cities, and smart homes are just a very few of the categorical examples where IoT is strongly established.
IoT is network of interconnected computing devices which are embedded in everyday objects, enabling them to send and receive data.

4.2 DEFINITION

A dynamic global network infrastructure with self-configuring capabilities based on standard and interoperable communication protocols where physical and virtual "things" have identities, physical attributes, and virtual personalities and use intelligent interfaces, and are seamlessly integrated into the information network, often communicate data associated with users and their environments.

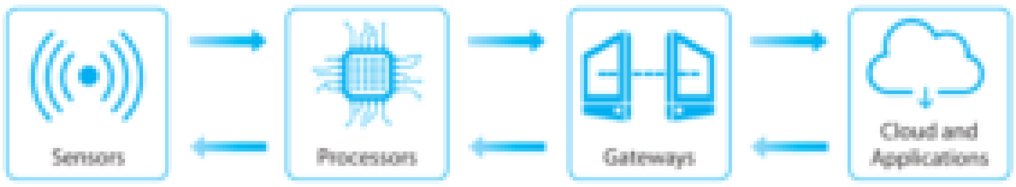
4.3 CHARACTERISTICS OF IoT

- **Dynamic & Self-Adapting:** IoT devices and systems may have the capability to dynamically adapt with the changing contexts and take actions based on their operating conditions, user's context, or sensed environment. For example, consider a surveillance system comprising of a number of surveillance cameras. The surveillance cameras can adapt their modes (to normal or infra-red night modes) based on whether it is day or night. Cameras could switch from lower resolution to higher resolution modes when any motion is detected and alert nearby cameras to do the same. In this example, the surveillance system is adapting itself based on the context and changing (e.g.. dynamic) conditions.
- **Self-Configuring:** IoT devices may have self-configuring capability, allowing a large number of devices to work together to provide certain functionality (such as weather monitoring). These devices have the ability configure themselves (in association with the IoT infrastructure), setup the networking, and fetch latest software upgrades with minimal manual or user intervention.
- **Interoperable Communication Protocols:** IoT devices may support a number of interoperable communication protocols and can communicate with other devices and also with the infrastructure. We describe some of the commonly used communication protocols and models in later sections.
- **Unique Identity:** Each IoT device has a unique identity and a unique identifier (such as an IP address or a URI). IoT systems may have intelligent interfaces which adapt based on the context, allow communicating with users and the environmental contexts. IoT device interfaces allow users to query the devices, monitor their status, and control them remotely, in association with the control, configuration and management infrastructure.
- **Integrated into Information Network:** IoT devices are usually integrated into the information network that allows them to communicate and exchange data with other devices and systems. IoT devices can be dynamically discovered in the network, by other devices and/or the network, and have the capability to describe themselves (and their characteristics) to other devices or user

applications. For example, a weather monitoring node can describe its monitoring capabilities to another connected node so that they can communicate and exchange data. Integration into the information network helps in making loT systems "smarter" due to the collective intelligence of the individual devices in collaboration with the infrastructure. Thus, the data from a large number of connected weather monitoring loT nodes can be aggregated and analyzed to predict the weather.

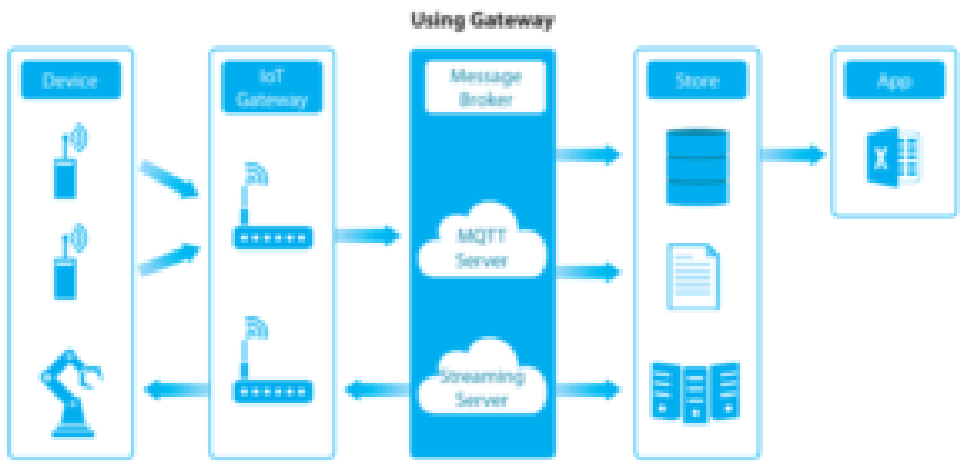
4.4 TECHNICAL BUILDING BLOCKS OF IOT

The Internet of Things denotes the connection of devices, machines, and sensors to the Internet. An IoT system comprises four basic building blocks: sensors, processors, gateways, and applications. This article will thoroughly discuss what each component of the IoT architecture represents.



The architecture of IoT components:

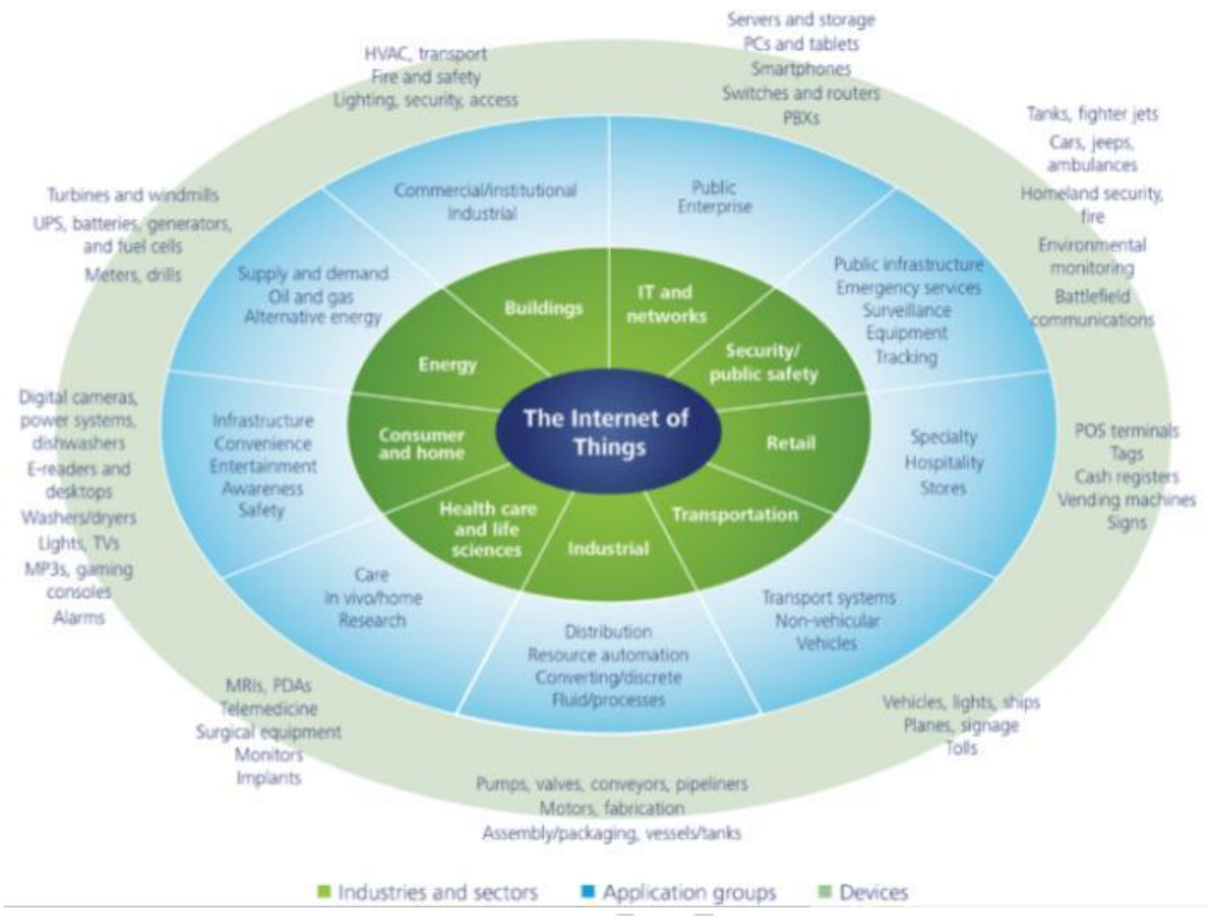
1. **Sensors** convert a non-electrical input to an electrical signal. Sensors are classified into two types: active and passive sensors. Whereas active sensors use and emit their own energy to collect real-time data (ex.: GPS, X-ray, radars), passive sensors use energy from external sources (ex: cameras). Additionally, sensors differentiate themselves by position, occupancy, and motion, velocity and acceleration, force, pressure, flow, humidity, light, radiation, temperature, etc.
2. **Processors** are the brain, the main part of the IoT system. They process the raw data captured by the sensors and extract valuable information. Examples of processors are microcontrollers and microcomputers.
3. **Gateways** are the combination of hardware and software used to connect one network to another. Gateways are responsible for bridging sensor nodes with the external Internet or World Wide Web. The figure below depicts how using gateways works.



Based on https://medium.com/enterprise-strategist/iot-architecture-basics-guide-ff4bcf8e6859

4. **Applications** provide a user interface and effective utilization of the data collected.

The figure above illustrates some examples of IoT applications.


The Internet of Things
Buildings
IT and networks
Security/ public safety
Retail
Transportation
Industrial
Health care and life sciences
Consumer and home
Energy
Commercial/institutional Industrial
Public Enterprise
Public infrastructure Emergency services Surveillance Equipment Tracking
Specialty Hospitality Stores
Transport systems Non-vehicular Vehicles
Distribution Resource automation Converting/discrete Fluid/processes
Care In vivo/home Research
Infrastructure Convenience Entertainment Awareness Safety
Supply and demand Oil and gas Alternative energy
HVAC, transport Fire and safety Lighting, security, access
Servers and storage PCs and tablets Smartphones Switches and routers PBXs
Tanks, fighter jets Cars, jeeps, ambulances Homeland security, fire Environmental monitoring Battlefield communications
POS terminals Tags Cash registers Vending machines Signs
Vehicles, lights, ships Planes, signage Tolls
Pumps, valves, conveyors, pipeliners Motors, fabrication Assembly/packaging, vessels/tanks
MRIs, PDAs Telemedicine Surgical equipment Monitors Implants
Digital cameras, power systems, dishwashers E-readers and desktops Washers/dryers Lights, TVs MP3s, gaming consoles Alarms
Turbines and windmills UPS, batteries, generators, and fuel cells Meters, drills
Industries and sectors
Application groups
Devices

4.5 COMMUNICATION TECHNOLOGIES

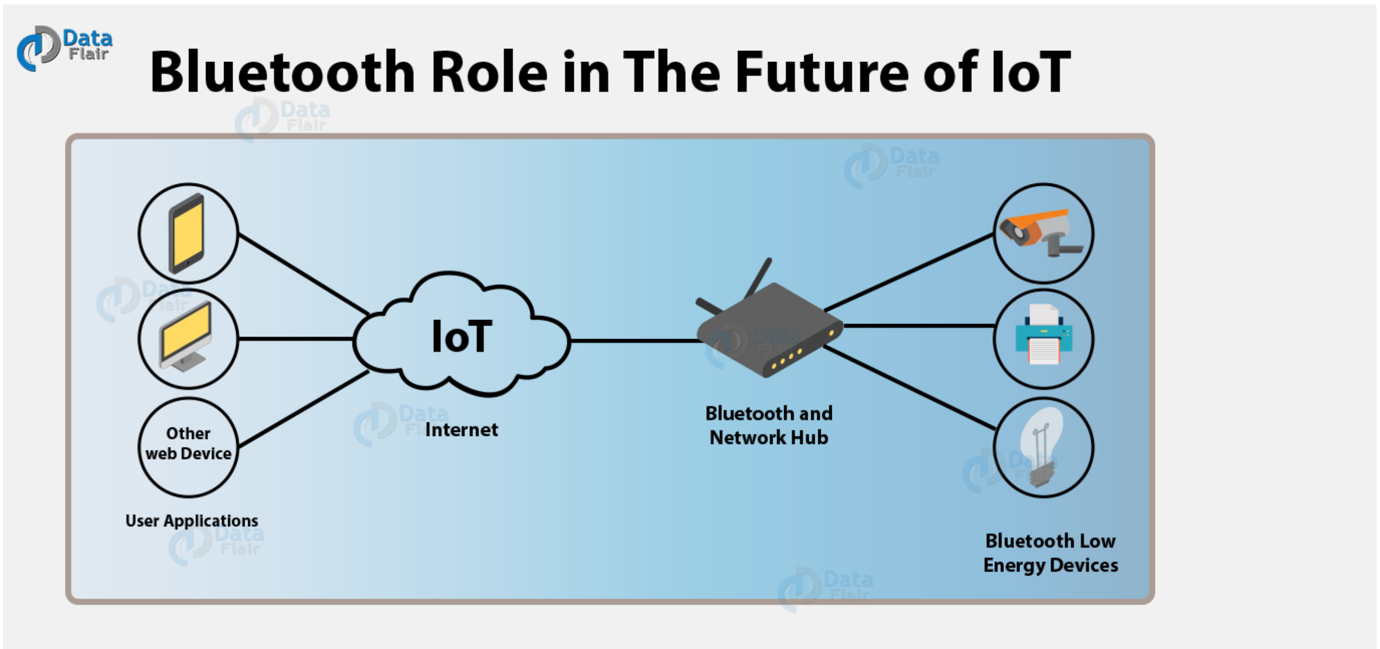
Several Communication Protocols and Technology used in the internet of Things. Some of the major IoT technology and protocol (IoT Communication Protocols) are Bluetooth, Wifi, Radio Protocols, LTE-A, and WiFi-Direct.

These IoT communication protocols cater to and meet the specific functional requirement of an IoT system. There are 6 IoT Communication Protocols/ Technology, let us ook each one of them.

a. Bluetooth

An important short-range IoT communications Protocols / Technology. Bluetooth, which has become very important in computing and many consumer product markets. It is expected to be key for wearable products in particular, again connecting to the IoT albeit probably via a smartphone in many cases.

The new Bluetooth Low-Energy (BLE) – or Bluetooth Smart, as it is now branded – is a significant protocol for IoT applications. Importantly, while it offers a similar range to Bluetooth it has been designed to offer significantly reduced power consumption.

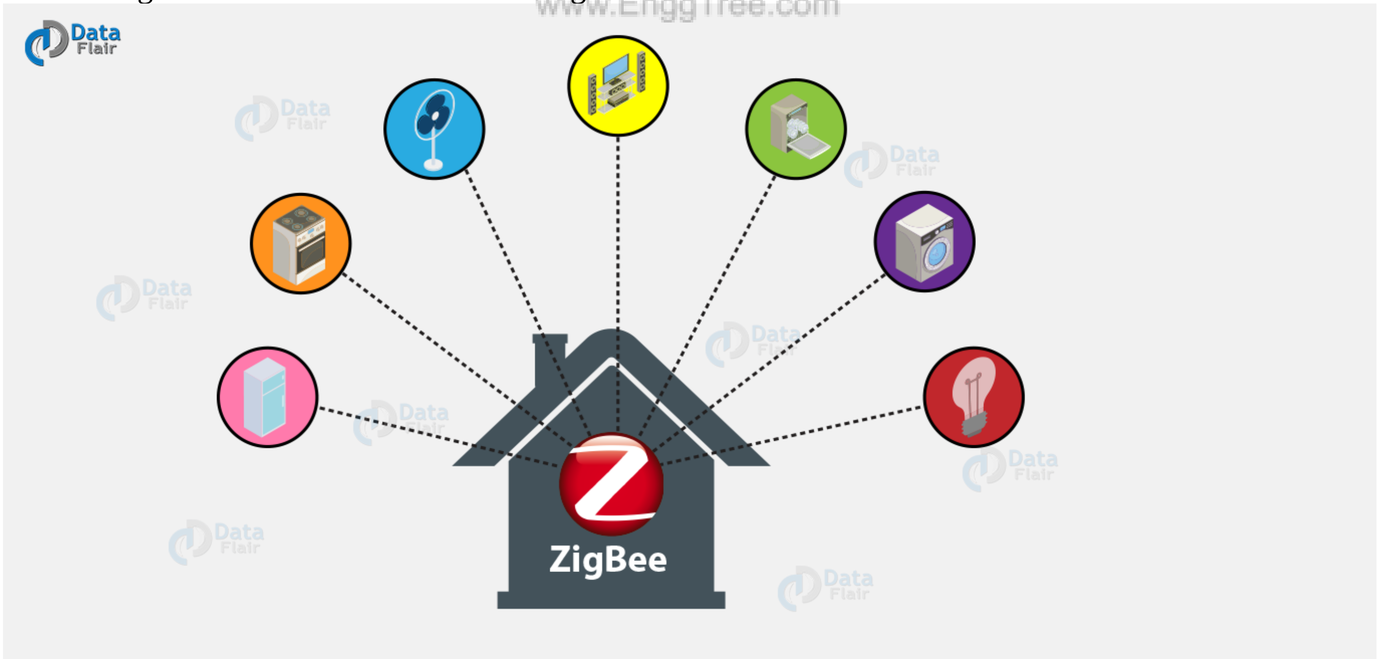

Bluetooth Role in The Future of IoT
IoT
Internet
Other web Device
User Applications
Bluetooth and Network Hub
Bluetooth Low Energy Devices

Iot Technology – Bluetooth

b. Zigbee

ZigBee is similar to Bluetooth and is majorly used in industrial settings. It has some significant advantages in complex systems offering low-power operation, high security, robustness and high and is well positioned to take advantage of wireless control and sensor networks in **IoT applications**.

The latest version of ZigBee is the recently launched 3.0, which is essentially the unification of the various ZigBee wireless standards into a single standard.



Iot Technology – ZigBee

c. Z-Wave

Z-Wave is a low-power RF communications IoT technology that primarily design for home automation for products such as lamp controllers and sensors among many other devices.
A Z-Wave uses a simpler protocol than some others, which can enable faster and simpler development, but the only maker of chips is Sigma Designs compared to multiple sources for other wireless technologies such as ZigBee and others.

Iot Technology – Z-Wave

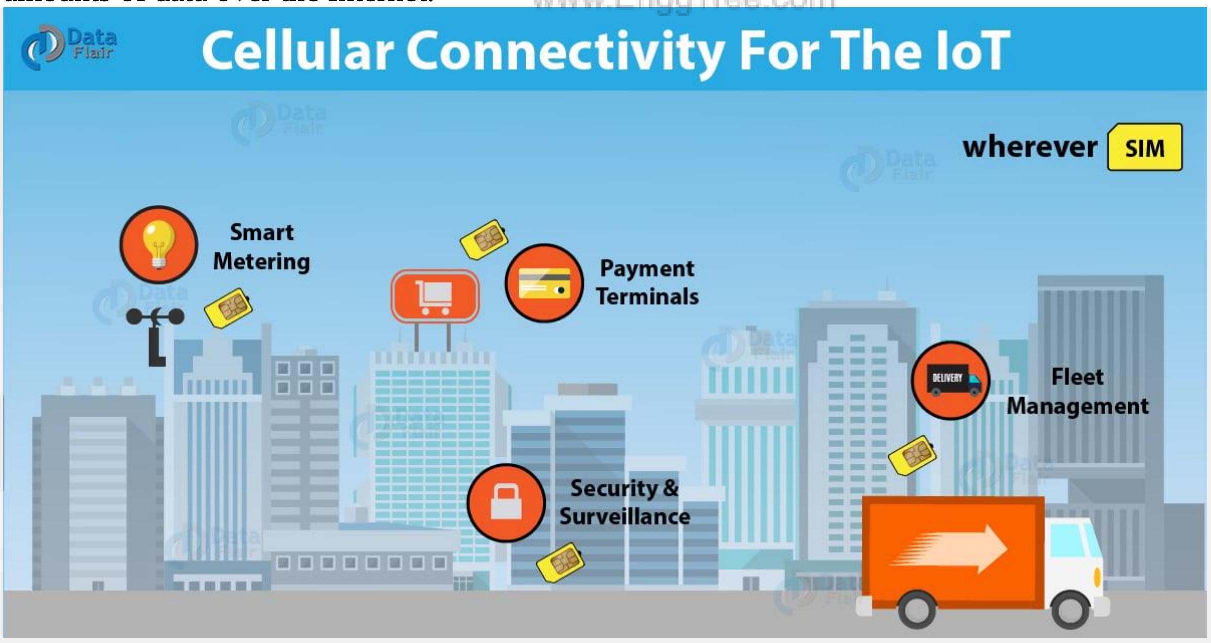
d. Wi-Fi

WiFi connectivity is one of the most popular IoT communication protocol, often an obvious choice for many developers, especially given the availability of WiFi within the home environment within LANs.There is a wide existing infrastructure as well as offering fast data transfer and the ability to handle high quantities of data.

Currently, the most common WiFi standard used in homes and many businesses is 802.11n, which offers range of hundreds of megabit per second, which is fine for file transfers but may be too power-consuming for many IoT applications.

e. Cellular

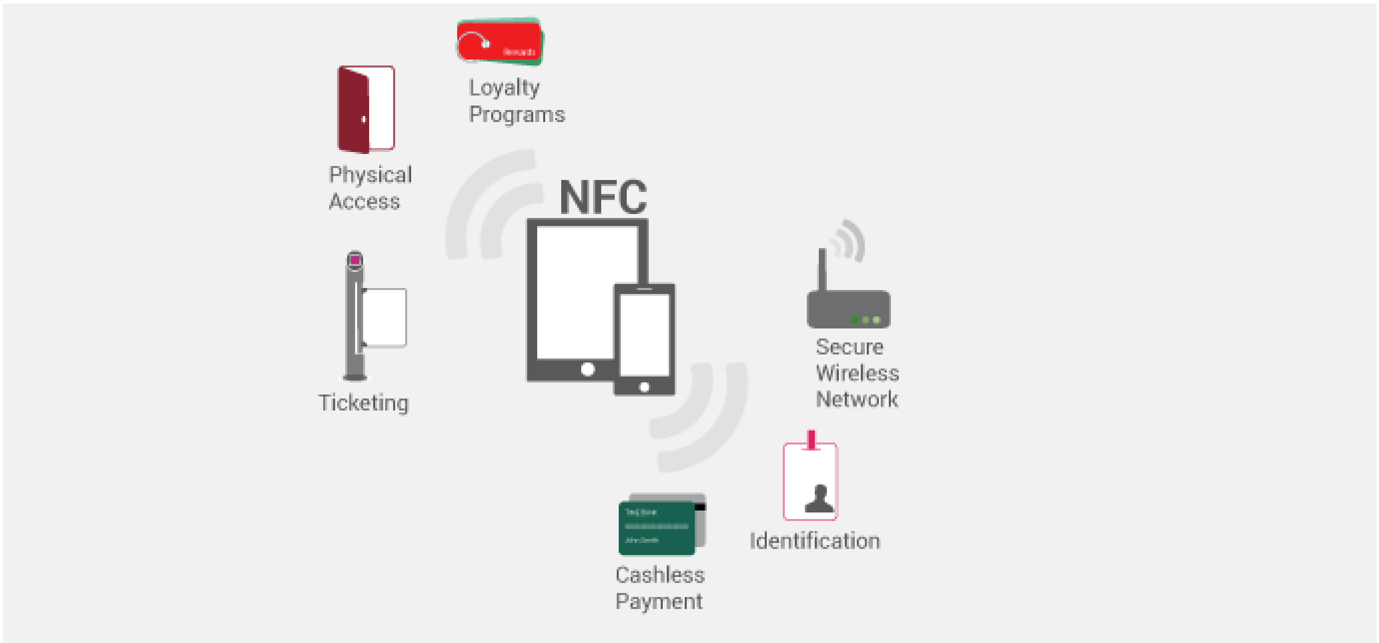
Any IoT application that requires operation over longer distances can take advantage of GSM/3G/4G cellular communication capabilities. While cellular is clearly capable of sending high quantities of data, especially for 4G, the cost and also power consumption will be too high for many applications.But it can be ideal for sensor-based low-bandwidth-data projects that will send very low amounts of data over the Internet.



IoT Communication Protocols – Cellular

f. NFC

NFC (Near Field Communication) is an IoT technology. It enables simple and safe communications between electronic devices, and specifically for smartphones, allowing consumers to perform transactions in which one does not have to be physically present.It helps the user to access digital content and connect electronic devices. Essentially it extends the capability of contactless card technology and enables devices to share information at a distance that is less than 4cm.



IoT Communication Protocols – NFC

g. LoRaWAN

LoRaWAN is one of popular IoT Technology, targets wide-area network (WAN) applications. The LoRaWAN design to provide low-power WANs with features specifically needed to support low-cost mobile secure communication in IoT, smart city, and industrial applications.Specifically meets requirements for low-power consumption and supports large networks with millions and millions of devices, data rates range from 0.3 kbps to 50 kbps.

Iot Technology – LoRaWAN

So, this was all about IoT Technology Tutorial. Hope you like our explanation of IoT Communication Protocols.

4.6 PHYSICAL DESIGN OF AN IOT

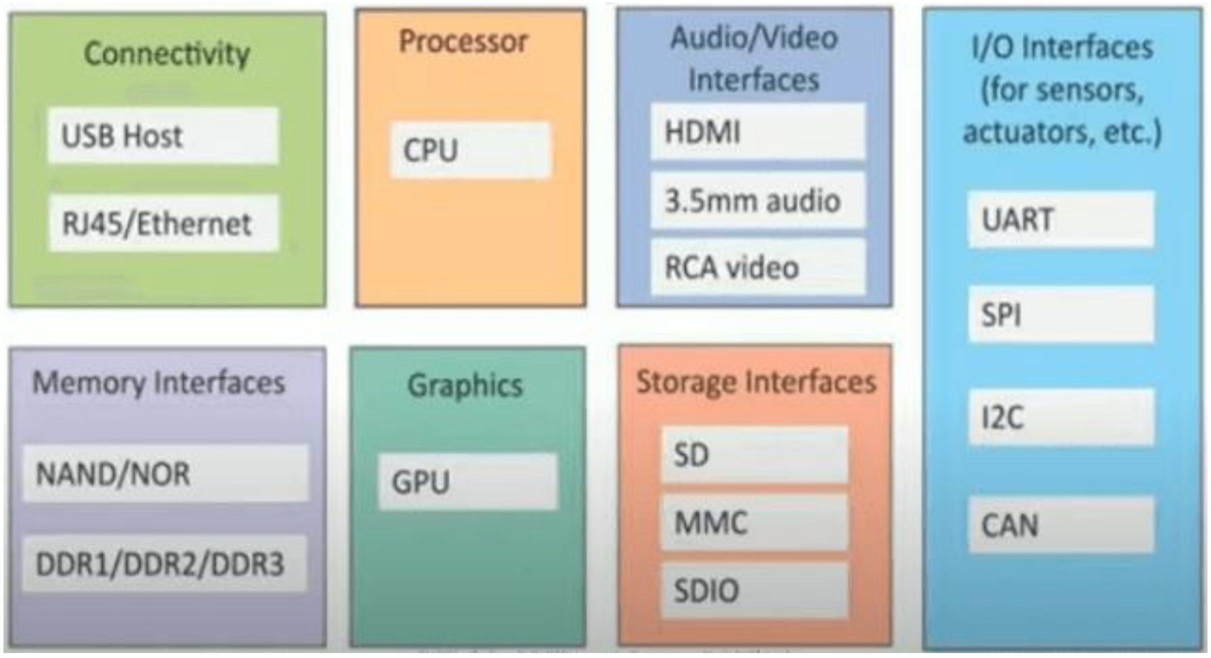
The **physical design** of an **IoT system is referred to as the Things/Devices and protocols that are used to build an IoT system. All these things/Devices are called Node Devices and every device has a unique identity that performs remote sensing, actuating,** and monitoring work. and the protocols that are used to establish communication between the Node devices and servers over the internet.

Physical Design of IoT

Things/Devices

Things/Devices are used to build a connection, process data, provide interfaces, provide storage, and provide graphics interfaces in an IoT system. All these generate data in a form that can be analyzed by an analytical system and program to perform operations and used to improve the system.

for example **temperature sensor that is used to analyze the temperature generates the data from a location and is then determined by algorithms.**



devices in IoT(Internet of things)

Connectivity

Devices like USB hosts and ETHERNET are used for connectivity between the devices and the server.

Processor

A processor like a CPU and other units are used to process the data. these data are further used to improve the decision quality of an IoT system.

Audio/Video Interfaces

An interface like HDMI and RCA devices is used to record audio and videos in a system.

Input/Output interface

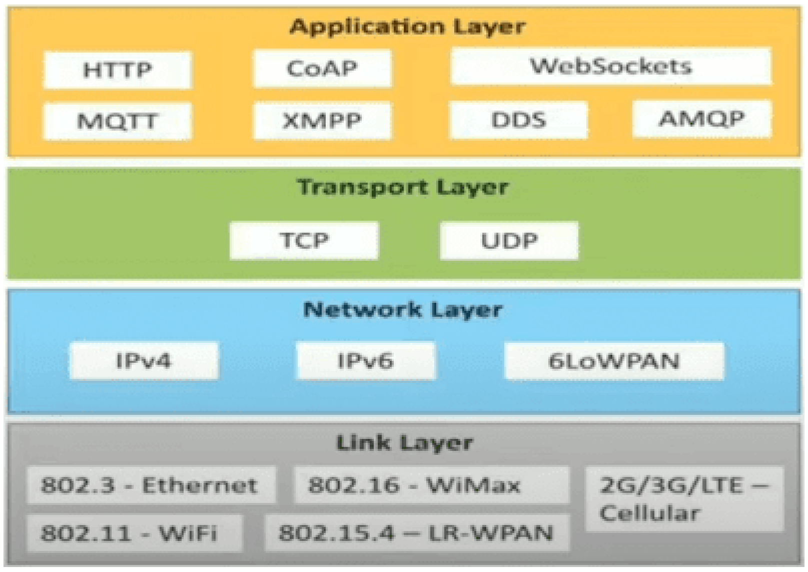
To give input and output signals to sensors, and actuators we use things like UART, SPI, CAN, etc.

Storage Interfaces

Things like SD, MMC, and SDIO are used to store the data generated from an IoT device.Other things like DDR and GPU are used to control the activity of an IoT system.

IoT Protocols

These protocols are used to establish communication between a node device and a server over the internet. it helps to send commands to an IoT device and receive data from an IoT device over the internet. we use different types of protocols that are present on both the server and client side and these protocols are managed by network layers like application, transport, network, and link layer.



IoT(Internet of Things) protocols

Application Layer protocol

In this layer, protocols define how the data can be sent over the network with the lower layer protocols using the application interface. these protocols include HTTP, WebSocket, XMPP, MQTT, DDS, and AMQP protocols.

HTTP

Hypertext transfer protocol is a protocol that presents an application layer for transmitting media documents. it is used to communicate between web browsers and servers. it makes a request to a server and then waits till it receives a response and in between the request server does not keep any data between the two requests.

WebSocket

This protocol enables two-way communication between a client and a host that can be run on an untrusted code in a controlled environment. This protocol is commonly used by web browsers.

MQTT

It is a machine-to-machine connectivity protocol that was designed as a publish/subscribe messaging transport. and it is used for remote locations where a small code footprint is required.

Transport Layer

This layer is used to control the flow of data segments and handle error control. also, these layer protocols provide end-to-end message transfer capability independent of the underlying network.

TCP

The transmission control protocol is a protocol that defines how to establish and maintain a network that can exchange data in a proper manner using the internet protocol.

UDP

a user datagram protocol is part of an internet protocol called the connectionless protocol. this protocol is not required to establish the connection to transfer data.

Network Layer

This layer is used to send datagrams from the source network to the destination network. we use IPv4 and IPv6 protocols as host identification that transfers data in packets.

IPv4

This is a protocol address that is a unique and numerical label assigned to each device connected to the network. an IP address performs two main functions host and location addressing. IPv4 is an IP address that is 32-bit long.

IPv6

It is a successor of IPv4 that uses 128 bits for an IP address. it is developed by the IETF task force to deal with long-anticipated problems.

Link Layer

Link-layer protocols are used to send data over the network's physical layer. it also determines how the packets are coded and signaled by the devices.

Ethernet

It is a set of technologies and protocols that are used primarily in LANs. it defines the physical layer and the medium access control for wired ethernet networks.

WiFi

It is a set of LAN protocols and specifies the set of media access control and physical layer protocols for implementing wireless local area networks.

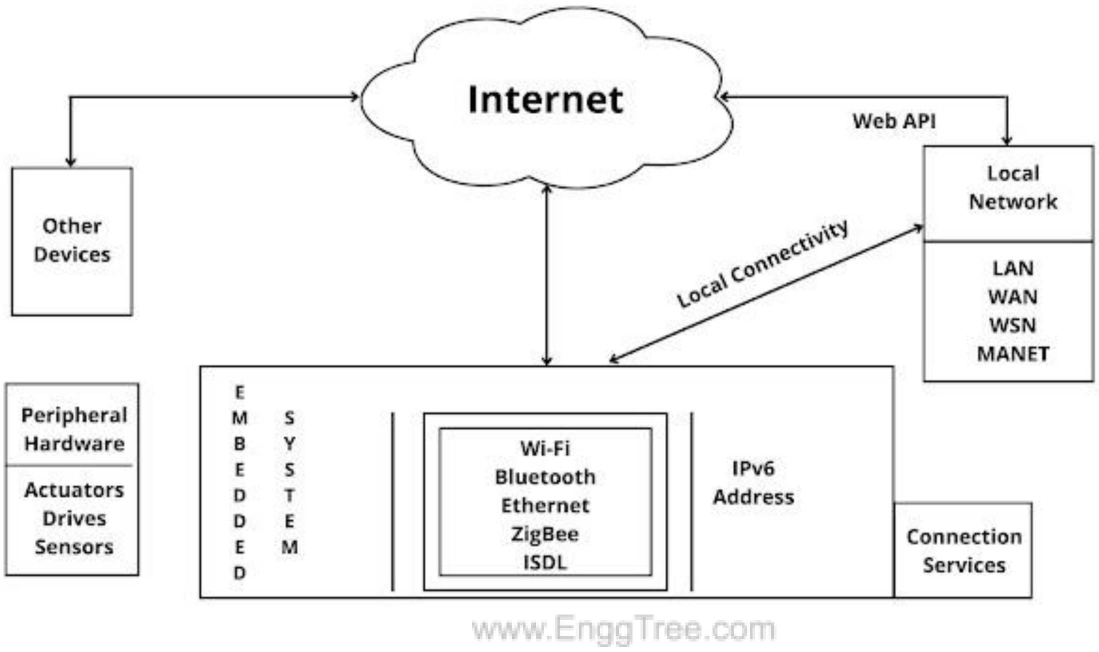
4.7 IOT SOFTWARE BUILDING BLOCKS

Software is the set of programs that facilitates the data collection, processing, storage, and evaluation instructions based on the processed data from the IoT Software. Operating Systems, firmware, applications, and middleware are some of the examples that fall into this category.

1. **Data Collection:** Data collection includes a core of the data collection aspects ranging from sensing the data, filtering it, measuring it, aggregating it, and in the end managing 2/ the security of the collected data. It can be performed from various sources, and once done is distributed over devices and then to a central data repository.
2. **Device Integration:** In this category, all components within the IoT system are integrated. It manages all the limitations, protocols, and applications are handled properly to ensure proper communication among the devices.

3. **Real-Time Analytics:** Over the collected data and the processing that is done over this data, there can be automated tasks that could run and analyze this data for specific patterns.
4. **Application and Process Extension:** This ensures that the data collection process can be accentuated to get the most out of it, from all possible sources, These are more like the enhancers over the existing data collection infrastructure.

We can understand this through the example. The basic architecture of IoT is shown in the diagram.



Building blocks of iot(Internet of things)

4.8 SENSORS AND SENSOR NODE AND INTERFACING USING ANY EMBEDDED TARGET BOARDS

(RASPBERRY PI / INTEL GALILEO/ARM CORTEX/ ARDUINO)

Industrial IoT requires a robust infrastructure comprised of sensor nodes, energy-efficient communication networks, and gateways that connect to the Internet and cloud. The datasets generated by the sensors pass through many phases before becoming valuable for businesses. This article introduces the concept of sensor nodes, low-power networks, and IoT Gateways used in industrial scenarios. We will use Arduino, XBee, Raspberry Pi 2 and open source software for prototyping an end-to-end solution.

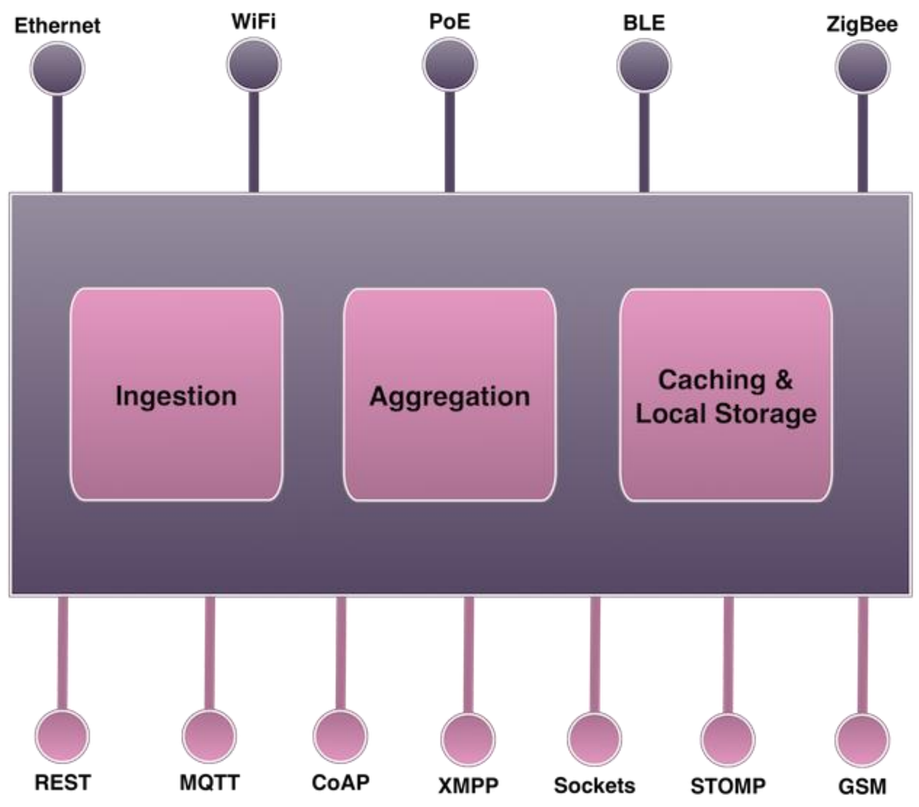
The latest Raspberry Pi 2 packs a lot more punch than its predecessor. Powered by a 900 MHz quad-core ARM Cortex-A7 processor and 1GB RAM, Raspberry Pi 2 Model B can be used for a variety of scenarios. We will use an open source software stack to aggregate, store and visualize sensor data in real-time.

What is an IoT Gateway?

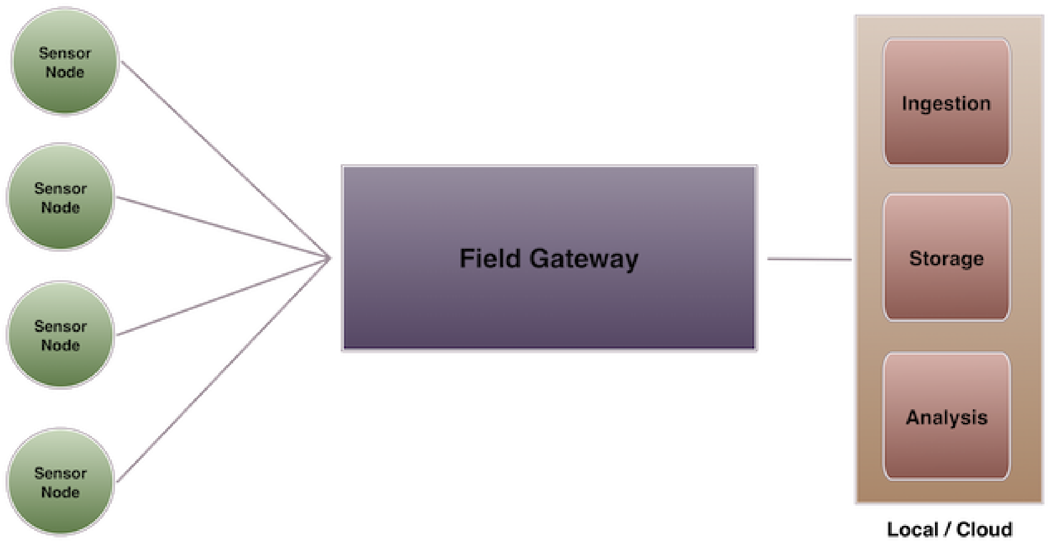
In an industrial IoT scenario, there are many sensors and actuators that interact with the machinery. Each machine would typically have multiple sensors tracking its health and monitoring the key parameters related to the production. Each sensor and actuator is attached to a microcontroller that is responsible for acquiring the data or controlling a switch through a pre-defined instruction set. The microcontroller — along with the sensors, power and a radio — is called a sensor node. It is a self-contained, deployable unit that captures the data generated by sensors. The sensor node doesn't have enough processing power, memory, and storage to deal with the data locally. It uses a low-energy radio communication network to send the data to a central location. The communication link between the sensors' nodes and the central hub is based on ZigBee, Bluetooth Low Energy (BLE), or Power over Ethernet (PoE). The hub that acts as an aggregator of multiple raw datasets generated by the sensor nodes is called an IoT gateway.

The second role of an IoT gateway is protocol transformation. Since the sensor nodes cannot use power hungry Wi-Fi or ethernet, they use low-powered communication networks. A gateway supports multiple communication protocols for accepting the inbound data sent by the sensor nodes. It uses a variety of protocols for the outbound communication, which typically connects the gateway to a process running in the cloud. Some of the popular outbound protocols used in the context of IoT are REST, MQTT, CoAP, STOMP and even SMS. In some scenarios, the gateway may also process the data and raise alerts in real time. But this is best left to the powerful stream processing pipelines running in the cloud.

The diagram below represents the deployment architecture of an IoT gateway.



Gateways act as an edge device, obscuring the sensor nodes from the public internet. Though the sensor nodes can make outbound connections to the internet and cloud through the gateway, they cannot be accessed directly. Thus, gateways play the dual role of routers and firewall securing the sensor nodes and internal network.Sensor nodes that are capable of connecting to the Internet still need a gateway for data aggregation and transformation. They connect to an appliance running in the cloud called a cloud gateway. The local edge device running on-premises is often referred to as a field gateway.



Arduino as a the Sensor Node and Raspberry Pi as the Gateway

Since the objective of this tutorial is to build a working prototype, we will keep the sensor node configuration simple. We will use an inexpensive DHT22 sensor that captures the ambient humidity and temperature and sends it to the gateway.

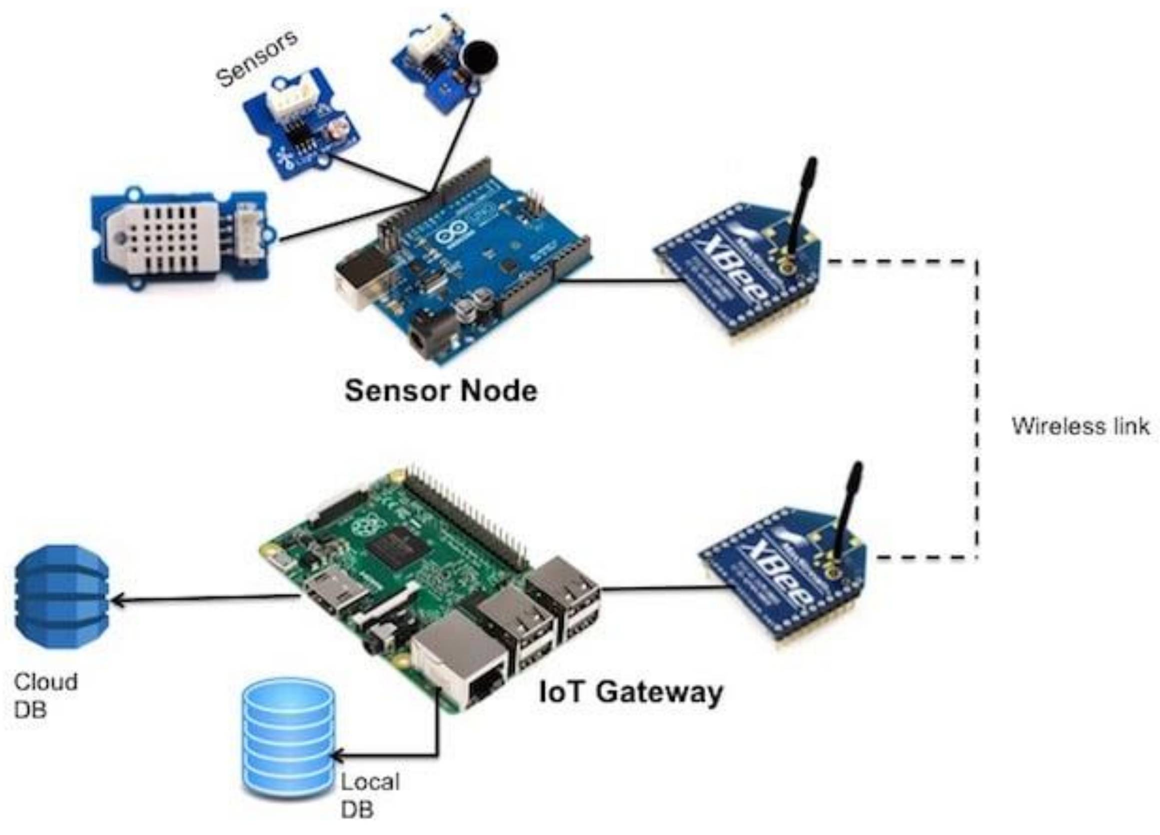
We use the following components:

Sensor Node

- Arduino Uno R3
- DHT 22 humidity / temperature sensor
- XBee breakout board
- XBee series 2

IoT Gateway

- Raspberry Pi 2
- XBee series 2
- Wi-Fi Dongle



A Short Introduction to XBee

XBee modules make it easy to create a wireless point-to-point or mesh network. They are configured with the standard AT commands. With built-in error correction, XBee modules offer a reliable wireless link. They come in multiple flavors, with support for protocols like ZigBee, Bluetooth, and even Wi-Fi. The XBee modules can be configured to operate either in a transparent data mode or in application programming interface (API) mode.

In the transparent mode, data coming into the Data IN (DIN) pin is directly transmitted over-the-air to the intended receiving radios without any modification. This feature is what makes XBee a drop-in replacement for an RS–232 cable. Incoming packets can either be directly addressed to one target (point-to-point) or broadcast to multiple targets (star). We will use XBees configured in transparent mode for connecting the sensor node to the gateway.

ZigBee is a specification for a high-level communication protocol mainly used for personal area networking based on small, low-power digital radios. ZigBee is based on an IEEE 802.15.4 standard. Its low power consumption confines the transmission range from 10 to 100 meters line-of-sight, depending on power output and environmental characteristics. ZigBee devices are capable of transmitting data over long distances by passing through a mesh network of intermediate devices to reach more distant ones.

ZigBee is typically used in low data rate applications that require long battery life and secure networking. ZigBee networks are secured by 128-bit symmetric encryption keys. Some of the use cases of ZigBee include wireless light switches, electrical meters with in-home-displays, traffic management systems, and other consumer and industrial equipment that requires short-range, low-rate wireless data transfer.

The technology defined by the ZigBee specification is intended to be simpler and less expensive than other wireless personal area networks (WPANs), such as Bluetooth or Wi-Fi. Philips Hue bulbs use ZigBee as the communication protocol between the hub and the lightbulbs. We will use ZigBee as the radio link between the sensor node and the IoT gateway.

The XBee radio connected to the sensor node will act as a router, while the one connected to the gateway will act as the coordinator. Multiple routers can talk to the coordinator. For advanced scenarios, it is recommended to use XBee in API configuration mode. For a detailed walkthrough of configuring the XBee radio modules in the router mode and coordinator mode, please refer to an excellent tutorial by SparkFun.

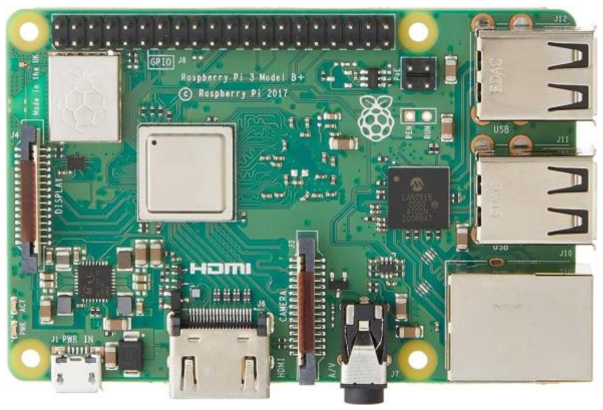
EMBEDDED TARGET BOARDS

Open Source Embedded Development Boards

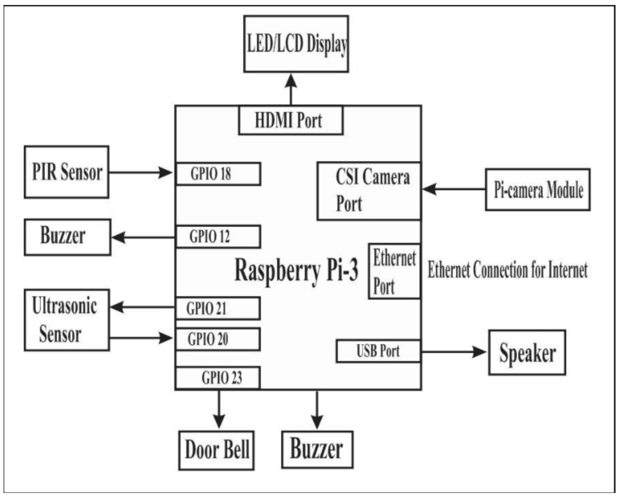
Returning to the main focus of this column, there are dozens of development boards available in the market; ten of the more popular offerings are as follows:

1.Raspberry Pi 3 B+

The Raspberry Pi development board is a small pocket-sized computer running the Raspbian operating system, which is a variant of Debian Linux



Raspberry Pi features a Broadcom processor. It is a low-cost embedded board with high reliability. It supports various on-board peripherals like I2C, SPI, an HDMI interface, a Camera interface, a UART interface, and SDIO (Secure digital input output) for a SD card interface.



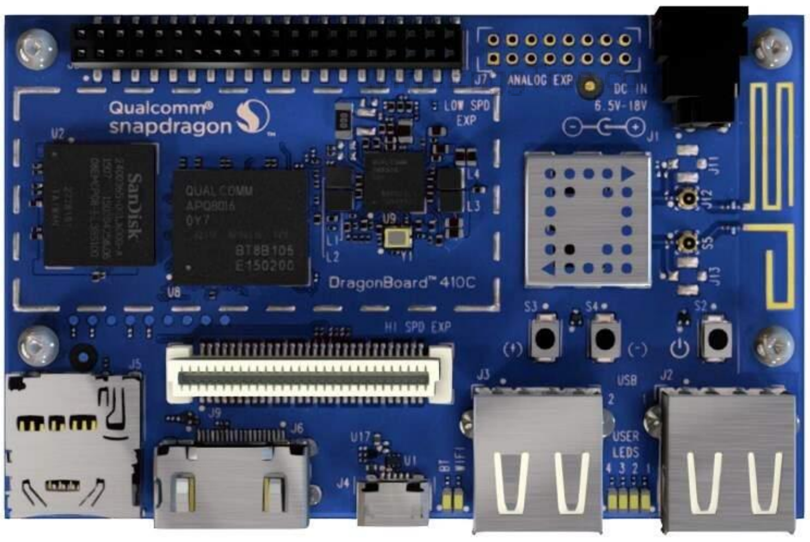
Block Diagram

The board comes up with software APIs and routines for application programming. It also has OTG (On the Go programming) for USB applications.

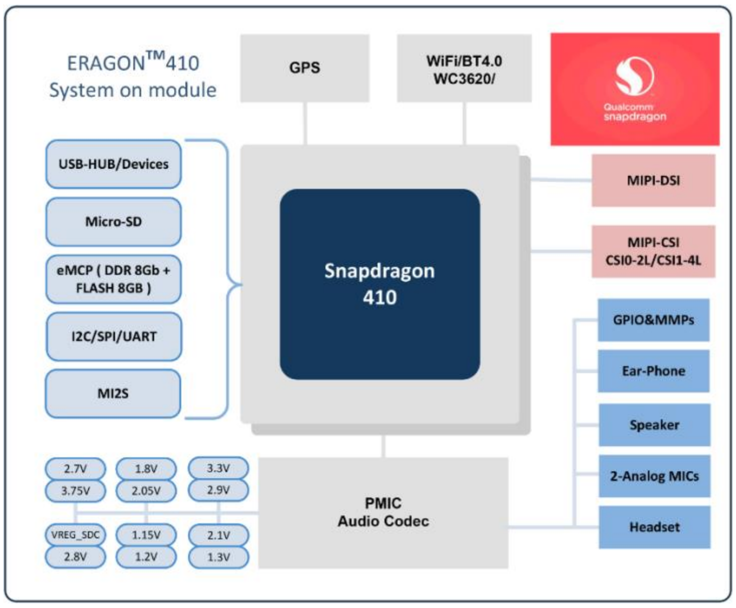
There are various raspberry pi models. The latest one being the Raspberry Pi 3 B+.

2.Qualcomm Snapdragon

This is single board computer (SBC) that uses the powerful Snapdragon processor from Qualcomm. It supports various interfaces like Wi-Fi, Bluetooth, and Global Positioning System (GPS). It is well suited for Internet of Things (Iot), medical, and robotic applications.



You can connect a keyboard, mouse, and USB interfaces. As you may know, some smartphones are coming equipped with this processor.

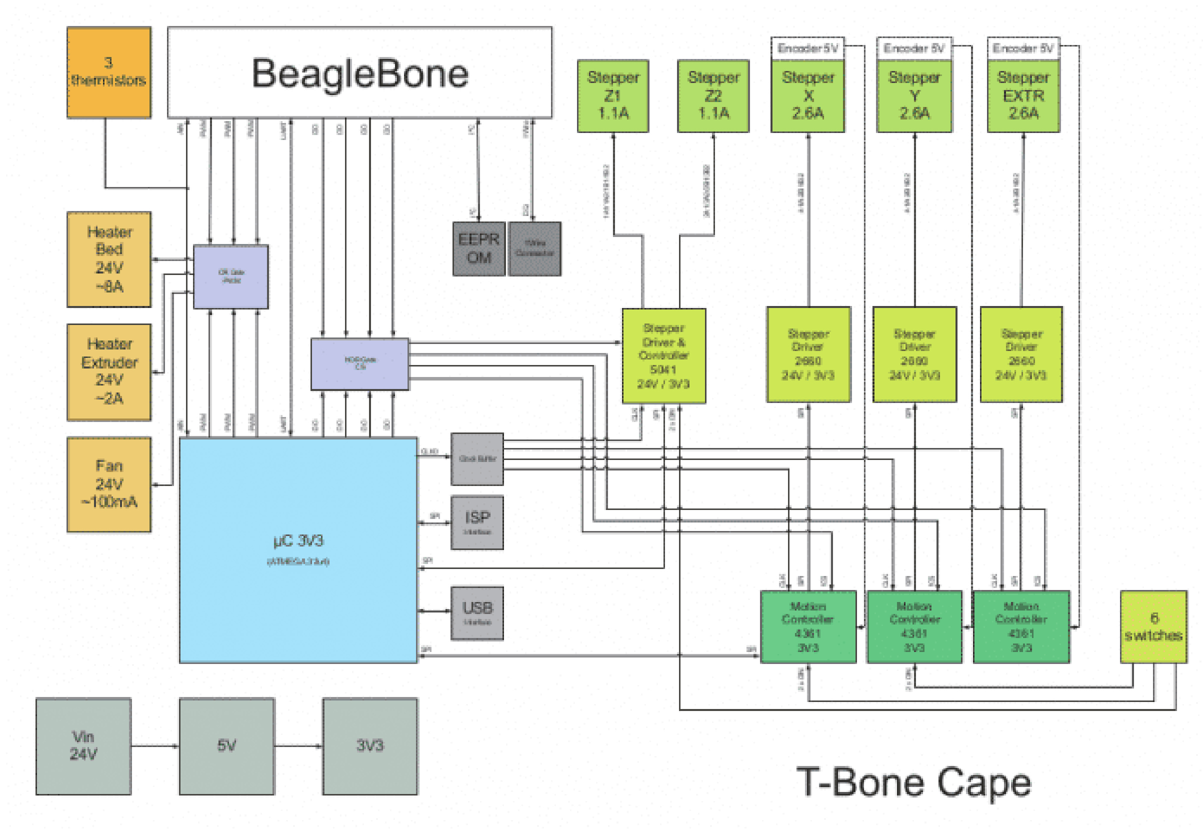


Block Diagram

3.BeagleBone Black

The BeagleBone Black uses a Texas Instruments (TI) Sitara processor running on ARM Cortex-A8 core. The system clock frequency is 1GHz, which supports various operating systems like Windows, Linux, Android, QNX, Embedded CE, and ThreadX.

This development board supports an Ethernet interface, a micro SD card interface, a UART, a Flash interface, and an HDMI interface for audio and video. The weight of the system is around 40 grams, making portable and easy to carry.

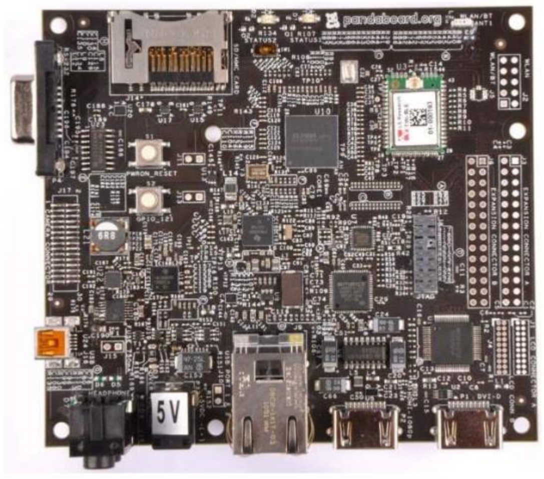


Block Diagram

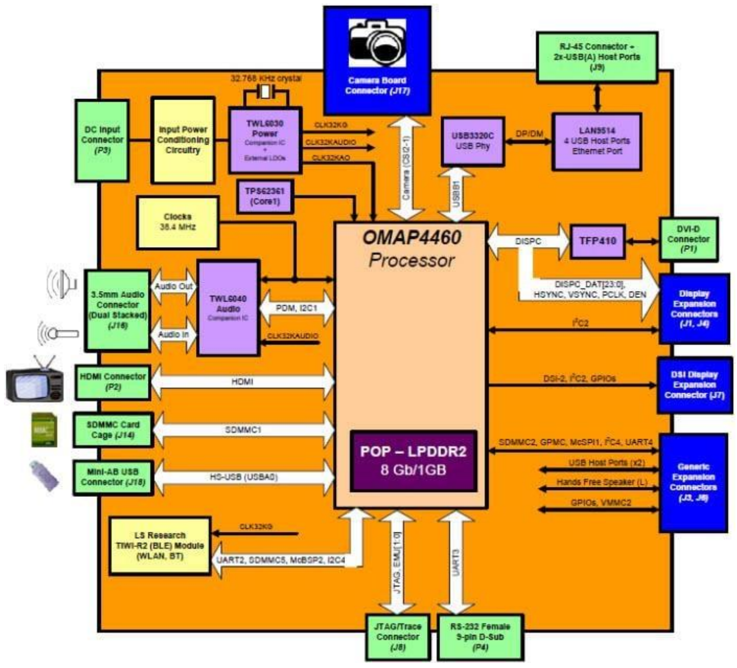
There's a USB option for downloading the stored data from the flash and also for updating the firmware.

4.PandaBoard

The PandaBoard is a low-power, low-cost development board based on TI's OMAP4460 (Open media application platform). This board supports operating systems like Windows, Linux, Window CE, Palm OS, and Symbian.



Boasting a Cortex-M3 processor running at 1.2GHz clock frequency, this board is well suited for image processing applications. The processor is accompanied by a 384MHz GPU (graphics processing unit). The board has two USB ports and supports Ethernet and Wireless connectivity using Bluetooth. The programming can be performed via the USB OTG port.

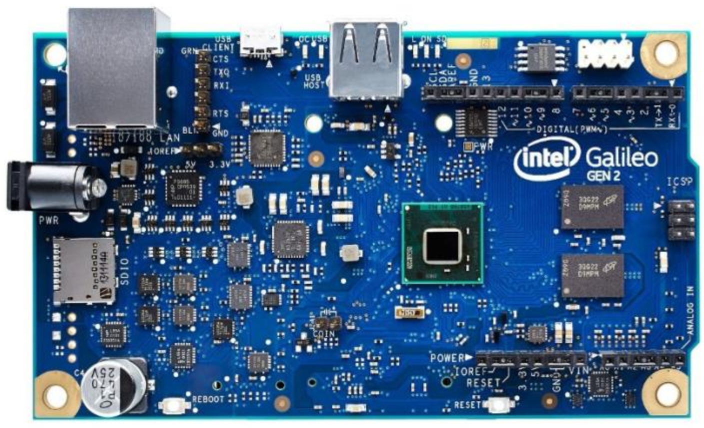

Camera Board Connector (J17)
RJ-45 Connector + 2x-USB(A) Host Ports (J9)
32.768 KHz crystal
DC Input Connector (P3)
Input Power Conditioning Circuitry
TWL6030 Power Companion IC + External LDOs
CLK32KG
CLK32KAUDIO
CLK32KAO
Camera (CSI2-1)
USB3320C USB Phy
DP/DM
LAN9514 4 USB Host Ports Ethernet Port
USBB1
TPS62361 (Core1)
Clocks 38.4 MHz
OMAP4460 Processor
DISPC
TFP410
DVI-D Connector (P1)
3.5mm Audio Connector (Dual Stacked) (J16)
Audio Out
Audio In
TWL6040 Audio Companion IC
PDM, I2C1
CLK32KAUDIO
DISPC_DAT[23:0], HSYNC, VSYNC, PCLK, DEN
Display Expansion Connectors (J1, J4)
I²C2
HDMI Connector (P2)
HDMI
DSI-2, I²C2, GPIOs
DSI Display Expansion Connector (J7)
SDMMC Card Cage (J14)
SDMMC1
SDMMC2, GPMC, McSPI1, I²C4, UART4
POP – LPDDR2 8 Gb/1GB
Mini-AB USB Connector (J18)
HS-USB (USBA0)
USB Host Ports (x2)
Hands Free Speaker (L)
GPIOs, VMMC2
Generic Expansion Connectors (J3, J6)
LS Research TIWI-R2 (BLE) Module (WLAN, BT)
CLK32KG
UART2, SDMMC5, McBSP2, I2C4
JTAG, EMU[1:0]
UART3
JTAG/Trace Connector (J8)
RS-232 Female 9-pin D-Sub (P4)

Block Diagram

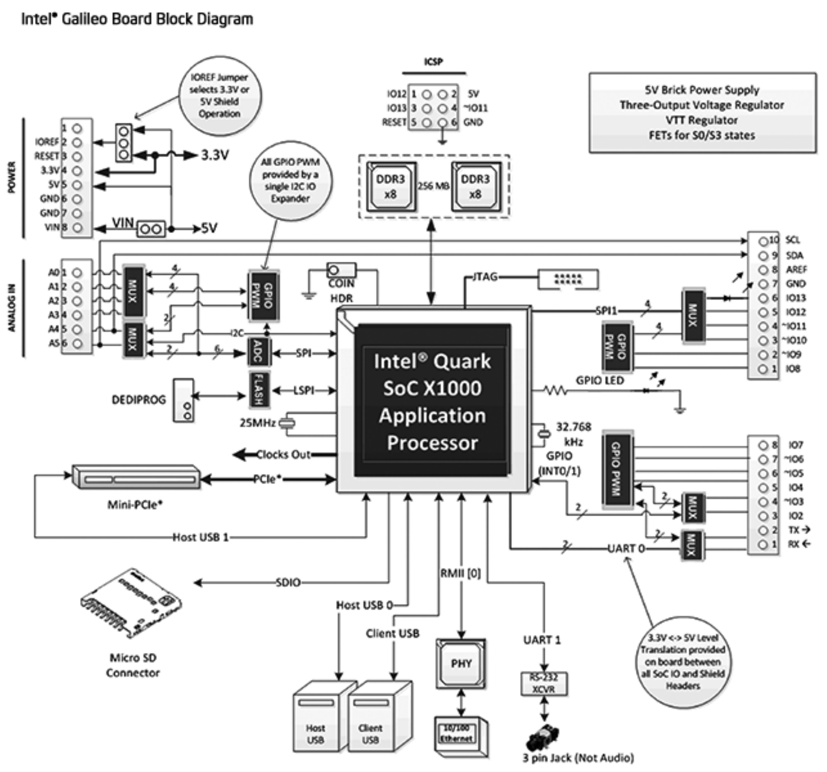
The video processing capability of the PandaBoard makes it a good for 1080 High Definition (HD) applications. This SBC is well-suited for entertainment applications; however, it is quite expensive when compared with other SBCs.

5.Intel Galileo Gen 2

The Galileo development board comes from Intel and features an Intel Quark SoC X1000 processor. It is designed using Pentium technology. The advantage of this board is it is compatible with shields for the Arduino Uno R3.



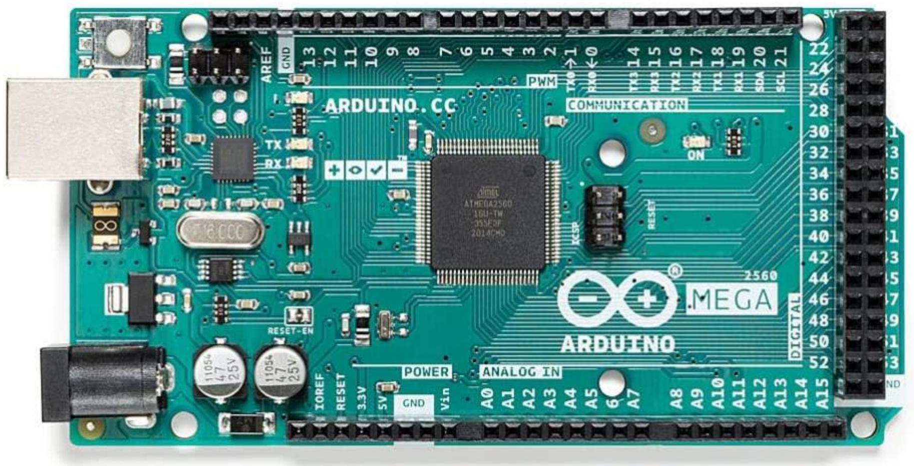
In addition to 8Mb Flash, the Galileo has various interfaces like USB, SD card, UART, and Ethernet. The Galileo also has a rich set of software libraries for developing applications. Due to its software compatibility and ease of interface, it is well suited for students and electronic hobbyists.



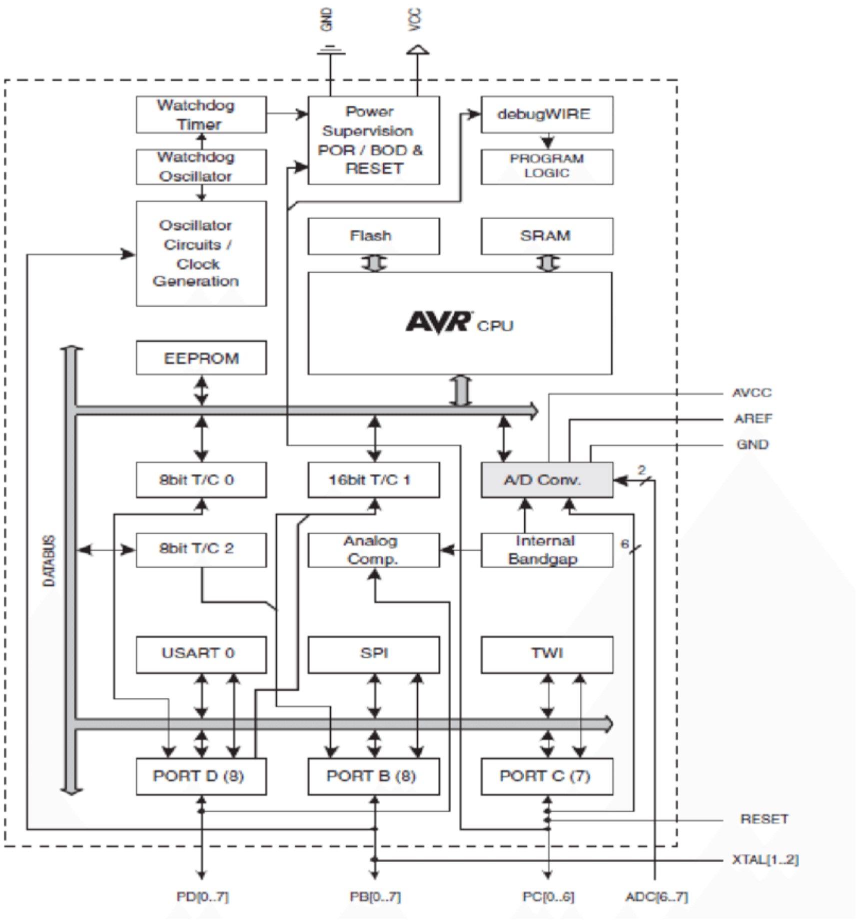
Block Diagram

6.Arduino Mega 2560

Arduino is an open source hardware and software platform family with thousands of active users and contributors. It is one of the best platforms for making electronic projects. If you are a beginner, you can quickly develop applications with less effort than with other platforms.



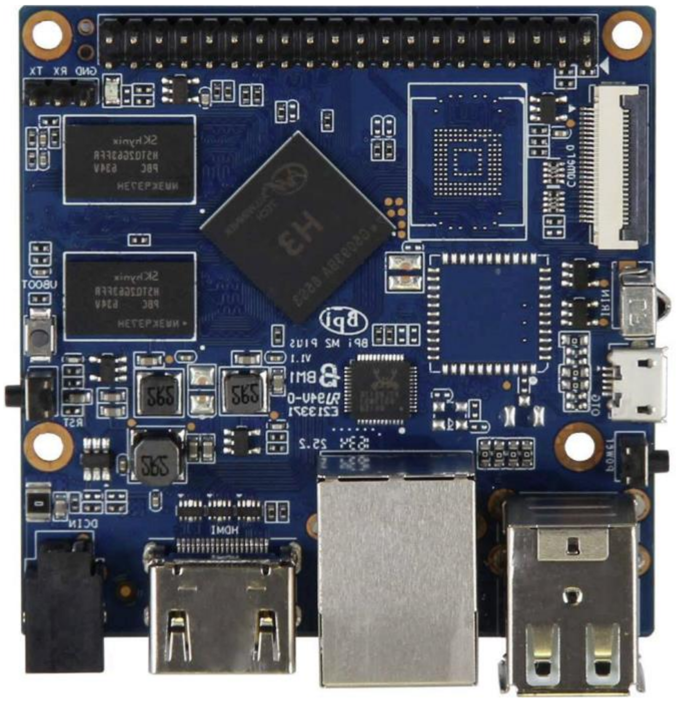
The board features an 8-bit ATmega2560 microcontroller running at 16MHz. It has 54 digital Input/output pins and 16 analog inputs. The board has four UARTs and can be programmed using the Arduino IDE. It is also compatible with other variants of Arduino shields.



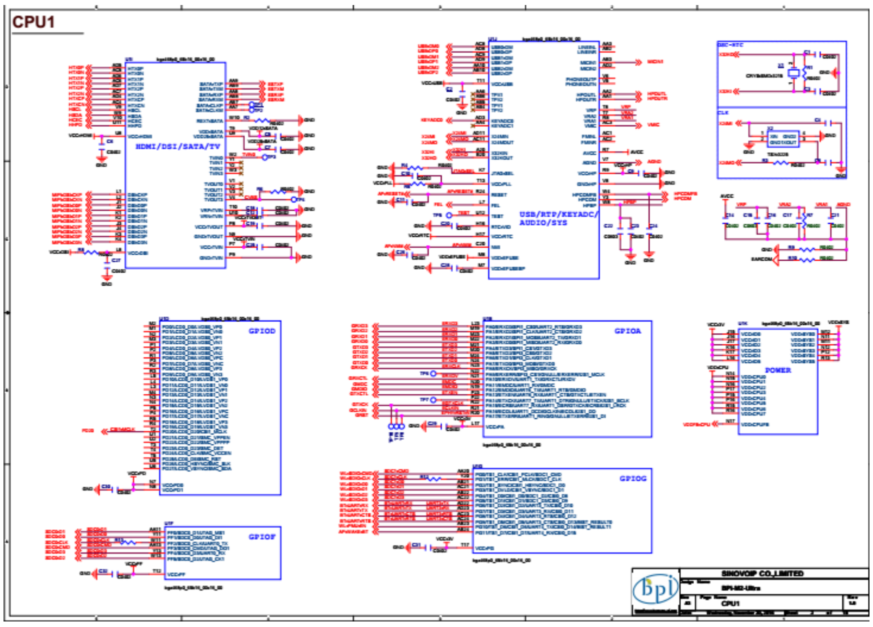
Block Diagram

7.Banana Pi M2+

The Banana Pi M2+ is a portable SBC that supports various interfaces like Bluetooth, Wi-Fi, and Ethernet. It offers great computing performance with its quad-core ARM Cortex-A7 processor running at 1.2GHz.



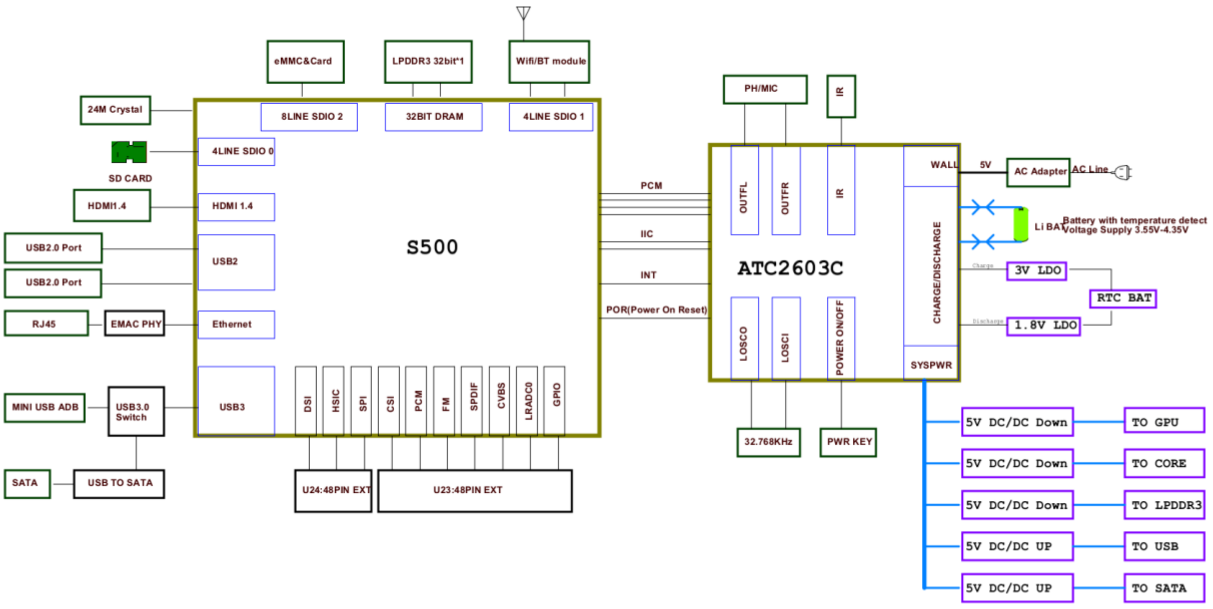
There are various versions of the Banana Pi M2+ named as H3, H2+, EDU, and H5. For example, the Banana EDU is well-suited for students and engineers to learn the functionality of small embedded applications, but it has no Wi-Fi or Bluetooth on-board.



8.CubieBoard6

CubieBoard6 is an SBC that runs operating systems like Linux. It also supports the Android OS. The Cubie is empowered with a quad core Cortex-A9 processor. The board can also be powered using a LiPo (lithium polymer) battery.

It comes with an infrared (IR) sensor and [Real Time Clock (RTC) module](). Moreover, it has Wi-Fi and Bluetooth support that enables network connection. The Cubie has an audio input and output via a 3.5mm jack or an HDMI cable.

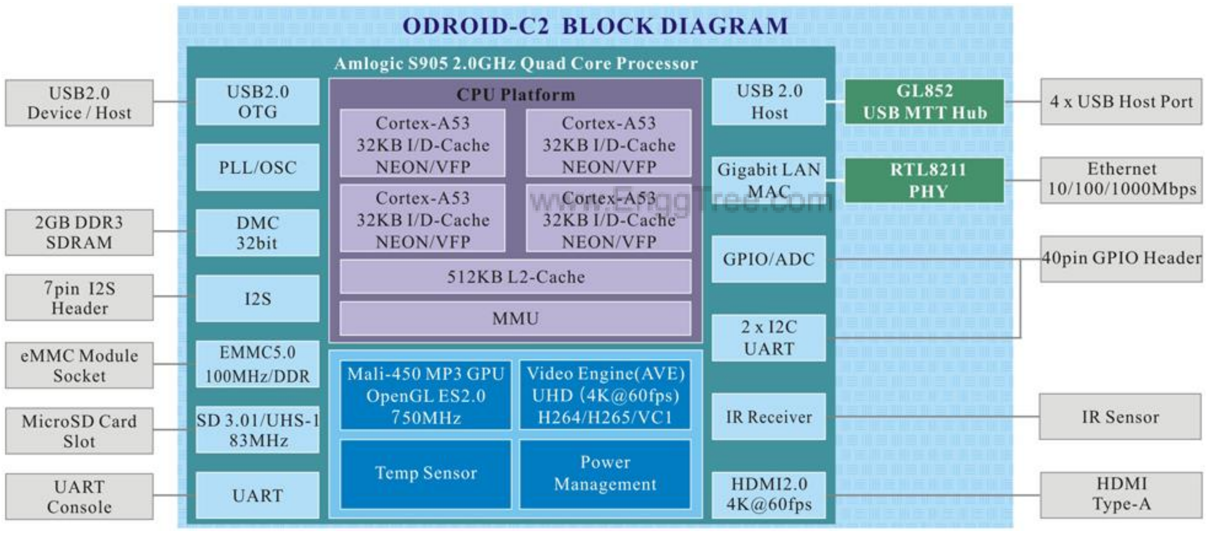


Block Diagram

9.Odroid-C2

The ODROID-C2 is a 64-bit quad-core (SBC) that is suitable for applications like multimedia, gaming, and consumer electronics. It can also work as a standalone computer with available open source software packages.

The ODROID-C2 is powered by an Amlogic S950 processor, which is based on an ARM Cortex-A53, making it a resource for wearable applications. In addition to a built-in temperature sensor, this board has four USB ports, and two UART and I2C interfaces. The ODROID-C2 also boasts 2GB 32-bit DDR3 RAM and 40 GPIO pins for connecting external peripherals.



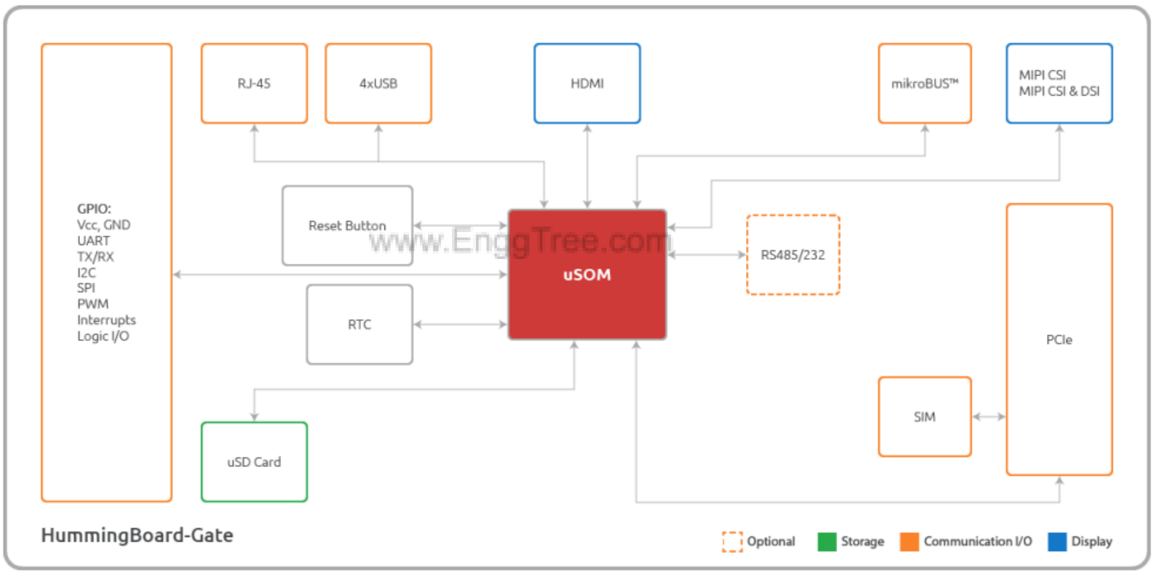
Block Diagram

10.HummingBoard Gate

The HummingBoard Gate has been described as: " The device you've been waiting for to fulfil all of your IoT needs."

Based on an NXP quad-core processor, this SBC comes with an integrated mikroBUS socket that is suitable for MikroElektronika development boards and external peripheral modules.

The HummingBoard Gate supports 2GB of DDR3 RAM and is useful for building modular projects and providing proof-



of-concept.

Block Diagram

4.10 Benefits of the Internet of Medical Things (IoMT)

1. **Improved Patient Outcomes:**
   The real-time monitoring provided by the IoMT enables healthcare providers to detect health problems early and take prompt action, leading to improved patient outcomes.

2. **Cost Reduction:**
   By monitoring patients in real time, the IoMT can reduce the need for frequent hospital visits and other medical procedures, leading to reduced costs.

3. **Increased Efficiency:**
   The IoMT can streamline healthcare operations by reducing the amount of manual data entry and enabling healthcare providers to access patient data more easily.

4. **Improved Patient Engagement:**
   The IoMT can improve patient engagement by providing patients with easy access to their health data and enabling them to track their progress over time.

5. **Better Decision Making:**
   The data collected by the IoMT can be used to make more informed decisions about patient care, leading to improved patient outcomes and reduced costs.

4.11 IMPACT OF IoMT

IoMT refers to a system of interconnected sensors and medical equipment that can gather and send real-time patient data to healthcare providers. This technology can potentially enhance patient outcomes and save healthcare costs by **enabling remote monitoring, individualised care, and preventive interventions.**

The six challenges we have to face when IoMT is implemented:

1. Cyber Security
2. Interoperability
3. Device Mobility
4. Licensing and Regulations
5. Improving Adoption Scale
6. Need for Advanced Analytics

4.12 CYBERSECURITY

Cyber security refers to every aspect of protecting an organization and its employees and assets against cyber threats. As cyberattacks become more common and sophisticated and corporate networks grow more complex, a variety of cyber security solutions are required to mitigate corporate cyber risk.

4.12.1 VULNERABILITY

Any flaw in an organization's internal controls, system procedures, or information systems is a vulnerability in cyber security. Cybercriminals and Hackers may target these vulnerabilities and exploit them through the points of vulnerability.These hackers can enter the networks without authorization and seriously harm data privacy. Data being a gold mine in this modern world is something that has to be secured preciously. As a result, it is crucial to constantly check for cybersecurity vulnerabilities because flaws in a network could lead to a complete compromise of an organization's systems.

Examples of Cyber Security Vulnerabilities

Here are a few examples of cybersecurity vulnerabilities

- Missing data encryption
- Lack of security cameras
- Unlocked doors at businesses
- Unrestricted upload of dangerous files
- Code downloads without integrity checks
- Using broken algorithms
- URL Redirection to untrustworthy websites
- Weak and unchanged passwords
- Website without SSL

Types of Cyber Security Vulnerabilities

Here are a few common types of cybersecurity vulnerabilities:

1.System Misconfigurations

Network assets can cause system mistakes with incompatible security settings or restrictions. Networks are frequently searched for system errors and vulnerable spots by cybercriminals. Network misconfigurations are increasing as a result of the quick digital revolution. Working with knowledgeable security professionals is crucial when implementing new technology. Cybercriminals frequently search networks for vulnerabilities and misconfigurations in the system that they can exploit.

2.Out-of-date or Unpatched Software

Hackers frequently scour networks for vulnerable, unpatched systems that are prime targets, just as system configuration errors do. Attackers may use these unpatched vulnerabilities to steal confidential data, which is a huge threat to any organization. Establishing a patch management strategy that ensures all the most recent system updates are applied as soon as they are issued is crucial for reducing these types of threats.

3.Missing or Weak Authorization Credentials

Attackers frequently utilize brute force methods, such as guessing employee passwords, to gain access to systems and networks. Therefore, they must therefore train employees on cybersecurity best practices to prevent the easy exploitation of their login credentials. An endpoint system security will be a great addition to all laptop or desktop devices.

4.Malicious Insider Threats

Employees with access to vital systems may occasionally share data that enables hackers to infiltrate the network, knowingly or unknowingly. Because all acts seem genuine, insider threats can be challenging to identify. Consider purchasing network access control tools and segmenting your network according to employee seniority and experience to counter these risks.

5.Missing or Poor Data Encryption

If a network has weak or nonexistent encryption, it will be simpler for attackers to intercept system communications and compromise them. Cyber adversaries can harvest crucial information and introduce misleading information onto a server when there is weak or unencrypted data. This may result in regulatory body fines and adversely jeopardize an organization’s efforts to comply with cyber security regulations.

6.Zero-day Vulnerabilities

Zero-day vulnerabilities are specific software flaws that the attackers are aware of but that a company or user has not yet identified.Since the vulnerability has not yet been identified or reported by the system manufacturer, there are no known remedies or workarounds in these situations. These are particularly risky because there is no protection against them before an attack occurs. Exercising caution and checking systems for vulnerabilities is crucial to reducing the risk of zero-day attacks

Causes of Cyber Security Vulnerabilities

There are many causes of cyber security vulnerabilities. A few of them are as follows:

1. **Complexity:** The likelihood of errors, defects, or unauthorized access increases with complex systems.
2. **Familiarity:** Attackers may already be acquainted with common code, operating systems, hardware, and software that result in well-known vulnerabilities.
3. **Connectivity:** Vulnerabilities are more likely to exist in connected devices. It is better to avoid connecting to multiple devices unnecessarily.
4. **Poor Password Management:** This can cause several data breaches because of weak or repeated passwords. It is important to change passwords using strong password generators regularly.
5. **Internet:** Spyware and adware that can be loaded on computers automatically are abundant on the internet.
6. **Operating System Flaws:** Operating systems can also be flawed. Operating systems that aren't safe by default might provide users unrestricted access and serve as a haven for malware and viruses.
7. **Software Bugs:** Sometimes, programmers may unintentionally introduce a vulnerability that can exploit.
8. **Unchecked User Input:** If software or a website presumes that all user input is secure, SQL injection may be executed without the user's knowledge.
9. **People:** For most organizations, social engineering poses the biggest concern. Therefore, one of the main sources of vulnerability can be people.

Vulnerability Management

The process of identifying, classifying, resolving, and mitigating security vulnerabilities is known as vulnerability management. Vulnerability management consists of three key components:

1. Vulnerability detection
2. Vulnerability assessment
3. Addressing Vulnerabilities

Vulnerability Detection

The process of vulnerability detection has the following three methods:

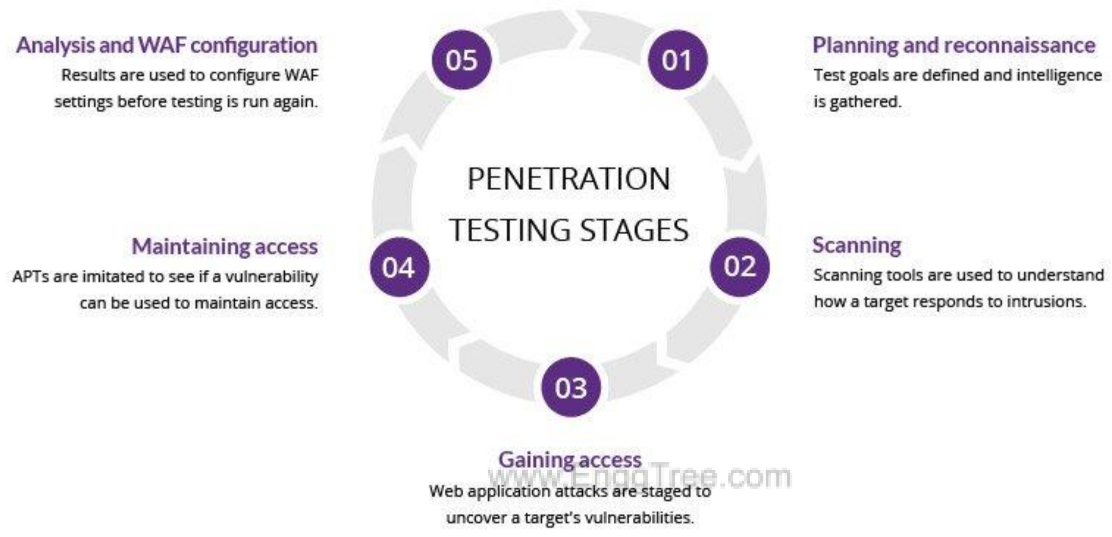
- Vulnerability scanning
- Penetration testing
- Google hacking

4.12.2 PENETRATION

A penetration test, also known as a pen test, is a simulated cyber attack against your computer system to check for exploitable vulnerabilities. In the context of web application security, penetration testing is commonly used to augment a web application firewall (WAF).Pen testing can involve the attempted breaching of any number of application systems, (e.g., application protocol interfaces (APIs), frontend/backend servers) to uncover vulnerabilities, such as unsanitized inputs that are susceptible to code injection attacks.Insights provided by the penetration test can be used to fine-tune your WAF security policies and patch detected vulnerabilities.

Penetration testing stages

The pen testing process can be broken down into five stages.


PENETRATION TESTING STAGES
01
Planning and reconnaissance
Test goals are defined and intelligence is gathered.
02
Scanning
Scanning tools are used to understand how a target responds to intrusions.
03
Gaining access
Web application attacks are staged to uncover a target's vulnerabilities.
04
Maintaining access
APTs are imitated to see if a vulnerability can be used to maintain access.
05
Analysis and WAF configuration
Results are used to configure WAF settings before testing is run again.
www.EnggTree.com

1. Planning and reconnaissance
The first stage involves:

- Defining the scope and goals of a test, including the systems to be addressed and the testing methods to be used.
- Gathering intelligence (e.g., network and domain names, mail server) to better understand how a target works and its potential vulnerabilities.

2. Scanning
The next step is to understand how the target application will respond to various intrusion attempts. This is typically done using:

- **Static analysis** – Inspecting an application's code to estimate the way it behaves while running. These tools can scan the entirety of the code in a single pass.
- **Dynamic analysis** – Inspecting an application's code in a running state. This is a more practical way of scanning, as it provides a real-time view into an application's performance.

3. Gaining Access
This stage uses web application attacks, such as cross-site scripting, SQL injection and backdoors, to uncover a target's vulnerabilities. Testers then try and exploit these vulnerabilities, typically by escalating privileges, stealing data, intercepting traffic, etc., to understand the damage they can cause.

4. Maintaining access
The goal of this stage is to see if the vulnerability can be used to achieve a persistent presence in the exploited system— long enough for a bad actor to gain in-depth access. The idea is to imitate advanced persistent threats, which often remain in a system for months in order to steal an organization's most sensitive data.

5. Analysis
The results of the penetration test are then compiled into a report detailing:

- Specific vulnerabilities that were exploited
- Sensitive data that was accessed
- The amount of time the pen tester was able to remain in the system undetected

This information is analyzed by security personnel to help configure an enterprise's WAF settings and other application security solutions to patch vulnerabilities and protect against future attacks.

PENETRATION TESTING METHODS

External testing
External penetration tests target the assets of a company that are visible on the internet, e.g., the web application itself, the company website, and email and domain name servers (DNS). The goal is to gain access and extract valuable data.

Internal testing
In an internal test, a tester with access to an application behind its firewall simulates an attack by a malicious insider. This isn't necessarily simulating a rogue employee. A common starting scenario can be an employee whose credentials were stolen due to a phishing attack
.

Blind testing
In a blind test, a tester is only given the name of the enterprise that's being targeted. This gives security personnel a real-time look into how an actual application assault would take place.

Double-blind testing
In a double blind test, security personnel have no prior knowledge of the simulated attack. As in the real world, they won't have any time to shore up their defenses before an attempted breach.

Targeted testing
In this scenario, both the tester and security personnel work together and keep each other appraised of their movements. This is a valuable training exercise that provides a security team with real-time feedback from a hacker's point of view.

4.12.3 ENCRYPTION TECHNOLOGIES

Types of encryption

- **Bring your own encryption (BYOE)** is a cloud computing security model that enables cloud service customers to use their own encryption software and manage their own encryption keys. BYOE may also be referred to as *bring your own key* (BYOK). BYOE works by enabling customers to deploy a virtualized instance of their own encryption software alongside the business application they are hosting in the cloud.
- **Cloud storage encryption** is a service offered by cloud storage providers whereby data or text is transformed using encryption algorithms and is then placed in cloud storage. Cloud encryption is almost identical to in-house encryption with one important difference: The cloud customer must take time to learn about the provider's policies and procedures for encryption and encryption key management in order to match encryption with the level of sensitivity of the data being stored.
- **Column-level encryption** is an approach to database encryption in which the information in every cell in a particular column has the same password for access, reading and writing purposes.
- **Deniable encryption** is a type of cryptography that enables an encrypted text to be decrypted in two or more ways, depending on which decryption key is used. Deniable encryption is sometimes used for misinformation purposes when the sender anticipates, or even encourages, interception of a communication.
- **Encryption as a Service (EaaS)** is a subscription model that enables cloud service customers to take advantage of the security that encryption offers. This approach provides customers who lack the resources to manage encryption themselves with a way to address regulatory compliance concerns and protect data in a multi-tenant environment. Cloud encryption offerings typically include full-disk encryption (FDE), database encryption or file encryption.
- **End-to-end encryption (E2EE)** guarantees data being sent between two parties cannot be viewed by an attacker that intercepts the communication channel. Use of an encrypted communication circuit, as provided by Transport Layer Security (TLS) between web client and web server software, is not always enough to ensure E2EE; typically, the actual content being transmitted is encrypted by client software before being passed to a web client and decrypted only by the recipient. Messaging apps that provide E2EE include Facebook's WhatsApp and Open Whisper Systems' Signal. Facebook Messenger users may also get E2EE messaging with the Secret Conversations option.
- **Field-level encryption** is the ability to encrypt data in specific fields on a webpage. Examples of fields that can be encrypted are credit card numbers, Social Security numbers, bank account

numbers, health-related information, wages and financial data. Once a field is chosen, all the data in that field will automatically be encrypted.

- **FDE** is encryption at the hardware level. FDE works by automatically converting data on a hard drive into a form that cannot be understood by anyone who doesn't have the key to undo the conversion. Without the proper authentication key, even if the hard drive is removed and placed in another machine, the data remains inaccessible. FDE can be installed on a computing device at the time of manufacturing, or it can be added later on by installing a special software driver.
- **Homomorphic encryption** is the conversion of data into ciphertext that can be analyzed and worked with as if it were still in its original form. This approach to encryption enables complex mathematical operations to be performed on encrypted data without compromising the encryption.
- **HTTPS** enables website encryption by running HTTP over the TLS protocol. To enable a web server to encrypt all content that it sends, a public key certificate must be installed.
- **Link-level encryption** encrypts data when it leaves the host, decrypts it at the next link, which may be a host or a relay point, and then reencrypts it before sending it to the next link. Each link may use a different key or even a different algorithm for data encryption, and the process is repeated until the data reaches the recipient.
- **Network-level encryption** applies cryptoservices at the network transfer layer -- above the data link level but below the application level. Network encryption is implemented through Internet Protocol Security (IPsec), a set of open Internet Engineering Task Force (IETF) standards that, when used in conjunction, create a framework for private communication over IP networks.
- **Quantum cryptography** depends on the quantum mechanical properties of particles to protect data. In particular, the Heisenberg uncertainty principle posits that the two identifying properties of a particle -- its location and its momentum -- cannot be measured without changing the values of those properties. As a result, quantum-encoded data cannot be copied because any attempt to access the encoded data will change the data. Likewise, any attempt to copy or access the data will cause a change in the data, thus notifying the authorized parties to the encryption that an attack has occurred.

UNIT 5
INTERNET OF MEDICAL THINGS

Enhancing the Performance of Decision Tree Using NSUM Technique for Diabetes Patients

Diabetesisacommondiseaseamongchildrentoadultinthisera.Topreventthediseasesisveryimportantbecause it saves the human lives. Data mining technique helps to solve the problem of predicting diabetes.Ithasstepsofprocessestopredicttheillness.Featureselectionisanimportantphaseindataminingprocess.In feature selection when dimension of the data increases, the quantity of data required to deliver adependable analysis raises exponentially. Numerous different feature selection and feature extractiontechniques are present, and they are widely used filter-based feature selection method is proposed whichtakes advantage of the wrapper, Embedded, hybrid methods by evaluating with a lower cost and improves the performance of a classification algorithm like a decision tree, support vector machine, logistic regression and soon. To predict whether the patient has diabetes or not, we introduce an ovel filter method ranking technique called Novel Symmetrical Uncertainty Measure (NSUM). NSUM technique experimentally shows that compared to the other algorithms in filter method, wrapper method, embedded method and hybrid method it proves more efficient in terms of Performance, Accuracy, Less computational complexity. The existing technique of symmetric uncertainty measure shows less computational power and high performance, but it lacks in accuracy. The aim of the NSUM method is to overcome the drawback ofthe filter method, i.e., less accuracy compared to other methods. NSUM technique results show highperformance, improved accuracy, and less computational complexity. NSUM method runs in 0.03 s with89.12%asaccuracy by usingWekatool.

SmartCityHealthcareCyberPhysicalSystem:Characteristics,Technologiesand Challenges

The recent pandemic has demanded a strong and smart healthcare system which can monitor the patientsefficiently and handle the situation that arises from the outbreak of the disease. Smart healthcare cyberphysical systems are the future systems as they integrate the physical and cyber world for efficientfunctioning of medical processes and treatment through external monitoring and control of patients,medicaldevicesandequipmentforcontinuouscommunicationandinformationexchangeofphysiologicaldata.Technologieslike InternetofThings,MachinelearningandArtificialIntelligencehave givenbirthto smart cyber physical systems like Smart Healthcare Systems, Smart Homes, Smart Vehicular Systemsand Smart Grid. Such systems are interdisciplinary in nature with multitude of technologies contributingto its effective working. This paper presents a case study on healthcare cyber physical systems presentingits characteristics, role of various technologies in its growth and major challenges in successfulimplementationofcyber physicalmedication systems.

Keywords: Artificialintelligence,Healthcarecyberphysicalsystems,Symbioticcyberphysicalsystems,Security

Introduction

In the current pandemic, a strong healthcare system is a backbone for any smart city. Effective andefficientmonitoringofpatientstogetherwithregularsupplyofnecessarymedicinesandtreatmentwithaid of medical devices is possible when a system can meet the demand and supply situation. Such ascenario is possible with a predictive mechanism which forecasts the healthcare situation and workssmartlyfor handling themedicalemergency.

Smart Healthcare Cyber Physical Systems (SHCPS) are the future systems capable of supporting themedical fraternity in handling the pandemic situation effectively. Such systems comprise of physicalworldofpatients,medicaldevicesandequipment; externallycontrolledandmonitoredmedicaltreatment,connected with cyber world through communication networks for data transfer and information exchangeof physiological data which are analysed for feedback and control signals. Such systems improve thequalityofmedicalcareby providingefficientand smartservices[1].

Thedesignanddevelopmentofhealthcarecyberphysicalsystems havebroughtsomeopenissues fordiscussion like autonomy level, security and reliability which are vital for healthcare cyber physicalsystems.

Autonomy Level

Cyber physical systems can be classified from low to high level of autonomy based on the tasksperformedbythehumansintheloop.Technologieslikemachinelearningandartificialintelligenceplayamajorroleinshifting thecontrolfromhumantomachineandartificialdistributed networks [2].

Variousfactorsthatarecrucialindefiningtheautonomylevelinhealthcarecyberphysicalsystems(HCPS)arebroadly categorizedintotwo domains.

- *Equipment based* Factors like the medical device type, interaction type and duration with the patient aredecidingfactorsforautonomylevel.Medicalrobotsforservingfoodandmedicinestocoronaviruspatientscanworkautonomouslyforassistingthe Medicaresystems.TheteamofresearchersfromHongKonghavecreatedahumanoidrobotcalled‘Grace’forinteractingwithisolatedcoronaviruspatientsandprovides services like temperature recording and responsiveness through the thermal camera fitted in the chest.Therobotwillalsosociallyinteractwiththecoronaviruspatientsinordertopreventthestressarisingfromisolation[3].
- *Patientrelated*Patientrelatedfactorsincludethediseasetypeandpatientrisklevel.Communicable diseaseslikecoronavirusnecessitatehighlevelofautonomousMedicarecyberphysicalsystemswhereas high patient risk level require continuous medical expert support along with the autonomous Medicaresystems.

SecurityMechanisms

A number of technologies play a vital role in the cyber physical system which can be classified intodistinctiveareasrangingfromdatacollectionwithsensortechnologyandIOT;handlingbigdataanditsstorage withCloudcomputing;dataanalytics anddecisionmakingusingartificialintelligence; control andcoordination signals to smart machines and actuators thus leading to the commencement ofnew era termed by Industry 4.0 [4] and specifically Healthcare 4 for medical applications. Acomprehensive security mechanism for authenticated, confidential and secure communication betweencyberandphysicalworldforms an integralpartofframeworkduring thedesignanddevelopmentof cyberphysical systems. The criticality of healthcare applications with high levels of autonomy demands apreventive approach for defence against cyber-attacks [5] and thus the Healthcare Cyber physical systemsmustincorporatethefollowingin their securitymechanism:

- Identificationofcyber-attacksonsensors.

- Lightweightcryptographyfortransmittingphysiologicaldata.
- EncryptionmechanismsandBlockchaintechnologyforsecurestorageoncloud-basedsystems.
- Encryptedfeedbacksignalstoactuators.

The data transmitted from patients to medical expert is vulnerable to attacks and hence requiresencryption techniques at the client end. Many cryptographic techniques exist in literature, the broadclassification is symmetric cryptography and asymmetric cryptography. Biological cryptographicapproachbasedonaminoacidcodes [6]is usedfortransmittingthedatafrompatients tohealthcareexperts in telemonitoring healthcaresystem.

Reliability

Reliability is an important metric for cyber physical systems and of utmost significance to healthcareindustry.Figure1showsthereliabilityofhealthcarecyberphysicalsystems(HCPS)whichisdependenton hardware units like health sensors and actuators, software reliability on software systems computingpatient's health status, and network reliability determined by communication networks for transfer ofpatientdata.

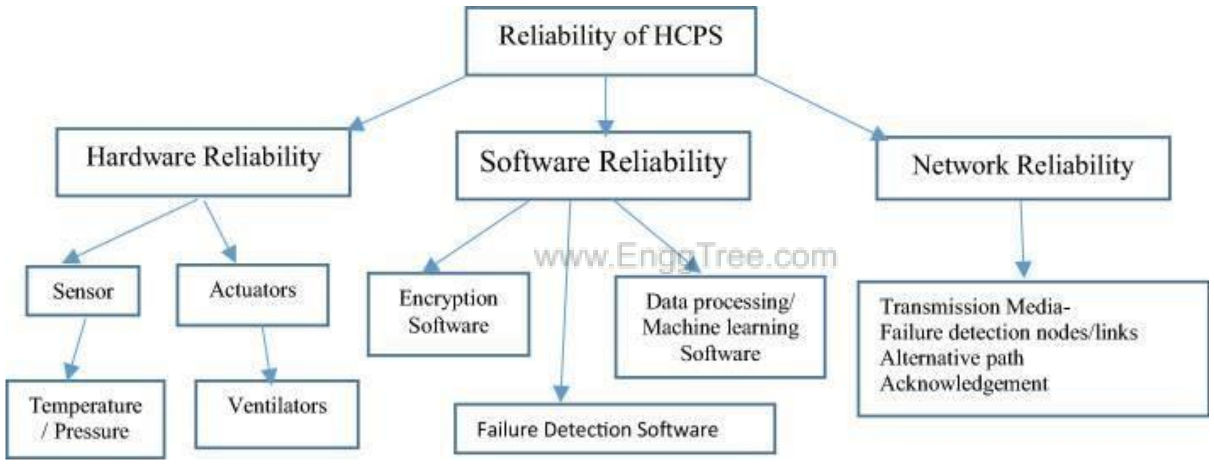


Fig.1

Thecommencementofcyberphysicalsystemsrequiresoptimizationofresourcesandself-adaptivebehaviourforefficient,reliableandimprovedservices[7].Theautonomoussystemsmustbeabletoidentifythe failure of different components in the closed loop system and take corrective measures in terms ofhandlingthetasks.Self-adaptivecomponentsofcyberphysicalsystemslearnfromthepastdataandbehaveinthecurrentscenario.Smartmachineslikehealthcarerobotscanself-organizetodynamicenvironmenttomeetthechallengeof servicequality.

Thispaperhighlightsthefollowing:

- (i)Characteristicsofhealthcarecyberphysicalsystems
- (ii) Amalgamation of technologies that have contributed to the growth and implementation ofhealthcarecyber physicalsystems.
- (iii)Realtimechallengeswhichpromptheresearchers,healthcareandmanufacturingsectortoretrospect and consider during design and effective execution of healthcare cyber physicalsystems.

The need for continuous communication, control and collaboration for efficient and effective systems forquality of service has brought a new era called Industry 4.0 that introduces cyber physical systems. ThetermcyberphysicalsystemwasfirstcoinedbyHelenGillatNSFinUSintheyear2006.Acyberphysicalsystemintegratescyberandphysicalworldwithsensors,whichactlikedatacollectorstogatherinformationliketemperature,pressureorspeed/activitytimefortransmissiontocyberworldforstorageinservers.Thisdataisfurtherprocessed andanalysedtoactasstimuliforcontrolandcoordinationsignalstoactuatorsthusformingaclosed loop systemasshownin Fig.2.

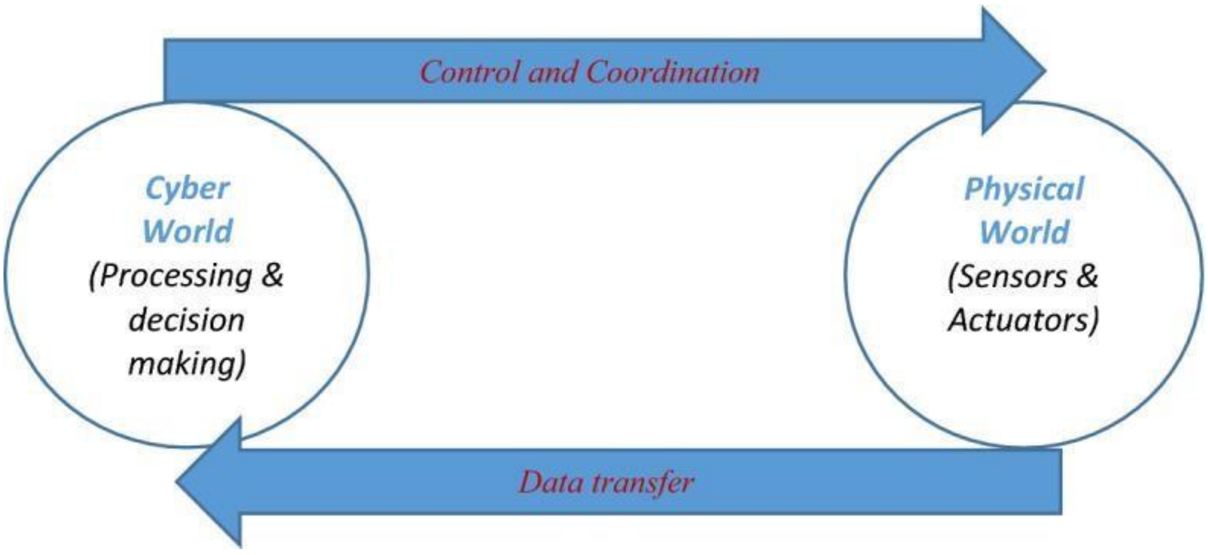
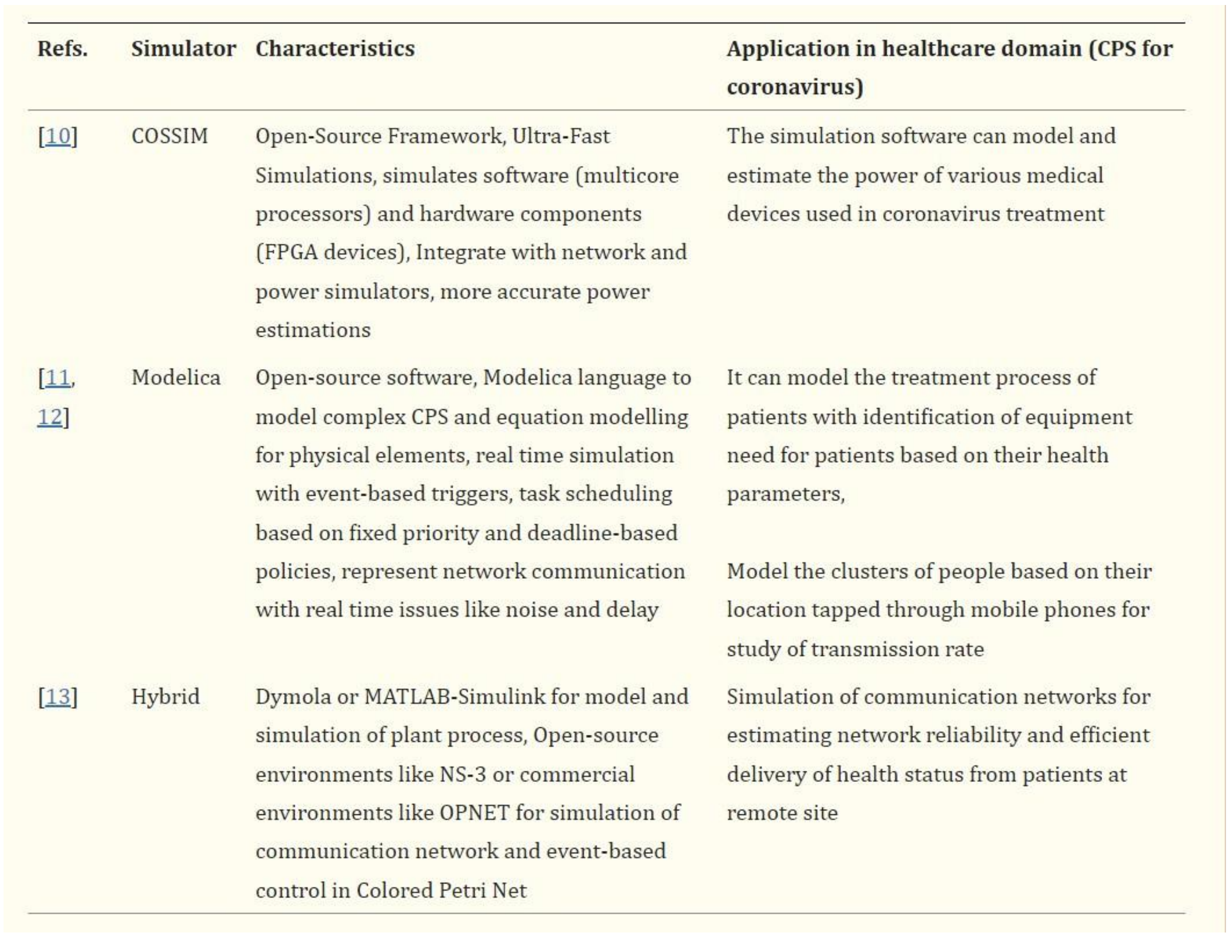

Control and Coordination
Cyber World (Processing & decision making)
Physical World (Sensors & Actuators)
Data transfer

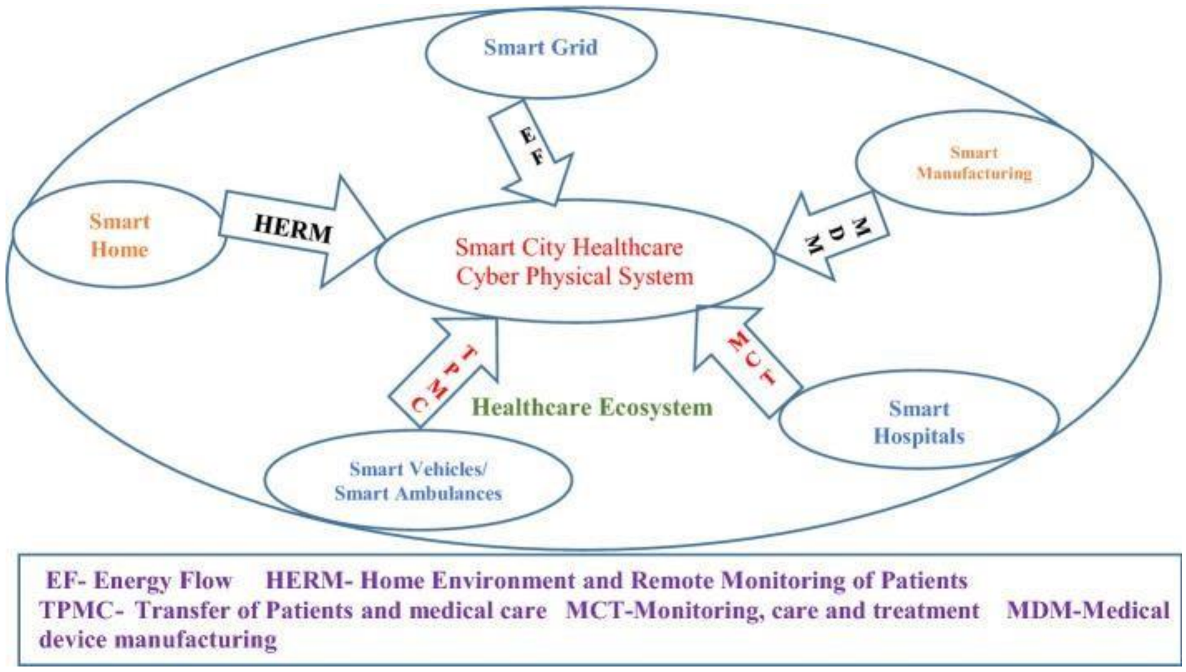
Fig.2

The systematic design and modelling of cyber physical systems play a key role in defining the architectureofthesystemwithvarioussoftwareandhardwarecomponents,thefunctionalrolesofeachcomponent; andthe communication and control mechanism between cyber and physical space. Thereafter, the simulationofthedesignwithwelltestedandvalidationstrategiesareappliedforfinaladoptionofthemodelinthereal time setup [8]. Unlike the conventional approach to design and deployment, the authors in [9] suggest ashift from design to runtime and implementation phase for critical decisions, which depend on real timeinputsandenvironment.Table
Table11presentsthevariousCPSsimulationsoftwareandtheircharacteristics, which can be used to test various concepts of modularity, scalability and complexity ofcyberphysicalsystemsin theapplication domain.



Refs.	Simulator	Characteristics	Application in healthcare domain (CPS for coronavirus)
[10]	COSSIM	Open-Source Framework, Ultra-Fast Simulations, simulates software (multicore processors) and hardware components (FPGA devices), Integrate with network and power simulators, more accurate power estimations	The simulation software can model and estimate the power of various medical devices used in coronavirus treatment
[11, 12]	Modelica	Open-source software, Modelica language to model complex CPS and equation modelling for physical elements, real time simulation with event-based triggers, task scheduling based on fixed priority and deadline-based policies, represent network communication with real time issues like noise and delay	It can model the treatment process of patients with identification of equipment need for patients based on their health parameters, Model the clusters of people based on their location tapped through mobile phones for study of transmission rate
[13]	Hybrid	Dymola or MATLAB-Simulink for model and simulation of plant process, Open-source environments like NS-3 or commercial environments like OPNET for simulation of communication network and event-based control in Colored Petri Net	Simulation of communication networks for estimating network reliability and efficient delivery of health status from patients at remote site

Somesystemshaveasymbioticrelationshipbetweenphysicalworldandvirtualworldgivingrisetoanewterm "symbiotic cyber physical systems" [14]. The term symbiotic relationship has arisen from biologywhere two organisms benefit from each other and such a relationship is symbiotic in nature. Smart Grid isa perfect example of symbiotic cyber physical system where technology drives to generate energy smartlyand the energy is the source of power for the technology elements in the cyber physical system. The smarthealthcare cyber physical systems are symbiotic to various autonomous systems as shown in Fig. 3. Smartgrid, smart home, smart vehicular systems, smart hospitals and smart manufacturing units provide servicesfor a smart city healthcare system. In turn, the smart city healthcare system provides healthcare to itsinhabitants. These inhabitants or humans are working in various organizations and manufacturing units,thusprovidingtheirserviceorroleinvariousautonomousunits.Todefinethepreciseroleofhumansintheautonomous CPSsystemsisabigandchallengingtask.Aneffectiveapproachinthisdirectionistoidentifythe control strategies and interactions of humans in the closed loop CPS at early stages of softwaredevelopmentandthereafter, validatethrough fastprototyping techniques[15].


Smart Grid
EF
Smart Manufacturing
MDM
Smart Home
HERM
Smart City Healthcare Cyber Physical System
TPMC
MCT
Healthcare Ecosystem
Smart Hospitals
Smart Vehicles/ Smart Ambulances
EF- Energy Flow HERM- Home Environment and Remote Monitoring of Patients
TPMC- Transfer of Patients and medical care MCT-Monitoring, care and treatment MDM-Medical device manufacturing

Thecyberphysicalsystemsdifferintermsoftheircharacteristicslikeautonomylevelwhichdeterminethetaskscon trolledandoperatedbyhumans,scalewhichdependsonthenumberofdevicesconnected andriskmanagement which depends on the critical environment of the system. Table Table22presents the variousapplications areas, characteristic features, technologies and services provided by the CPS in differentdomains in faceof thevarioussecurity risksand challenges.

Applicatio narea	**Characteristics**	**Technologies**	**Services**	**Security/risks/challenges**
Healthcare4.0 [4]	Wearabl edevices , IOT of medicaldevic es	Dataanalyticso npatient'sdata	Telemedicine, Roboti csurger y, Mobilehealth	The accuracy of sensors andother monitoring devices is abigchallengeasitdetermine sthemedicationlevel
Manufacturin gunit [4,16–18]	Smart machinesliker obots, Smartproduct	Machinetomachin ecommunication	Autonomou smachines inproductio nprocess, Self-organizingcap ability ofsmart machines,	Delaysabovesomethreshol dvalue lead to disruption ofmanufacturingcycle Malfunctioningofnodes

Applicatio narea	**Characteristics**	**Technologies**	**Services**	**Security/risks/challenges**
			Detection andmanageme nt ofindustrialha zards, Node failuredete ctionthroug hsimilarity iepatternm atching	
Smart home[19– 21]	Cameras,te mperatures ensors Smart televisions,smar t locks,smart lights,smart switchesandsma rtmeters	Remotecontrollingo fhomedevices	Reduce powerconsu mption, Detect andclassify safetyhazard s, Remote healthmanage mentand emergencyser vices	Attackonprivacyofresidents by gathering data like userinitiatedanddeviceactio ns Local network attacks bybringing devices close tovicinityof home
Smartvehicles Autonomous [22,23]	Cameras, sensors, Vehiclestates	Sensingtechnologyfor obstacledetection, Navigationtrajector yundercontrolof software,Prediction mechanism todeterminemotion ofothervehicles, Proactivemechanisms forfaultdetectionand management	Stable vehiclemove mentwithcon tinuousmonit oring ofvehicleheal th, Lane detectionand emissioncont rolmechanis ms	Softwaredefects Difficulty in integration ofdiagnostictoolsinframew ork
Smartgrid [24]	Powerline, Communicatio nline,	Remote control andmonitoringofele ctrical componentsingrid,	Reliabledeliv ery ofpower withenergy storageatlarge scale,	Vulnerabletocyberattacks affecting public life due todependency on power forrunning of appliances at homeandequipmentinprofes sional

				sector
Applicatio narea	**Characteristics**	**Technologies**	**Services**	**Security/risks/challenges**
	Sensors,Act uators,Smart meters	Distributed energyresources act asmicrogenerationun its, Distributed renewableenergygen eration Big data analytics onpower generated andusage, Predictionand recommendationusin gmachine learning techniques	Energycons ervationwit h reducedloss andgreenho useemission s	

CharacteristicsofSmartCityHealthcareCyberPhysicalSystem

HealthcareCyberPhysicalSystemscanbedividedintovariouslevels:

- UnitlevelHCPS
- IntegrationlevelHCPS
- SystemlevelHCPS
- AcceptancelevelHCPS
- EvolutionarylevelHCPS

Unit level HCPS are basic or the first level of healthcare cyber physical systems which providemonitoring and control of patients in intensive care units or at the hospital level. At this level itcontinuously monitors the physiological parameters like temperature, pressure, heart rate etc. of thepatients and feed the data to the intelligent systems which analyse and control the health actuatorsconnectedtothepatient.Also,thehealthstaffis intheHCPSloopforsupportandinformationtohealthexperts for immediatehealthcareaidtothepatients.

Integration level HCPS are the second stage of HCPS where the hospitals integrate with smart homes toprovide remote monitoring and remote healthcare service to the patients. In case of transfer of high-riskpatients in ambulances to hospitals, the latter can integrate with smart ambulances to continuouslymonitorthepatient'shealthstatusandmakenecessaryemergencyservicesinhospital like availabilityofbed,ventilator supportetc.

SystemlevelHCPS arethethirdstageofHCPS wheredifferentautonomoussystems supporttheHCPS toform a Smart City Healthcare Cyber Physical System. The smart grid, the powerhouse of energy andbackbone for various cyber physical system together with smart home, smart ambulances, smart hospitalmanufacturing units and smart hospitals form a healthcare ecosystem providing quality of healthcareservicetothepatients.

Acceptance level HCPS is the level of HCPS where the researchers, technologists, engineers, healthexperts,academicianscoordinatetomakethehealthcaresystemeffectivewithpoliciesandstandardsorientedtowards successfulimplementationof thehealthcareecosystem.

Evolutionary level HCPS is the ideal future HCPS systems which have properties of self-adaptability andself-management. Self-adaptive components of cyber physical systems learn from the past data andbehaveinthecurrentscenario.Hence,theroleofevolutionarybehaviouriscriticalandofsignificanceforthedynamicenvironmentof cyber physicalsystems.

The Cyber Physical System has physical components and processes which can be represented by a **statediagram and different states**: healthy, unhealthy, critical and non-working state as shown in Fig. 4. Theoverall health of the CPS depends on the working conditions of different components in the closed loopsystem[25].

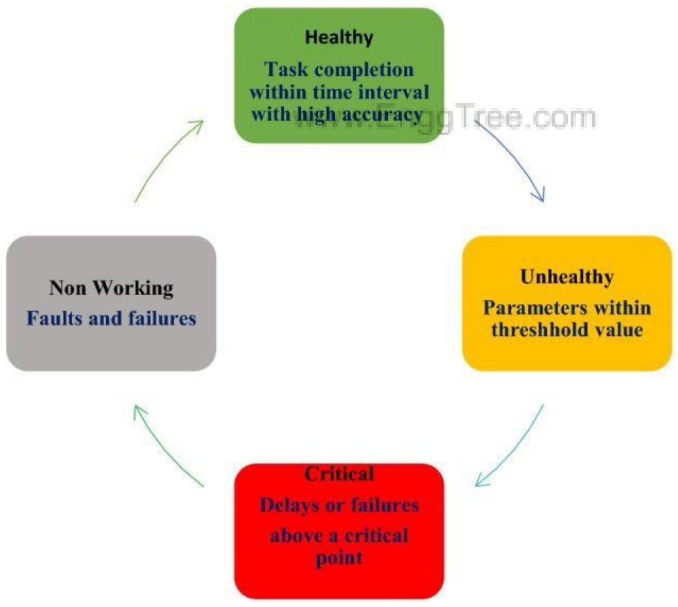

Healthy
Task completion within time interval with high accuracy
Unhealthy
Parameters within threshhold value
Critical
Delays or failures above a critical point
Non Working
Faults and failures

Fig.4

The nature of devices in CPS is **heterogeneous** which range from sensors, actuators to physical machinescontrolled by external inputs like mobile devices at dew layer and systems at cyber level. The input andoutputofdataofvariousdevicesdifferinthedataformats,whichmaybestructured/unstructuredinnature,whichnecessitatethe conversionofformatsthroughinterfaces[16].Thehealthcarecyberphysicalsystemsare application domain where time is a critical factor in determining the performance and reliability of thesystem. A delay in seconds or microseconds in medication to patients can reduce the effectiveness anddependabilityofthesystem.

Cyber physical systems are monitored and controlled continuously for effective operation and hence thisgives rise to a new term called health-monitoring system (HMS) for CPS [25]. HMS can perform dualfunction:tocheckthestatusofeachphysicalcomponentinCPSandtoapplypre-emptiveapproachin determining probabilistic health of CPS. Study of behavioural models of physical components is a passivemonitoring approach where as a stimulus-based response is an active strategy to determine for anomalydetection[25].

Basedontheabovecharacteristics,theunitandtheintegrationlevelofHCPSareanalysedintermsofcharacteristics of CPSin TableTable33.

Table3

CharacteristicsofunitandintegratedlevelofHCPS

Level ofhealth CPS	**Characteristics**			
	Statetransition&statediagram	**Heterogeneousdevices/dataformats**	**Timecriticalapplications**	**Seamlessmonitoring &control**
Unitlevel	Healthy working state ofall sensors and actuators insmart hospitals as patientsareinriskzone Therefore, earlyreplacement of faultymedicaldevices	Medical applicationinterfacestodealwithheterogeneousdata	Timecriticalandrequiresefficientdecisionmaking	Regular monitoring andcontrolinSmarthospitals
Integrationlevel[26]	Healthy state depicts goodworkingcondition, Smart ambulancesperiodically checked forworking condition of itsmedical devices andconnectivity, Coordination andcommunicationforintegration of	differentautonomous systems forhealthyworkingstate, Requires continuousmonitoringforfailure	Audio, image and video(behavioural)analysis, Different autonomoussystems connectedrequire medical API's todeal with heterogeneousdatasets	Homemonitoringisnottime critical whereasmonitoringofpatients duringtransitfromhometohospitalinsmartamb

ulanceisti mecritical	Monitoring andcontrol ofelderlypatients
	Remindersyst ems
	Wearablesens ors

	Characteristics			
Level ofhealth CPS	**Statetransition&state diagram**	**Heterogeneous devices/dataformats**	**Timecriticalapplications**	**Seamlessmonitoring &control**
	detection of nodes andlinks			

Section 4, presents the role of various technologies in seamless monitoring the time critical domain withheterogeneous devices in cyber physical systems where each state is defined by a set of variables. Thetechnologies playacrucialrolein definingthehealthofthesystem.

TechnologiesinHealthcareCyberPhysicalSystems

The recent pandemic has put an unprecedented pressure on the healthcare system of any city in the worldand long working hours of medical experts and health workers. Hence, the future healthcare systemsdemand Smart City Healthcare Cyber Physical Systems where technology plays a significant role in itssuccessful implementation. Cyber Physical Systems is an application domain where there is integration ofplethoraoftechnologiesforsmartandefficientworkingofinterconnecteddevices.Internetisthebackbonefor communication in cyber physical systems and is primarily the most crucial technology and enabler forother technologies like IOT, cloud computing and blockchain [4]. This section presents the role of varioustechnologies like digital twin, IOT, big data, cloud computing, blockchain, artificial intelligence, machinelearningandrobotics inthefieldof healthcarecyberphysicalsystems as shownin Fig.5.

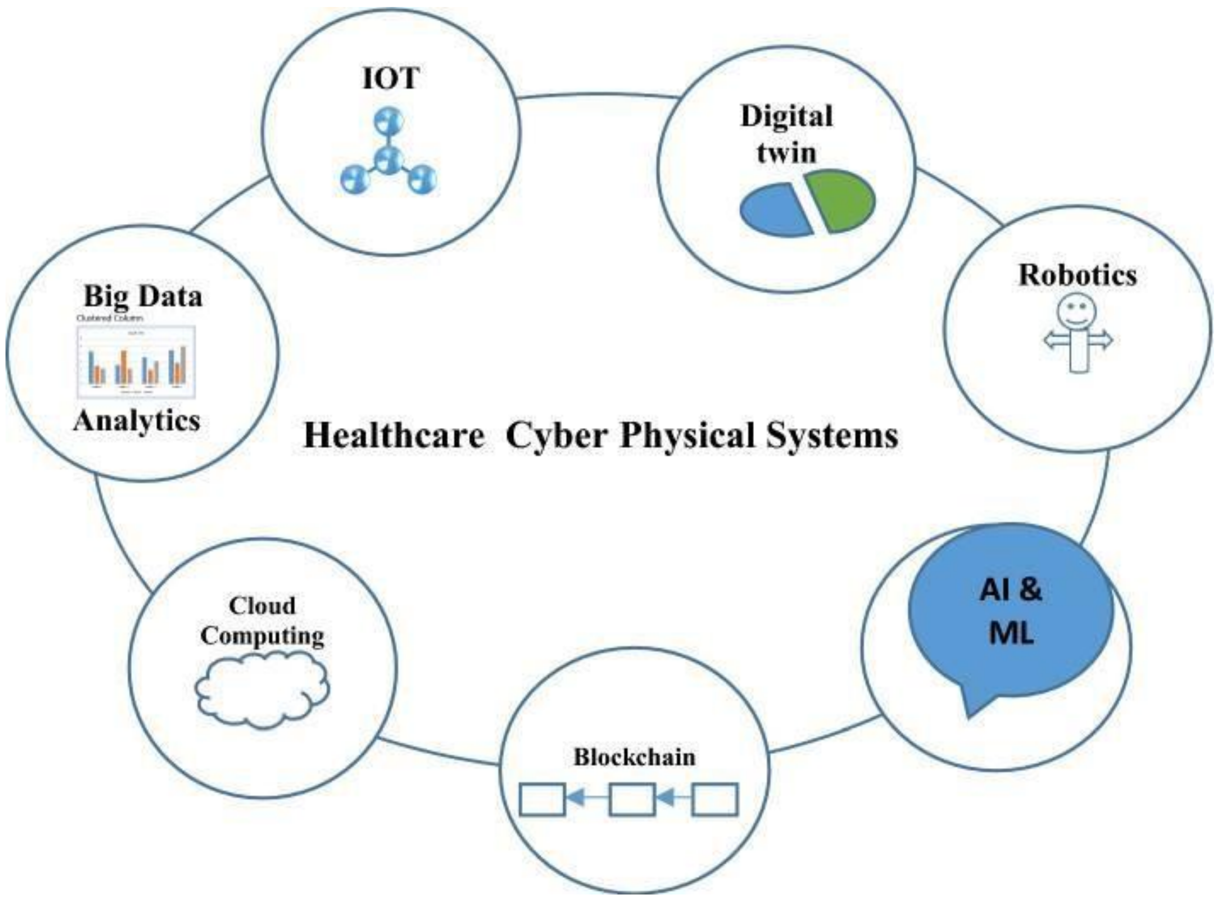


Fig.5

DigitalTwin

Digital twin as shown in Fig. Fig.66is a virtual twin for a physical object or a process. In smart healthcaresystems,thedigitaltwincreatesvirtualassetsincyberspacesothatthedigitalinformationofresourcesca nbeusedforplanning,controlandcoordination[27].DigitaltwinforSHCPSisbasicallyasimulationmodelfor medical devices and equipment, and for behavioural analysis of patient treatment process to assist thehealthcareexpertstostudy, analyseand predictthehealthstatusofpatients.

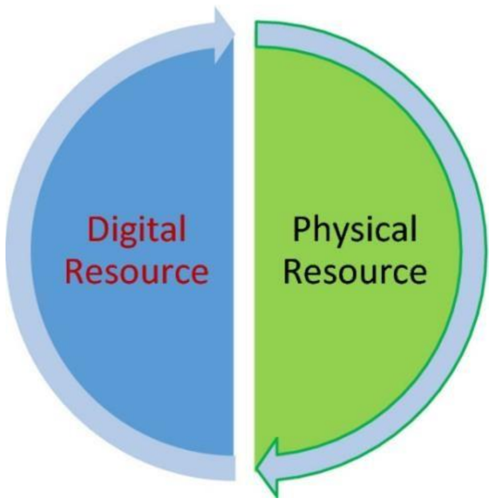


Fig.6

The digital twin maintains a resource graph for various medical equipment which is stored as a three-tuplevector represented by {resource id, allocation status, patient id}. The allocation status be busy or idle andexpectedreallocationmaydependuponthepatienthealth.Thepatienttreatmentprocesscanberepresentedby a graph where the nodes represent the health status at different intervals of time and the edges representthetransitionbasedonchangeinphysiologicalparameters.Thesegraphscanbeusedformachinelearning forcomputation andclassification of patient's healthstatus.

InternetofThings(IOT)

TheInternetofThingsbroughtaneweraofmachine-to-machinecommunication[4]whichcanbethroughwirelessnetworks,BluetoothandothertechnologieslikeNearFieldCommunication,radiocommunicationetc.TheIOTenablesinintegrationlevelsmarthealthcarecyberphysicalsystemswherethesensornetworksgenerate high volumes of data which are then transmitted to remote servers for analysis and controloperations. Since some devices in Medical IOT are resource constrained in terms of processing power andmemory therefore lightweight authentication and lightweight cryptographic schemes are essential forintegrity and confidentiality of data and information exchange. The Medical IOT has helped in recentpandemic in remote monitoring of patients [28, 29] through smart wearables like smart watch and smartbandtocollectpatientsheartrateandbloodpressure;andsmartthermometerstomeasurebodytemperature.Thesedevicesareconnectedtosmartphones,which ultimatelysend thedatatothecloudserversforhealthanalytics.

BigDataAnalytics

In healthcare cyber physical systems with IOT of Medical devices, machine to machine communicationamong heterogeneous nodes and sensors capturing data continuously lead to large data sets which may bestructured, unstructured or semi structured and hence require storage, processing and analysis for medicaladvice. The big challenge is to deal with unstructured data, complexity and veracity issues. The newcomputing paradigms with high processing power and big data technologies enable to extract hiddenpatterns and relationships in large amount of data which are gathered from various sources in healthcarecyberphysicalsystems.

Thepandemichasledtoregularcheck-upofphysiologicalparametersliketemperature,pressure,heartrateso that the symptoms could be easily predicted. The different type of data generated related to coronavirusare the human physiological parameters like temperature, pressure, heart rate; the hospital data whichincludecurrentcoronapatientintake,patienthealthstatus,facilitiesavailablelikenumberofbeds,ventilators;thecitydatalikethenumberofresidentscurrentlycoronainfected,numberofpatientshealthy,number of corona deaths, number of residents vaccinated. The researchers and analysts are interested insurvey of spread and predictive analysis for future effect. High end computing devices and deep learningmodels enable to work on large datasets and identify these hidden patterns. These hidden patterns can beunlockedusing statistical,machinelearning and deeplearningtechniques.

The complexity and uncertainty of real world data generated in real time systems like healthcare CPS canalso be dealt by methodology based on computational intelligence which includes fuzzy logic approachbased on approximation techniques and fuzzy rules for decision making and inferences; evolutionaryalgorithms like genetic programming and swarm intelligence for natural selection; and artificial neuralnetworks which have many hidden layers with neurons, mimic human brain and trained on large data setstosetthelearningparameters forpatternrecognition,predictionandclassificationofdatasets [30].

CloudComputing

Healthcare data silos are medical information of patients at discrete locations which may be redundant ornon-coherentbasedondatamanagementstrategies.Cloudcomputingisacomputingparadigmthatprovidesinfrastructure,resourcesandservicestoendusersonpayperusebasiswherethecloudserversprovidedatastorageandcomputingpowertousers.Inhealthcarecyberphysicalsystems,theElectronicHealthRecordscan be digitally stored in encrypted format at the cloud server so that it can be shared and accessed by thedifferententitieslikepatients,hospitalmanagement, insurancecompaniesandbanks[31].Thechallengingissueistoensuresafetyofpatientrecordswhichdependsonthesecurityframeworkadoptedforsafeguardingthekeygenerationcentrethatmaintainsprivatekeys of allauthenticusers[31].

Besides the storage of encrypted data in cloud servers, industry 4.0 standards have defined new servicemodels for cloud manufacturing like control as a service, machinery as a service and industry automationas a service [4]. These new cloud-based models can be applied for hard real tasks with fog computingresulting in service closed to end users with efficient real time communication in information transfer andcontrolbetween thehealthcarelevelandcontrollers.

DewComputing

The cloud architecture provides services which can be enhanced by edge/fog computing paradigm, adistributed service architecture which improves the efficiency of cyber physical systems by reducing thetransmission delay of services provided by cloud model. Dew computing further reduces the delay andprovides energy efficiency by introducing another level with smart interfaces or smart devices closer innetworktoIOTdevicesascomparedtoedgedevices.Thesesmartsystemsprovideprocessingcapabilities,work on data from physical components, control the actuators and have additional technology benefits ofscalabilityandresilience[32].TheDewcomputinglayerisveryusefulforprovidingservicesinhealthcaredomain [33] where the patients can be monitored more effectively by providing processing and analyticservicesclosetomonitoredarea. AdewcomputingarchitecturewithIOTdevices;sensorsandactuatorsinfirst layer; smart devices like smart phones and tablets in second layer called the dew computing layer;storage systems and network equipment in edge device layer; cloudlets and servers at fourth layer callededge server layer which represents the edge or fog computing distributed service and finally the fifth layerwith cloud servers providing various infrastructure and software based services[32]; can be very timeefficientforhealthcaredomains.ThelightweightapplicationsontabletsandsmartphonesinDewcomputingparadigmscansharepatientinformationandareinteroperablewhichhelpsincollaborationwithothersystemstobea partof healthcarecyber physicalsystemof systems.

Blockchain

Blockchaintechnologyprovidesadecentralisedand distributeddatabaseforsecureandauthenticaccesstoelectronichealthrecordsmaintainedbycloudservers[31].It providesadecentralisedplatformformaintaining untampered records of events [4] for various medical transactions that may be at device levelorvaccines;andeventsatpatientlevel.Theblockchainisdefinedbyachainofblocks,each withanumberof transactions or communications which are hashed and structured by a Merkle tree. Each block isidentified by a hash value and contains the hash of previous block with the exception that the first block iscalled genesis and has no parent hash value stored in it. Such a link with parent block gives an immutablestructurewhichcannotbetampered.

Suchimmutablestructurescanstorethetransactionsof differentmedicaldevices andvaccines; andinformation related to patients. Hence, this technology can have different blockchains for healthcare:Blockchain of medical devices, Blockchain of coronavirus infected patients, Blockchain of vaccines in ahospital.TheIBMblockchain[34]helpsintransparentdistributionofcoronavirusvaccine,bymaintainingthe transactions safe and traceable. QuillTrace [35] is a blockchain based technology that helps in trackingmedicinesinsupplychainandidentificationoffakemedicineswithhelpofQRcodeonmedicines.

ArtificialIntelligenceandMachineLearning

AI and machine learning can be applied at various aspects of healthcare which include medicines, medicalequipment,patientanddisease.Theresearchers,academiciansandhealthcareexpertsareworkingtogetherto extractusefulinformation,thehiddenpatternsfromlargedatabasesofpatientdata.Theselargedatabasesarealso usedtotrainmachinelearningmodelswhichareusedtoclassifythepatientsandhelpinautomaticdisease detection and thus support medical experts. The recent pandemic has seen the urgent need of studyofdrugdiscoveryforcoronavirus,thehighdemandofventilatorswhichhavebecomethelifesavingdevicefor thehigh-riskcoronaviruspatients,thestudyofpatientswithhighrisklevelsandtheeffectofdiseaseonotherorgansof thebody.

Basedontherisklevel,thecoronaviruspatientscanbedividedintolowrisklevelandhighrisklevel.Atthefirst stage the patients can be isolated intheir homes and the various physiological parameters like theirtemperature, pressure and heart rate at discrete time points can be sent to medical experts for predictingsymptoms based on machine learning techniques and healthcare can be provided through telemonitoring.The next stage demands hospitalization of patients with continuous medical support, care and monitoring.AI and machine learning algorithms have been used to predict the mortality rate of high-risk patients [36]by training the model with balanced dataset; and features are chosen based on wrapper and filter-basedapproaches.Thesefeaturesincludesymptoms,pre-healthstatusanddemographicfactorswhichsignificantlycontributeto thediseasestatusof thepatient.

Robots

Robots are autonomous machines which are programmed to perform a particular task with precision andaccuracy. They are cognitive models based on artificial intelligence with capabilities to continuouslycapture the environment data with sensors to work in complex environments and perform pre-definedactions with high frequency. Robots have a vast role to play in cyber physical systems like medical robotstoassistinsurgeryandpatientcare,industrialrobotstoperformmanufacturingtasksandsurveillancerobotsforsecurity and safety.

The recent pandemic has seen rising number of coronavirus cases which resulted in patient overload inhospitals.Thehealthcarerobotshavefoundakeyroleinpatientcareprovidingservicesfrompatienttestingtopatientservicebydeliveringregularmedicines,foodetc.RobotslikeMoxiperformvariousserviceslikedelivering PPE kits, covid 19 tests and provides pick/drop service to patients [37]. The robot Mitra assistshealth staff by taking temperature readings of patients and helps patients in connecting with their relativesthroughvideo conferencing [38].

Industrialrobotshelpinmanufacturingproductslike covidtestingkitsandventilatorswithhighfrequencyand precision; and short development life cycle. Industrial cyber physical working environments withhuman robot collaboration aim for agile product development and demand for context awareness in robotswhich identify human working zones and adjust their area and speed [39]. Unlike machines, humanbehaviourisflexibleandrequiresconstantremindersintheformofaudio-visualmessagesformaintaining asafedistanceduringcollaborativeworkwithhigh-speedautonomousmachines.Therefore,highprecisionalgorithms are required for depth estimation to send control signals to robots and other autonomousmachines toavoidaccidentsduetoerroneouscomputations andmovements.

Securityrobotscanhelpindetectingadherencetocovidprotocolslikewearingofmasks,socialdistancingetc.They captureimages,identifyobjects,generatesreportsandcommunicateusingmultimodaltechnologies through audio, videoandtextualdata.

Figure Figure77presents the various technologies at various domains and the benefits which helps inautomation, resource control and intelligence-based decision making based on big data analytics. Though,various technologies have contributed to the growth of modern cyber physical systems which have led toan evolutionary process resulting in smart, self-aware and self-healing systems but still the present daycyberphysicalsystemsfacemanychallengeswhichhaveanadverseimpactonthehealthofcyberphysicalsystems affecting its devices, their communication and collaboration resulting in human intervention forreinstatingtheworking stateof system.

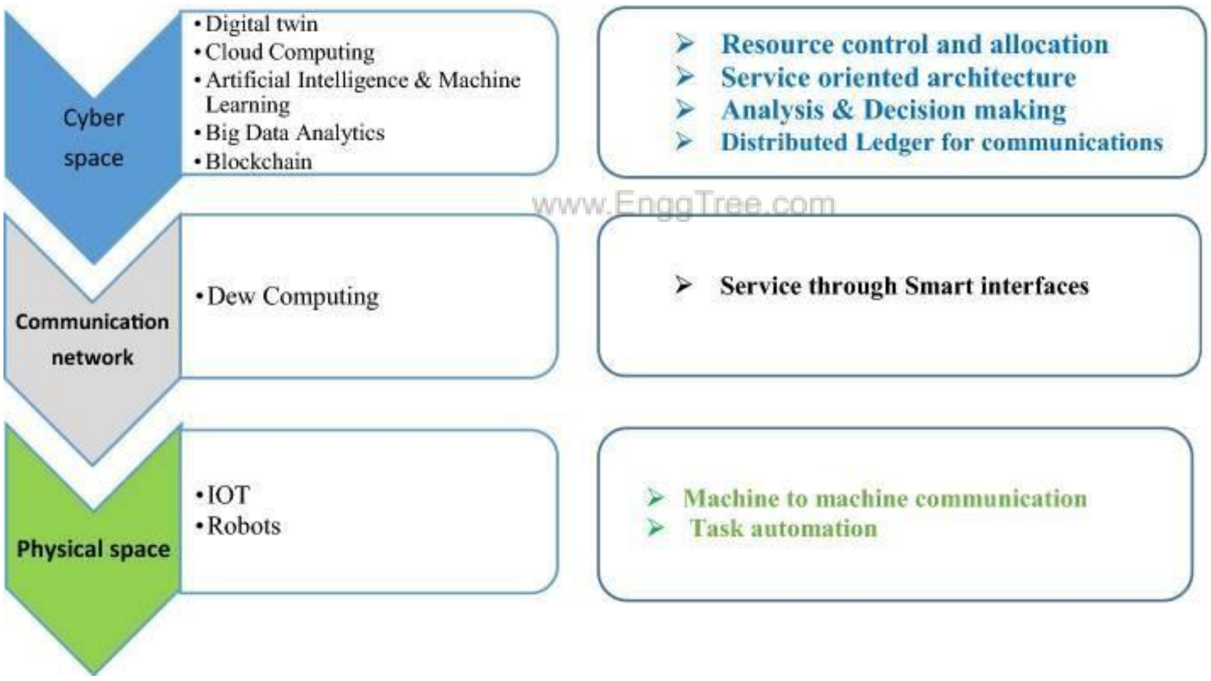

Cyber space
• Digital twin
• Cloud Computing
• Artificial Intelligence & Machine Learning
• Big Data Analytics
• Blockchain
➢ Resource control and allocation
➢ Service oriented architecture
➢ Analysis & Decision making
➢ Distributed Ledger for communications
www.EnggTree.com
Communication network
• Dew Computing
➢ Service through Smart interfaces
Physical space
• IOT
• Robots
➢ Machine to machine communication
➢ Task automation

Fig.7

Section5,presentssomechallengesforasmartcityhealthcarecyberphysicalsystemlikeenergyflowinCPS, integration of diverse devices/levels, the delay or latency which can affect the closed loop systemandthecyber-attacks whichhavesurfacedforinformationtheftanddisruptionofdevices.

ChallengesinSmartCityHealthcareCyberPhysicalSystem

There are numerous challenges for a successful implementation of HCPS like energy

flow,integrationofheterogeneousdevicesandatvariouslevelsofHCPS,minimumacceptabledelayfortimecriticaloperationsandsecurityrisksinhealthcarecyberphysicalsystemsasshowninFig.8.

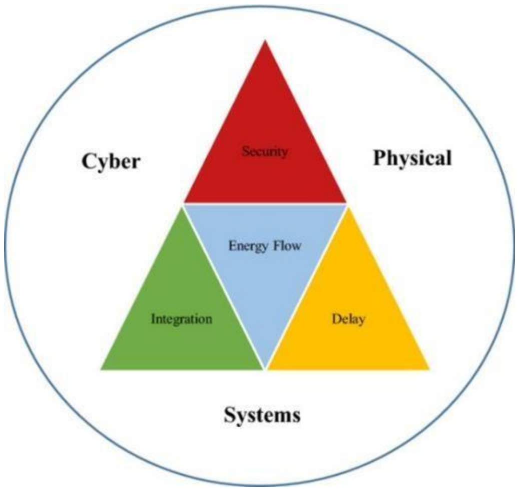


Fig.8

EnergyFlowinCyberPhysicalComponents

Theautonomyofcyberphysicalsystemsandtheirapplicationsincriticaldomainlikehealthcaredemandacontinuous flow of energy either from high-powered batteries or from direct power supply. During thesystemdesignandmodellingofHCPSitbecomesimperativetoestimatethevariouscomponentsandtheirenergy requirement so that a continuous flow of energy is maintained for smooth functioning of thehealthcare cyber physical system. Resource constraint devices with limited battery power is a challengingissue and various energy saving mechanisms exist to manage energy efficiently like the smart sensors [20]withinthesedevicesadapttotheneedofenvironmentandchangetheworkingmodeaccordinglybutgivesarippleeffectonenergyqualitytrade-off.ThesuccessfulimplementationofCPSanditsintegrationtoformasmartcityhealthcareCPSdependsprimarily ontheSmartGridwhichisabackboneofenergytothecyberphysicalsystems.Thesmarthome,smarthospitalsandsmartmanufacturingunitscanaidtheSmartgridbyinstalling solar panels and contribute as energy production units in order to meet the energy requirementswhichhaveincreased exponentiallyincyber physicalsystems.

IntegrationofHeterogeneousDevices/LevelsofHCPS

The smart city healthcare ecosystem requires the integration of cyber physical systems which depends onthirdpartyserviceprovidersforcommunicationserviceandthird-partycloudserviceprovidersforsafedatastorage which can be accessed by medical experts, insurance companies, patients and researchers. At theinternal level, each cyber physical system has heterogeneous nature of devices with different data formatswhich requires integration and encapsulation of the devices for efficient and effective environment. Themajor challenge in a heterogeneous environment will be the need of interfaces to communicating deviceswith different technology level in terms of hardware and software resources. The integration and higherlevels of HCPS require data sharing and collaborative functionality which demands system level planning,designand prototypetesting [26].

Delay/Latency

Lowlatencyisacriticalrequirementforrealtimeapplicationslikehealthcare.Thedelaysaboveathresholdvalue in transmission of patient data are unacceptable and lead to disrupting the telemonitoring cycle ofcyber physical systems affecting timely medication and care to the patients. Also, it is significant in realtime applications like healthcare that the fault latency which is a measure of time delay between theoccurrence of fault and its recognition must be a small value so that its timely management could enhancethereliability ofthesystem[23].

Security

Security of healthcare cyber physical systems is a critical issue and needs to be addressed in view of thedifferent types of attacks like denial of service, replay, false data injection and deception attacks. Theauthors in [40] have proposed a tree-based attack model for cyber physical systems where the branchescategorisetheattacksinvarioussubdomainslikefaultsignalinjectionandhardwaretamperingofsensors, packetreplayattackandinformationtheftincommunicationchannelsofclosedloopwithsensors,controllers and actuators; equipment failure and software malfunctioning in computing resources. Criticalinfrastructures like smart healthcare systems require cyber security systems, which monitor continuouslyfor identifying fault injections in system that lead to incorrect working of equipment and faulty readings[24].

Smarthealthcarecyberphysicalsystemsarefuturedigitalsystemswhichmustbeforensicreadytodealandcounter with cyber security attacks [19, 41]. Artificial intelligence or machine intelligence have wide rolein autonomous systems performing tasks like abnormal behaviour detection due to faults or cyber-attacks[42]. Malware detection is an important task for smooth and efficient working of cyber physical systems.Manymachinelearningapproachesbasedonsystemcalls,operationcodesandenergyconsumptionpatter nsareused to identifythemalwares[43].

Along with challenges related to digital world, the healthcare systems are facing the major physicalchallenge of smart waste management. The hospitals are generating the waste at an alarming rate whichneeds effective management strategies includingits collection at generationsites,transportation andhandling techniques for maintaining the health of city environment. The recent pandemic has seen the riseofPPEkits and useofdisposablemasks which needssmartwastemanagementplan.

The Healthcare Cyber Physical systems are currently working at unit or integration level, with continuousgrowth of technologies they will evolve into higher level CPS. Smart manufacturing systems that producevaccines or medical equipment are advanced level CPS with subsystems having characteristics of self-awareness and self-management, and can set to self-configuration mode to optimize the various real timeproductionprocesses[20].

Conclusion

The pandemic has taught us that healthcare systems are the lungs of every society and a smart healthcarecyber physical system provides an ecosystem which merges the two wheels cyber world and the physicalworld connected by a closed loop and steered by various technologies like digital twin, IOT, cloudcomputing, artificial intelligence, machine learning and big data analytics which play a major role in itseffective functioning. The working and implementation in physical world controlled by cyber space facemanychallengeslike heterogeneousnature ofphysical components,incompatibledata formatsexchanged between components, resource constraint devices and vulnerability of devices to attacks. Apart from thedigital challenges, the physical challenge of waste management still holds its critical place. Though, thebirth of new technologies contributes to smart interconnected systems but the challenges of physical andvirtualworldmustbeaddressed foritsgrowth, efficiencyand effectiveness.

AnIntelligentIoTBasedHealthcareSystemUsingFuzzyNeuralNetworks

Healthcare facilities in modern age are key challenge especially in developing countries where remote areas face lack of high-quality hospitals and medical experts. As artificial intelligence has revolutionized various fields of life, health has also benefitedfrom it. The existing architecture of store-and-forward method of conventional telemedicine is facing some problems, some ofwhich are the need for a local health center with dedicated staff, need for medical equipment to prepare patient reports, timeconstraint of 24–48 hours in receiving diagnosis and medication details from a medical expert in a main hospital, cost of localhealthcenters,andneedforWi-Ficonnection.Inthispaper,weintroduceanovelandintelligenthealthcaresystemthatisbasedonmodern technologieslikeInternetofthings(IoT)andmachinelearning.Thissystemisintelligentenoughtosense andprocessapatient’sdatathroughamedicaldecisionsupportsystem.Thissystemislow-costsolutionforthepeopleofremoteareas;theycanuseittofindoutwhethertheyaresufferingfromaseriousshealthissueandcureitaccordinglybycontactingnearhospitals.Theresultsoftheexperimentsalsoshowthattheproposedsystemisefficientandintelligentenoughtoprovidehealthfacilities.Theresultspresentedinthispaperaretheproofoftheconcept.

1. Introduction

Internet of things (IoT) is a network in which many devicesare connected, and these devices can communicate by computernetwork[1].Bythisworldwidenetwork,wecangetinformationthroughsensorswhichrelatetoit.Byusingcomputernetwork,wecanaccessthisinformationanywhereinthisworld.InternetofthingscanconnectphysicalobjectstoInternetandcanprovideopportunityofbuildingsystemswhicharebasedonvarioustechnologiessuchasnearfieldcommunication(NFC)andwirelesssensornetwork (WSN).Inwirelesssensornetwork,sensorssensethe environmentandsendinformationtobasestation.

IoT has different methodologies such as smart dustbin, monitoring environment, IoT based irrigation system, smarthealthcare system, and traffic control. In healthcare system, IoT brings gadget for monitoring health [2]. Health data can beaccessed with the help of IoT by using sensors. Healthcareis a system which is used to improve health and help in treatingdiseases[3].

Health related issues/complications are increasing day by day, among which lung- and heart-related issues are top-listed.Health can be monitored by wireless technology,which is a modern concept. In wireless health monitoring systems, differenttechnologies are used, including wearablesensors, portable remote health system, wireless commu- nications, and expertsystems. Life is precious; even a singlelife is also valuable, but due to lack of health facilities, awareness about diseases, andproper access to healthcare systems, people are dropping their lives. In all situations, Internet of things (IoT) helps in theindicationofdiseasesandtreatmentofpatients[4].

In IoT healthcare system, there exit wireless systems in which different applications and sensors are attached to patients,informationisobtained,andthisinformationisforwardedtoadoctororspecialistthroughanexpertsystem [5].Medicaldevicesfor Internet of things (MD-IoT) are remotely accessed, where devices are connected to the In- ternet and sensors, actuators, andother communication devices can monitor patient health [6]. Through these de- vices,thepatientinformation anddataaretransmitted bythe expert system via gateway onto a secured cloud based platform where the information is stored and can beanalyzed.

In developing countries like Pakistan, telemedicine is used to handle health issues. Telemedicine refers to the practice ofcaring for patients remotely when the provider and patient are not physically present with each other. Telemedicine is simplydefined as “the remote delivery of healthcare services.” Although telemedicine brings with it many benefits, it has somedownsides as well. Providers, payers, and policymakers alike know that there are some gray areas that are difficult to keep upwith. While the field will grow exponentially over the next decade, it will bring with it both practical and technologicalchallenges.

1.1. *Unclear Policies.* Because technology is growing at sucha fast pace, it has been difficult for policymakers to keep upwiththeindustry.Thereisgreatuncertaintyregardingmatterslikereimbursementpolicies,privacyprotection,andhealthcarelaws. Inaddition,telemedicinelaws varyfrom statetostate.

Therearecurrently29states withtelemedicineparitylaws,whichrequireprivatepayerstoreimbursetelemedicineservicesin the same way they would reimburse in-person visits. As additional states adopt parity laws, private payers may institutemore guidelines and restrictions for tele- medicine services. Although it is a step in the right direction,there is still uncertaintyregardingreimbursement rates, billingprocedures,andmore.

1.2. *Fewer Face-to-Face Consultations.* Several physicians and patients are finding it difficult to adapt to telemedicine,especiallyolderadults.Physiciansareveryconcernedaboutpatientmismanagement. Whileadvancesinmedicinehavemadeit more efficient to use technology, there are times when system outages occur. There is also the potential for error astechnologycannotalways capture whatthehuman touchcan.

1.3. *Technology Is Expensive.* Healthcare systems that adopt telemedicine solutions can attest that they require a lot of time andmoney. Implementing a new system requires training, and sometimes staff members find it difficult to welcome this change.Practice managers, nurses, physicians,and more have to learn how to utilize the system so that practices can see the benefits.Althoughtelemedicineisexpensiveinthebeginning,healthcaresystemsshouldseeapositivereturnoninvestmentovertimeduetomorepatientsandlessstaff.

The major components of healthcare systems are identification, location, sensing, and connectivity as shown in Figure 1.Smarthealthcareisimplementedthroughawiderangeofsystems:emergencyservices,smartcomputing,sensors,labonchips,remote monitoring, wearabledevices,connectivitydevices,andbigdata.

The IoT based systems are equipped with body sensor networks within telemedicine systems. They include deviceswithspecialtypeofnodesthatsenseperiodicdifferenceofpatientdata;tochecktheventilationcondition sforthepatientsinrooms,sensors are used to collect data for dif- ferent measures contributing to ventilation process of a room. These sensors areprogrammed to assess data of different ranges for temperature, pressure, humidity, and othersignificantenvironmentalvariables.

Thesearrangementshelptomonitorthepatientcon-ditions remotely. The system can send periodic reports to thehospital and maintain the patient history. The hospital staffcan view the data and prepare the treatment plan for thepatientunder observation. The second type of devices usedin IoT healthcare systems is based on wireless sensor net-works. Thesituation is more complex than the above sce-nario in terms of remoteareapatient monitoring andmanagementtask. Insomesituations,IoTisthemostre-liableandcheapestsolution,andtherelationshipbetweendifferent devices andinteractive communication systemsalso needs to be investigated with more formal objectives. Technologymakesiteasiertomonitorthepatient health by sending information to healthcare teams such as adoctor, nurses, and specialists throughIoT (Internet ofthings)andmobiletechnologies.Itwouldbehelpfulforprofessionals to save and gather patient datausing store-and-forward method so that it is accessible at any time. Therole and services of IoT in modern healthcare aredepictedinFigure2.

Internet of things (IoT) has different methodologies:smart healthcare, traffic control, smart dustbin, and vehicle parking.Thehealthofpatientismonitoredbyscreen,soitisdifficulttoexaminethepatientallthetime.Therefore, here,patient’scurrentstatus,i.e.,pulserate,temperature,positionofbody, bloodglucose,and ECG,canbe measuredinten-sivelybyusingsensors.

ThesensorsareattachedtoArduinoUNOsensorsthat,whenattachedtothebodyArduinoboard,getinformationa ndtransmititto theserver.From thisserver,theinformation isforwarded to thedoctorwhoadvisesformedicine.

Smarthealthcaresystemisactuallyatechnologyinwhichtreatmentsofpatientsarepossibleandcanimprovet hestandardof life [7]. In the concept of smart health, the e-health concept is also included which has commands on many technologieslike electronic record management, smart home services, and intelligent and medical connected devices. Sensors, smartdevices, andexpertsystemssupportthehealthpracticeforsmarthealthcaresystem.

Healthcare facilities in modern age are key challenge especially in developing countries where remote areas face lack ofhigh-quality hospitals and medical experts. As ar- tificial intelligence has revolutionized various fields of life, health has alsobenefitedfromit.Theexistingarchitectureof

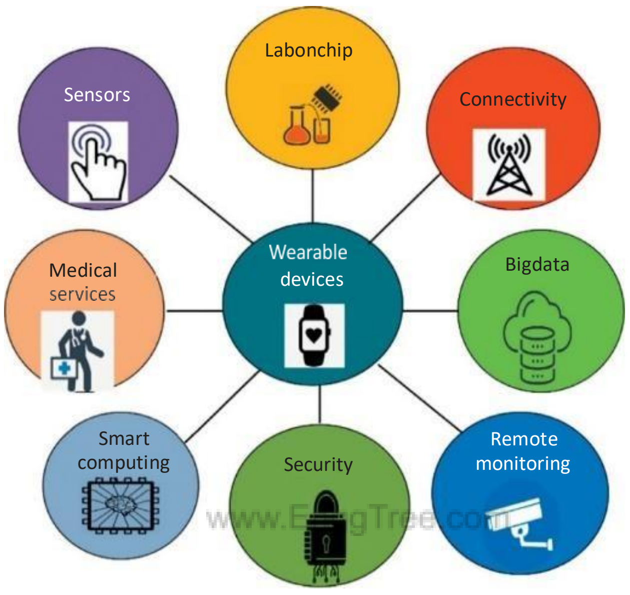


FIgurE1:TypicalcomponentsofIoTbasedsmarthealthcare.

combination of sensors [8], imbedded electronics, and software components for decision support systems [9]. The IoT alsosupportssmartsystemswiththehelpoflightweightnetworking connectionsandsensorsdata[10].TheIoTcoversalmostalldomains such as home based smart systemslike security, entertainment, and health; transport systems like smart parking,traffic, logistics emergency services, andhighway management; community based systems like smartmetering, businessintelligence,surveillance, environment,andretailsystems[10].

Healthcare is one of the primary concerns that need to beimproved by IoT and related technologies [11]. IoT basedhealthcare consists of three stages of automation, namely, data collection by sensors, analytics based on the collected data,anddecisionmakingbasedonthecollecteddata[12].Healthcaresystemshavealotofpotentialtobeimpro vedbyIoTbasedsystems.Therearemanydifferenttypesofhealthcareapplicationsproposedbytheresearchers likee-health[13],communityhealth[14],homehealthmoni- toring[15],telemedicine[16],andclinicalsupportfordoctors[17].Theprimarycontributionofthisresearchist oprovidedevicesthathelpinmonitoring,management,andcommunicationbetweendifferentstakeholdersin thehealthcaredomain.

Table1showsthecomparisonofourworkwithrelated work.ItshowsthatIoTbasedmodelsprovidemuchas- sistancetopatients,butthetimeconstraintscanbereducedwiththehelpofCDSSintheabsenceofmedicalstaff.

The methodology of IoTfor smart home, vehicle parking, andtrafficcontrolisdifferentascomparedtothehealthstore-and-forwardmethod ofconventionaltelemedicineisfacingsomeproblems:

(i) Needforalocalhealthcenterwithdedicatedstaff.

(ii) Needformedicalequipmenttopreparepatientreports.

(iii) Timeconstraintof24–48hoursinreceivingdiag- nosisandmedicationdetailsfromamedicalexpertinamainhospital.

(iv) Costoflocalhealthcenters.

(v) NeedforWi-Ficonnection.

Inthispaper,anovelandintelligenthealthcaresystemisproposed;itisbasedonmoderntechnologieslikeInter netofthings(IoT) and machine learning. This system is intelligentenough to sense and process a patient's data through amedical decisionsupport system. This system is low-cost solution for the people of remote areas; they can use it to findout whether they aresufferingfromaserioushealth issueandcureitaccordinglybycontactingnearhospitals.

2. RelatedWork

The IoTterm was initially coined in 1999 and got attention ofcommunity asoneof the advanced technologiesasit is abenefitistomonitorthepatients24/7[26],whichisalmostimpossiblewithmanpower.ThesecondgoalachievedbyIoTbasedsolutionistomonitorprimarymeasuresneedstodeterminethe patientconditions,andtreatmentplanmayincludepulserate,bodytemperature,respiratoryrate,bodyposition,bloodpressure,ECG,andglucoselevel.ThesesensornetworksareconnectedthroughArduinoboardtocollectthe informationthroughattachedsensors.Thecol-lectedinformationcanbetransmittedto theserverandfurtherrefined fordecisionmaking ordecisionsupport systems.

Investigational experiments are made with the assistanceof sensors, and the patient's health is traced with Internet. Whatremains is the keen observation of pulse rate, eco of heart, pressure level, temperature, etc. If there is any dis- turbance orchange in pulse rate or temperature, the system alerts the person taking care of the patient. Through the Internet, the systemshowsthepulserate and temperatureofthepatient.

The IoT with mobile technology provides smart andeasier ways to look after the patients under observation, theirbodymovements, and health conditions and provides in- telligent mechanisms to handle and share the relevant in- formation withrelevantstakeholders.Thestudy[11]designedasystemwhichcollectspatientdataandsendsittocloud forfurtherutilizationby people investigating health domain. The multipurpose application may also provide thefamilies of patients with regularupdatesregardingpatient health. Ghosh et al. [26] demonstrated a system to auto- matically gather data from patients and store the gathered data intocloud for permanent use to help health profes- sionals. The system also helps the guardians of the patients to know the healthinformation.

The study [27] proposes a system to track the patient records with the measures of pulse rate, ECO, blood pres- sure, andbody temperature and maintain the patient his- tory. If the system detects any abnormal behavior in the measures observed, itimmediately alerts the emergencyteam to handle the situation. The article [28] provides a survey on the smart healthcare. Itdiscusses in detail the importance, application, requirements, and classification of healthcare along with the challenges,vulnerability, and se- curity attacks. Healthcare system plays a vital role in in- creasing application by using connectivitytechnologies. The body sensor asamedicaldeviceisused toimplementsmart healthcare as shown in Figure 3. Smarttelemedicinesystems [15] are designed to monitor and manage the patient recordsby using sensors and microcontrollers. The system observesthebodyconditionsandtransmitsthe datatocloudservers.The patientconditionisobservedandstoredonserversforfurtheruseanddecisionmaking.

The study [29] investigates the challenges and conse- quences of remote health systems. The system comprises wirelesstransmission system which collects ECG, bodytemperature, and pulse rate of the patients in remote lo- cations for severeproblemslikecardiogenicshock.Thepatientismonitored,anddataissenttothedoctortoanalyzethemandpreparethetreatmentplan.Thisdataalsohelpsthesupportingstafftotakethenecessaryactions[30].

The study [31] states that health monitoring system is essential for a good health because health problems areincreasingdaybydaylikecardiacfailure,lungsfailure,andheartrelateddiseases.Nowadays,IoTbecameaplatformformanyservicesandapplications in which sensor nodes are used. The monitoring of patients that is continuously done by doctors is the base ofrevealedconsequencesofgenerichealthmonitoringsystem.

The data analytics with big data enhanced the capacitiesof healthcare management system. The IoT healthcare is based onsensors, data collection devices, cloud services or connectivity provider devices, and mobile applications. Themain concernof the physician is to separate the informationof one patient from all other massive information of patients in the healthcaresystem.Fromsuchhugeinformation,thephysicianmakescriticaldecisionaboutthepatienthealthandsuggeststhetreatment.He etal.concluded thatalteringpatientinformationinrealtimeisveryimportant[32].

In order to build a smart system or application, thephysical objects are connected by using IoT

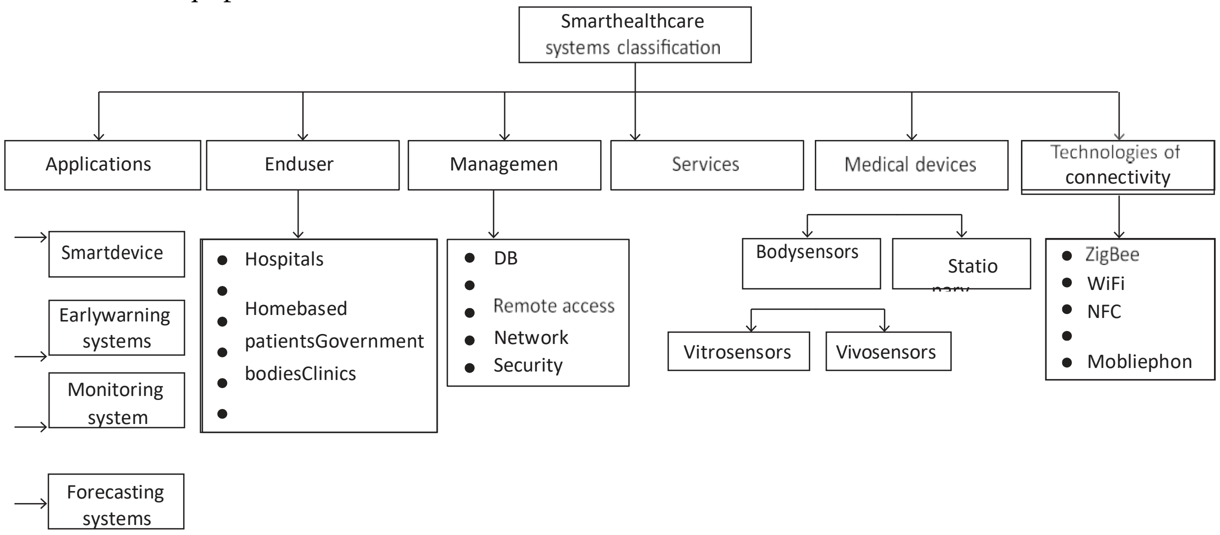
(Internet of things). Thestudy used IoT for smart resource managementsystem (SRMS) and intelligent chair system (ICS). An Arduino board isattached to the sensors, user ID is con- nected through RFID reader with the chairs, and chair al- lotment is managed andmonitoredbythissystem.

The study [33] investigates the influence of medicalsystem with remote patients. The patient monitoring is the mainpurpose of the system with a prototype application. The main service provided by this prototyping system is themonitoringof vital signs of patient health in ICU. Thesystem is more effective for patients undergoing surgical procedures or othertreatmentsthatneedintensivecareandmonitoring.Themajorbenefitsclaimedbytheauthorsofthestudyar elowpowerprofilesensor, wireless communication,and gateway for cheap communication. The system is also available on web domain for thepatientcaretakersto ob-serveandbeinformedaboutthepatientstatus.

A smart health monitoring system [34] is designed and implemented for ambulance coupled with communicationchannels. The IoT was used with this smart healthcare system with the capability of low power sensors. The humanbodysensorsareconsidered anefficientmodeofcommu- nicationfornearfieldbodysensornetworkapplication.

The study [35] presents a generic model for IoT based healthcare system. The model identifies key components with anend-to-endIoThealthcaresystem.Theauthorsclaimthattherewasnoend-to-endIoTbasedremotepatientmonitoringsystem.Thesystemconsistsoffivesensors,threeofwhichwereformoni toringpatientconditionslikepulserate,respiratoryrate,andbody temperature. The other two sensors were used to monitor blood pressure and bloodoxygen. The paper also identifiestechnologicalchallengesandpotentialopportunitiesforremotehealthcaremoni- toringandmanagementsystemwithIoT.

Thestudy[36]presentstheapplicationsandpotentialusageofIoTinhealthcaresystems.Themajortaskper- formedby thisexperimentalstudywastomonitorthepatientconditionsandmakeitpossibletousemoreoptimizedand accuratemedicalequipment.Thebasicarchitectureof



Classificationofsmarthealthcaresystem.

the proposed system was based on sensor data and analyzesthe patient data to make it possible to take basic decision forthepurpose.Theproposedsystemwascollectingbodytemperature,respirationrate,andheartbeatsandobserv ingthepatientbodymovements.

Wearablebodydevices[37]wereusedtodesignsystemsformonitoringandmanagement ofpatientshealth. Thestudyusedbodywearablesensorswithl oTforsmartpatientmonitoringandmanagement.Thefocusofthisstu dywastoobservethepatienthealth during surgical procedures, considering more useful and reliable data collection during such complex situations wherehumanobservationalskillsarenotenough.Thesecondmajorbenefitwasthereductionofequipmentsizefo rpatientmonitoringand wireless en- vironment for patient care. Devices like Fitbit health monitor, Pebble smart watch, and Google glass are con-sidered as modern devices for health monitoring and body care solutions. The important wearable devices are mea- suring theblood pressure which is used to assess the stress on a human mind. IoT imparts a valuable role to electronicsand electricaldevicestomonitorand managehealthcareforhumankind.

3. ArchitectureofSmartHealthcareSystem

The proposed smart healthcare system has the capability of decision making as per the observed conditions of the pa- tientbased on body temperature, pulse rate, and heartbeats. This architecture is also energy efficient solution because it does notturnonallthe sensorsallthe time.The algorithmusedinthesystemwillhandle the usage ofthesensorsandcontroltheircostand lifetime. The proposed system ad- dresses the issue of remote monitoring of patients and provides them with necessarytreatmentthrough expertsinthehospital.

Thesmarthealthcaremonitoringandpatientmanage-mentsystemproposedinthisstudyconsistsofcommuni- cationchannels,embeddedinternaland externalsensors,IoTserver,andcloudstorageandissupportedbyagateway.

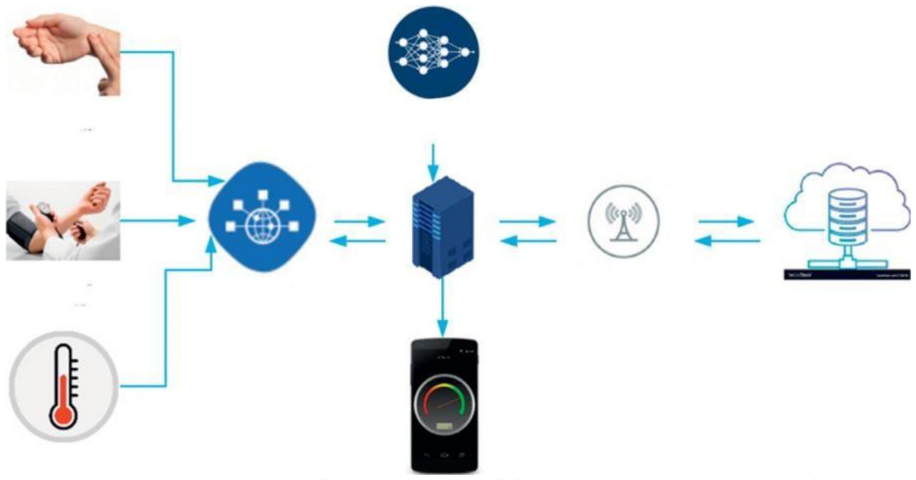
These activities are performed at different levels of refine- ment named application layer, management layer, network layer,anddevicelayer. ThearchitectureoftheproposedsystemispresentedinFigure4.

The architecture shown in Figure 4 is revised to show more details. The use of sensors and decision support systemintelemedicineisanovelideathatimprovesworking performanceoftelemedicineinruralareas.

3.1. *Data Collection through Sensors for Smart Healthcare System.* With the help of IoT (Internet of things), the proposedsystem will be designed to implement a device in remote clinic. The device will take data of patient’s heart- beats,bodytemperature,andbloodpressureasinputand willsenditttothedoctorconcernedinthehospital.Withthehelpofthedata,thedoctorwillanalyzetheconditionoft hepatientandwillinformtheremoteareacliniccrewaboutthenecessarystepsforpatient’sbesttreatment.

The architecture presented in Figure 5 shows physical view with necessary components of the proposed system. Thesystem consists of three sensors: body temperaturesensor, pulse rate sensor, and heartbeat sensors. These threesensors areconnectedthroughArduinoboardtocollectandclassifythepatientdata.Thedatatransmissionismanaged bycommunicationandnetworkingdevices.Thedataana-lyticsprovidesthedecision-makingfacilities,andthefuzzylogicsystemisusedinthisarrangement to provide decisionmaking. The doctor view provides the facility to hospital staff to monitor and communicate withthepatient atremoteplace.

The next subsection explains the fuzzy logic system implemented in this smart patient monitoring and man-agementsystemfordecisionmaking.Thefuzzysystemisplacedattheserveranditwillorderthedecisionsregardingpatientconditionsand treatment and alert the doctor about the situation of the patient. The system is fully automated. The last subsection givesthetechnicaldetailsanddescriptionoftheproposedsystem.



3.2. *Fuzzy Logic-Based Smart Healthcare Monitoring and Management.* There are the following problems: a single model isnot enough, so two or more models are combined to solve that problem [38]. When different models are combined, theyprovideaneffectivesolutiontotheproblem,referredtoasahybridsystem.Ahybridsystemisusedtoobtainindoorairquality usingthefuzzylogicsystemand neuralnetworksrepresentedasthefuzzyneural network(FNN).

Neural networks focus on perceiving patterns, not on thelogic of how the decision is made [39]. The fuzzy logic systemsare good at explaining how the decision is made, butthe inference rules are difficult task as prior knowledge is required [40].Theselimitationsleadtothefuzzyneuralnetwork.Rulesoffuzzysystemsareacquiredfromtheneuralnetworkspatterns[41].

Thisprocessbeginswitha "fuzzyneuron," andtheprocessofthefuzzyneuronisdividedintotwostepsasfollows[42]:

(i) Evolutionofafuzzyneuronmodel.

(ii) Developmentofthemodelanditsalgorithmthatconsolidatefuzzinessintotheneuralsystem.

Figure 6 indicates that neural inputs are provided for neural network that provides neural outputs. Neural outputsare theinference rules for the fuzzy interface that are storedin the system as a database and used for decision making andprovidelearningalgorithmsfortheneuralnetworkaspriorknowledge.Dataofneuralnetworksisgatheredby propa-gationalgorithm,sotheprocedureisslow.Includingspecificdataintothe neuralnetworktoclarifylearningtechniquesisadifficulttask.Fuzzyrules are explained, and they provide better performance, so fuzzy systems are used in restricted systems and knowledgeacquisitionis a difficulttask.Tosolve these problems insolutiondesign,the fuzzyrulesaredesignedfromnumericaldata.

The neural network model named Approximate Rea- soning Intelligent Control (ARIC) (see Figure 7) uses fuzzyneuronsystem. This fuzzy neuron system is trained by physical system forecast. It applies a fine-tuning refreshing data method tocontroltheinformationbase.

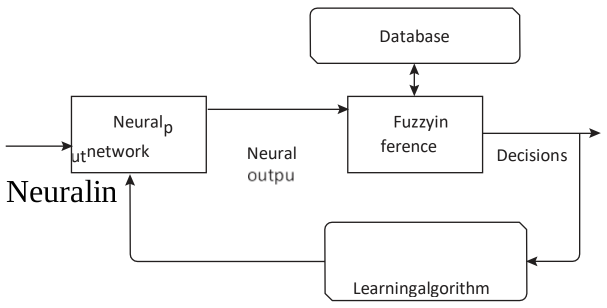


FIgurE6:Modeloffuzzyneuralnetwork

ancestor and acquires a unit *u*. The unit *u* communicates with the activity control. It is called defuzzified mixture whichfinishes the process.The informationlayerinthisframeworkisfuzzified,itismonotonicinnatureandhasthecapacitiestoutilize its components in ARIC model. The fuzzy tag is used in rules to balance local standards. The ancestor's enrolmentis estimated by these standards and then duplicated and uses load joining with the association ofinformation component.The qualities base of this system produces the final input. Each unit which was obscured is exceptional monotonic workcommunicating with final standard. The monotonicity of this function yieldsthe output. The process is effortlesslydetermined by the op- posite capacity. This esteem is produced with the function ofheaviness and with the association ofhiddenunit. Theyieldvalueisfinallydeterminedbyweightedaveragemethod.

Theactionoperatorsusedtoevaluatethenetwork,whichtriesto forecastthe model activity.Theneuralnetowrkmethodused in this system is a typical feedforward neural network system. This feedforward neural network system isbased onshrouded layer which collects the model states as information. It uses the blunder flag *r* from the physical model as a pieceofhelpingdata.Theprocessgets*vt*, *t*ofthe proposed system produced as a forecast for future. Thissystem relies on loadoftime*t*andthemodelconstraints.The *t*iseither*t*or*t*+1.Theconditionsinthissystemare

[ ]

portrayed by fortification of higher values for information collected for decision making. The change in load is man- agedby support method that uses the output of the system states of the network and action state evaluation method. TheengineeringofARICwasconnectedtopostadjustment.Itisalsodemonstratedthatthemodelwiththecompre-hensionanditsassignments.

Thesignsandweightsarerealnumberswithinputneurons.

This is a perfect combination of fuzzy system with neuralnetworks which boost the advantages of both decision-makingmethods.Theframeworkcanlearn,andinformationutilizedintheframeworkhasthetypeofif-thenfuzzysystem.Therulesaredefined in advance, and the system canstart without outside help, so it adapts quicker than a standard neural system. TheframeworknamedARICconsistsofAENactionpositionwhichisusedtoevaluatethenetworkconstructedthroughinformationbase.TheARICalsocontainsASNoperationusedfornetworkselection.Itisamultilayerneuralnetworktechniquewithfuzzycontrol system. The ASN component has two separate fuzzy in- terfaces in the first layer of the proposed system. The neuralnetworkisplaced in thesecond layer. Theneuralsystemfinds$f[a,a+1]$operation,apartofconfidenceacquired throughfuzzy inference. Itshould gain$p$ $(a+1)$using theamountoftimedenotedby$t$andtheconditionofframework$t+1$.Amodificationmodulewhichisstochasticinnature improvesthecontrolwithp(t)offluffypartandtheexpectedlikelihoodregradingdecisionsandproducesthefinaloutput.

$$u'(t) = r(u(t), d[t,t+1]). \qquad (1)$$

The unit$c_i$offuzzyinference isorganizedtoassessthefuzzyguideline.Theunitforinformation$a_j$isastandard Theinformationdoesnotchangethesesigns.Theyieldisverymuchequivalenttotheinformation.Thesignal$a_i$maycol-laboratewiththeloadto$w_i$toconstructtheseitems.

$$d = w_i a_i, \quad i = 1,2. \qquad (2)$$

The inputdata $d_i$iscollected,byaddition, todelivertheinformation,

$$\text{net} = d_1+d_2 = w_1a_1+w_2a_2, \qquad (3)$$

to the neuron. The neuron utilizes its exchange work f, whichcould be a sigmoidal function$f(x)(1$

$+e^{-x})^{-1}$, to figureouttheoutput:

$$y = f(\text{net}) = f w_1a_1+w_2a_2. \qquad (4)$$

This basic neural net, which utilizes duplication, addi- tion, and sigmoidal f, will be called a(nordinary neural net asshowninFigure8.

Inthisevent,weutilizedifferentactivitieslikea$t$-norm,ora$t$-conorm,tojointheapproachinginformationtoaneuron;weget what we call hybrid neural net. These modifications lead to fuzzy neural engineering dependent onfuzzy arithmetic tasks.Thisgivesusachancetoexpressthesourcesofinfoa_1,$a_2$andtheweightsw_1,$w_2$overtheunitintervals[0,1].The

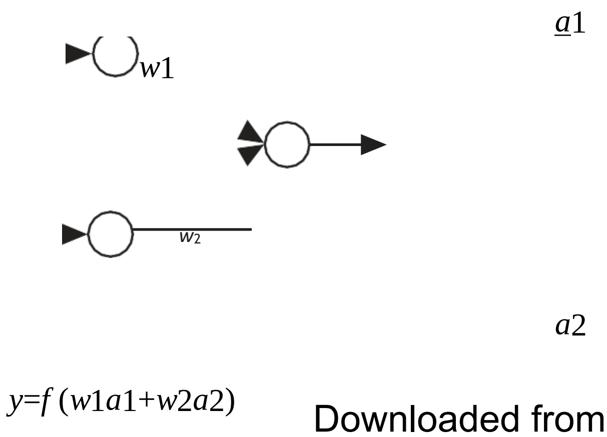


TaBlE2:Directfuzzificationofneuralnetwork.

Fuzzyneuralnet	Weights	Inputs	Target
Type1	Crisp	Crisp	Fuzzy
Type2	Fuzzy	Crisp	Crisp
Type3	Fuzzy	Fuzzy	Crisp
Type4	Crisp	Fuzzy	Crisp
Type5	Crisp	Fuzzy	Fuzzy
Type6	Fuzzy	Fuzzy	Fuzzy

FIgurE8:Neuralnet.

systemsisto broadenassociationweightsandadditionalinputstofuzzynumbersasshowninTable2.

A set of fuzzy rules were defined for the clinical decisionsupport system used for IoT based telemedicine. These rulesarebasedon the factsand fuzzydatashown in Table2.Followingareafewexamplesoffuzzyrulesdefined. �� �� ��

IF(Temperature �� High)AND(Pulse_Rate �� Low)AND(Blood_Pressure �� Very_High)
THENDecision High

IF(Temperature �� High)AND(Pulse_Rate Low)AND(Blood_Pressure �� High) ��
THENDecision High

IF (Temperature Normal) AND (Pulse_R-ate High)AND(Blood_Pressure Medium)
THENDecision Low
�

IF(Temperature Low)AND (Pulse_Rate High)AND(Blood_Pressure Medium)
THENDecision Low
�

IF (Temperature �� Normal) AND �� (Pulse_R-ate �� Normal)AND(Blood_Pressure Low)
�� �� ��
THENDecision High

3.3. *Implementation Details.* A microcontroller board (Arduino) (see Figure 9), which has model numberATmega328, has 4digitalpinsforinputandoutputsources.Thesixi/opinsarePWMoutput.Themicroprocessorhas16MHzwithapowerjack,USB connection. The other components on this microcontroller chip are analog input and reset button with ICSP header. Thepower is supplied bya USB interface, and Arduino is designed as open electronicplatform. The basic settings on Arduino boardareinput/output,set/resetbutton,sensorlights,andactivating motorwithoutputLED.

HC-05Bluetoothmodule:Toaddwirelessfunctionalityoftwoways(fullduplex)toyourproject,HC-05isverycoolmodule.If communication is required between two microcontrollers, Bluetooth module is used as Arduino and can communicate withanydevicewiththefunctionalityofBluetoothlikealaptoporaphone.BluetoothSSP(serialportprotocol)moduleisdesignedfor wireless transport. HC-05 can be used in a master or slave configuration that will begreatsolutionforwirelesscommunication.

Temperaturesensor(seeFigure10)isusedtodetectheatstroke,bodytemperature,andfever.Inwearablehealthcare

system, body temperature is used as a diagnostic tool. For themeasurement of body temperature, thermistor type sensors areused.Temperaturesensingaccuracyislimited.

The temperature sensor (see Figure 11) is integrated circuit which is used to measure the body temperature in centigrade.Thetemperatureisshownasvoltageoutput.ThemodelnumberofthissensorisLM35.Thismodelofbodytemperaturesensorisconsideredbetterinperformancethanlineartemperaturesensor.ThereasonisthatuserneednotconvertKelvinscaletocentigradescalebyusingthismodel.ThesensorunderthissetupisveryusefulforremotesensingandcalibratesCelsiusscale.Theemergencyconditionsaremeasuredthroughcardiacarrest,pulmonaryembolism,vasovagalsyncope,andpulsesensor. Thepulserateisprimarymeasureforcriticalmedicalconditionsandbodyfitnessconditions.Thepulseratesensoristhemostused and researched sensor in patient care and managementdomain.It isused toassess heartbeats and complex diseaseslike heart attack. The sensor works when theobject places finger on input panel. The output is detected onoutput panel. The powerrequiredforthissensoris5voltdirectcurrent.Theworkingprincipleofthismoduleisbasedonbloodflowra tethroughfinger.The heartbeat sensor normal reading was 60–100 bpm. Figure 12 shows the used blood pressure sensor to measure the bloodpressure ofthepatientandrecorditinanExcelsheetforfurtherprocessing.

1. ExperimentalResults

1.1. *Experiment Setup*. The system is tested under the su- pervision of medical staff. Samples are collected from dif- ferentareas of South Punjab using the proposed device. Thedata collected through sensors was forwarded to the server. The resultsare presented at the Arduino application and webbrowser. Table 3 shows the information about locations thatwe selected totesttheproposedmodel.Almosteightdif- ferentlocationsareselectedfortesting.Thedistancefrom BVHandtestingperiodofselectedlocationsaredifferent.

1.2. *Dataset*. Table 4 shows the report sample of the patient that is generated on the server after receiving data collectedthrough sensors and forwarded through smart device. Thereport has three sections: patient’s data, sensor data, andsymptomsofthepatient.

Table 5 shows the comparison of response time of the queries that are responded to by CDSS and by physician. Almost270 queries were received on the server from se- lective areas. Most of the queries were treated by CDSS. Table 5 clearlyshowsthataverageresponsetimeofthe queriesthatarerespondedtobyCDSSisquiteshort ascomparedtotheresponsetimeofthe queriesrespondedtobyphysician. The proposed systemislow-costand efficientsolutionforthepeopleofremoteareas;they can use it to findout whether they are suffering from a serious health issue and cure it accordingly by contacting nearhospitals.

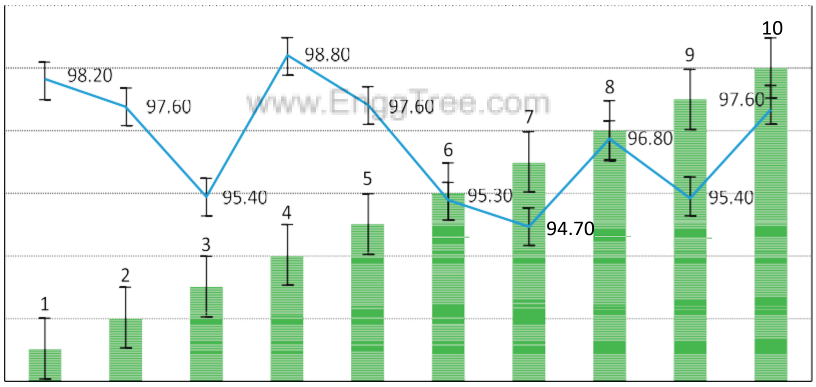
Usingsensorsanddecisionsupportsystemintele-medicineisanewidea,andTable5showshowitminimizestimeconstraintincomparison to theclassicaltelemedicinemethod.

1.3. *UsedToolsandDataAnalysis.*Theuseofanalyticspotentiallyimprovestheaccuracyandpermitsearlydiseasedetection,personalization,andcostreductioninmedical

12

1.ConclusionsandFutureWork

The proposedmethodconsistsofsensorsforbody tem-perature, pulserate,and blood pressure to assessthe con-dition ofthepatientunderobservation.Fordeterminingthepossibleconditionsandcure,thesystemusedaknowledge baseandfuzzylogicsystemforintelligentdecision makingforpatientcare,monitoring,and management.



AdaBoostwithFeatureSelectionUsingIoTtoBringthePathsforSomaticMutationsEvaluationinCancer

Abstract

Nowadays, the research in bioinformatics helps in finding out numerous ways in storing, managingorganic information, and developing and analyzing the computational tools for better understanding. Sofar, much of the research has been carried out to overcome the difficulties in experimental methods whilestoring vast amounts of the data in different sequencing projects. In this process, many of thecomputational methods and clustering algorithms were brought to light in the past to diminish blocksbetween newly sequenced gene and genotypes by applying identified jobs. The latest specific applicationsinvented in bioinformatics are paving way for more advancement by adding developments in machinelearning and data mining fields. Because of a large quantity of applications acquired by various featureencoding methods, the existing classification results remained inadequate.

Hence, the present study isintended to create awareness among the readers on the various possibilities available in finding somaticmutations by using machine learning algorithm, AdaBoost with feature selection, a classification invarious feature selection techniques with their applications, and detailed explanation on the distinct typesof advanced bioinformatics applications. This study presents the statistical metric-based AdaBoost featureselection in detail and how it helps in decreasing the size of the selected feature vector, and it explainshow the improvement can be attributed through some measurements using performance metrics:correctness, understanding, specificity, paths of mutations, etc. The present study suggests some IOTdevices for early detection of breastcancer.

A Fuzzy-Based Expert System to Diagnose Alzheimer'sDisease

Abstract

Soft computing techniques came into reality to deal effectively with the emerging problems related tomany fields. A medical diagnosis is totally based on human abilities, uncertain factors, ambiguoussymptoms, high accuracy, and bulk of medical records. Soft computing techniques are suitable to obtainresults in an efficient way in medical diagnosis. Fuzzy logic (FL) is one of the popular soft computingtechniques. FL is a mathematical approach for computing and inferencing which generalizes crisp logicandsetstheoryemployingtheconceptoffuzzyset.Fuzzylogichasbeensuccessfullyappliedinthefieldsof pattern recognition, image processing, knowledge engineering, medical diagnosis, control theory, etc.,Alzheimer's disease (AD) is the most popular dementia in aged people. AD is an irreversible andprogressive neurodegenerative disorder that slowly destroys memory, thinking skill, and degrades of theability of performing daily tasks. Hippocampus is a key biomarker for AD to identify the disease at anearly stage. To detect and diagnose Alzheimer's disease at an early stage, fuzzy logic is playing a vitalrole. In this study, computerized system for classification of AD was constructed using fuzzy logicapproach, i.e., fuzzy inference system (FIS) to classify the subjects into AD, mild cognitive impairment(MCI),andnormalcontrol(non-AD)on thebasisofvisualfeaturesfromhippocampusregion

Secured Architecture for Internet of Things-Enabled PersonalizedHealthcareSystems

Abstract

The Internet of Things (IoT) is the emerging area. This technology is made to connect any object aroundus to the Internet with the unique IP, and these connected objects can be communicated each otherremotely as per the user's convenience. It has applications in all most all the fields like industries,factories, environment, agriculture, transport, education, healthcare, energy, and retail. IoT leads to thenewtechnologieslikebigdataandcyber-physicalsystems.Connectinganyobject,fromanywhereatanytime,isnotsimple.Ithasvariouschallengeslikediscovery,scalability,softwarecomplexity,interoperability,faulttolerance,security,andprivacy.Oneofthemajorchallengesissecurity.Duetotheweak links used to connect the things to the Internet leads to many security issues in different levels oftheIoT.ThispaperpresentsthevarioussecurityissuesandnovelsecurityarchitecturefortheloT-enabledpersonalizedhealthcaresystems.**KeywordsIoT**·InternetofThings·Security·Architecture·Healthcare

7.1Introduction

Sinceadecade,InternetofThingsisevolving.TheenablingtechnologiesofIoTarethecloudcomputing,wireless sensor networks, communication protocols, big data analytics, embedded systems, etc. InternetofThingsisatrulyubiquitouscomputing,i.e.,anywhere,anytimeforeveryonecomputingsaidbyWeiser [1]. The phrase "Internet of Things" is first coined by the Kevin Ashton in 1999 at MIT. Nowadays, themobilephonesanddataratebecomeverycheaper,thewirelesscommunicationdevicesbecomingsmallerandcheaperandtheprocessingcapabilitiesismore.Sothesmartphonebecomesthemediatorforthings,Internet,andpeople[2].Ithasvastapplicationdomainsliketransportationandlogistics,healthcare,smartenvironments, Personal, Social and Futuristic applications of IOT [3]. The major relevant scenarios forthis domainareshown in Fig. 7.1

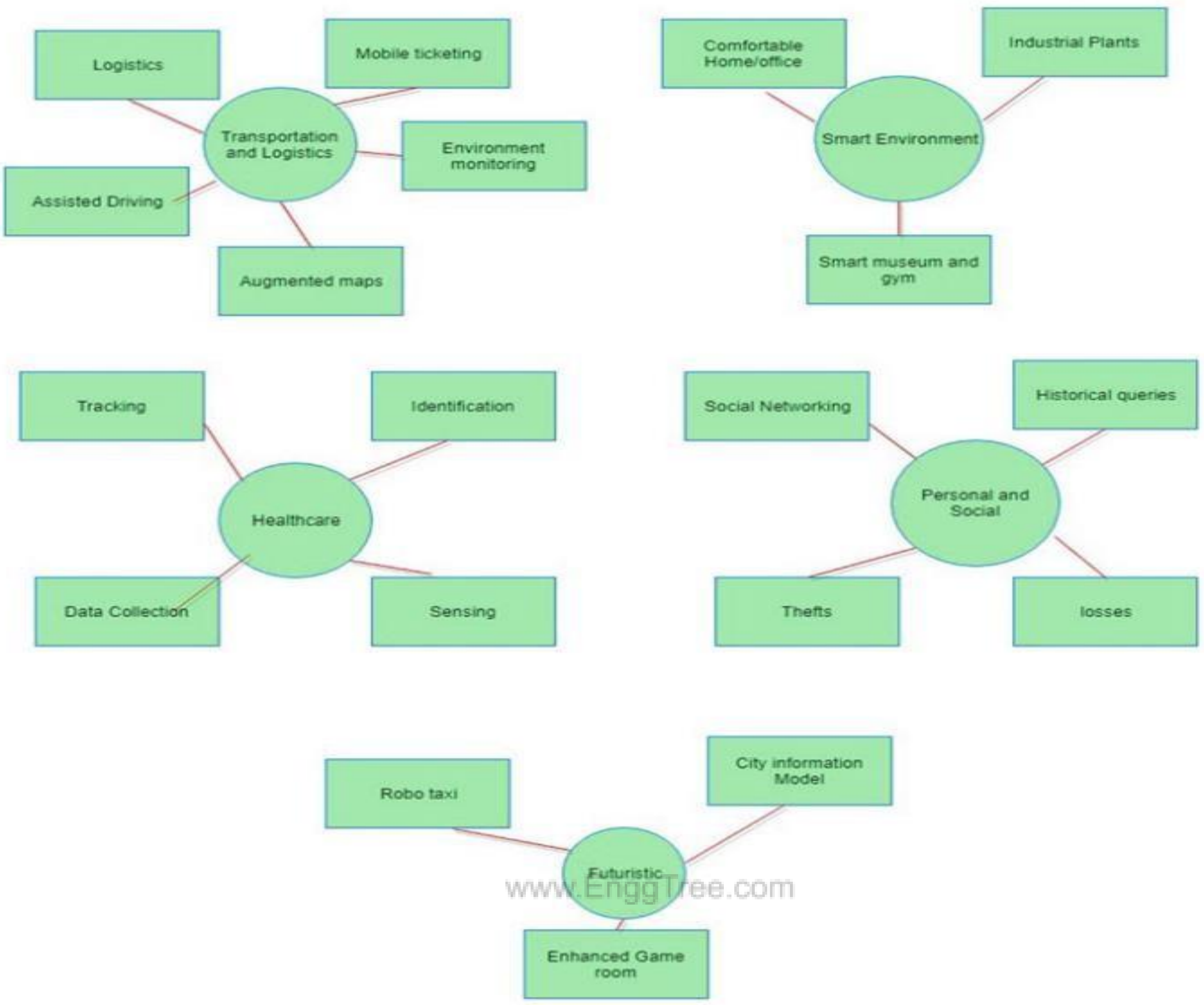


Fig. 7.1 Major relevant scenarios for various application domains in IoT

By 2025, the market share of the IoT applications is projected as healthcare 41%, manufacturing 33%,electricity7%,urbaninfrastructure4%,security4%,agriculture4%,resourceextraction4%,vehicles2%,and retail 1% [4]. The projected analysis shows that major application of IoT healthcare leads a goodmarket. Remote monitoring of the health plays a key role in the domain of the healthcare. If many IoTdevices are connected to Internet, it may lead to many attacks. Particularly in the healthcare, if any data ismisused, then the patient's life will be expired. The major attack in healthcare IoT is impersonate attack.Inthisattack,theintruderwillpretendashe/sheisapatientordoctorforcollectingthedata,etc.Thisleadstomanysecurityissues.Itis necessarytodevelopsecurityarchitectureintheareaofhealthcareIoT.

This paper proposed the novel security architecture for the IoT-enabled personalized healthcare systems.The next sections in this paper are organized as follows: In Sect. 7.2, related works are discussed. TheproposedarchitecturesarediscussedintheSect.7.3.Finally,inSect.7.4,conclusionandfutureworkwerediscussed. 7.2 Related Work In this research work we have [5] proposed an architecture for an IoT, itconsists of the IoT nodes and the protocols where we can plant the sensors and the actuators and thisarchitecturecanbefitfordifferenthealthcaresystemsandvariousotherapplicationsindifferentdisciplines. The experiments also proved that this is effective architecture for the IoT. In [6], authorsproposed a detection method for various attacks in the household appliance. The analysis was done in thetheoretic fields by the simulation methods. In terms of the accuracy and localization, the proposed methodproved as good. In paper [7], authors proposed a novel architecture for 5G smart dieses, in the layer-wiseand comparison of the various diabetic versions and discussed about the personalized healthcare systems.In the paper [8], proposed the various medical care services like the metabolic syndrome with the cloud-based personalized healthcare systems. Here service broker is responsible for the dynamic creations in theperspectiveoftheusers.Thisarchitecturecanbeusedforthehealthcaresystemswiththeadditionofsomesecurity features. In paper [9] proposed the security features for the hybrid cloud architecture for the IoTwith some security features. Particularly the authors had concentrated on the issues like scalability andinteroperability, and some research challenges have been discussed. 7.3 Proposed Architecture In theproposed system, the communication will be from the sensors that equipped with the patient's body to thecloud, and from the cloud to the corresponding doctor. The authorized doctor will access the data fromcloud and treat the patient based on the obtained values. This will be done in layer-wise. The layers in IoTare user-side layer, edge-side layer, and cloud-side layer [10]. The security should be provided in andbetween those layers. The data flowed from one layer to other layer (Fig. 7.2). Here the goal is to remotemonitor the patient that living in their home from the hospital. To achieve this goal, the sensor values,communications links, data in cloud, etc., in all layers should be secured. To provide security for thoselayers in

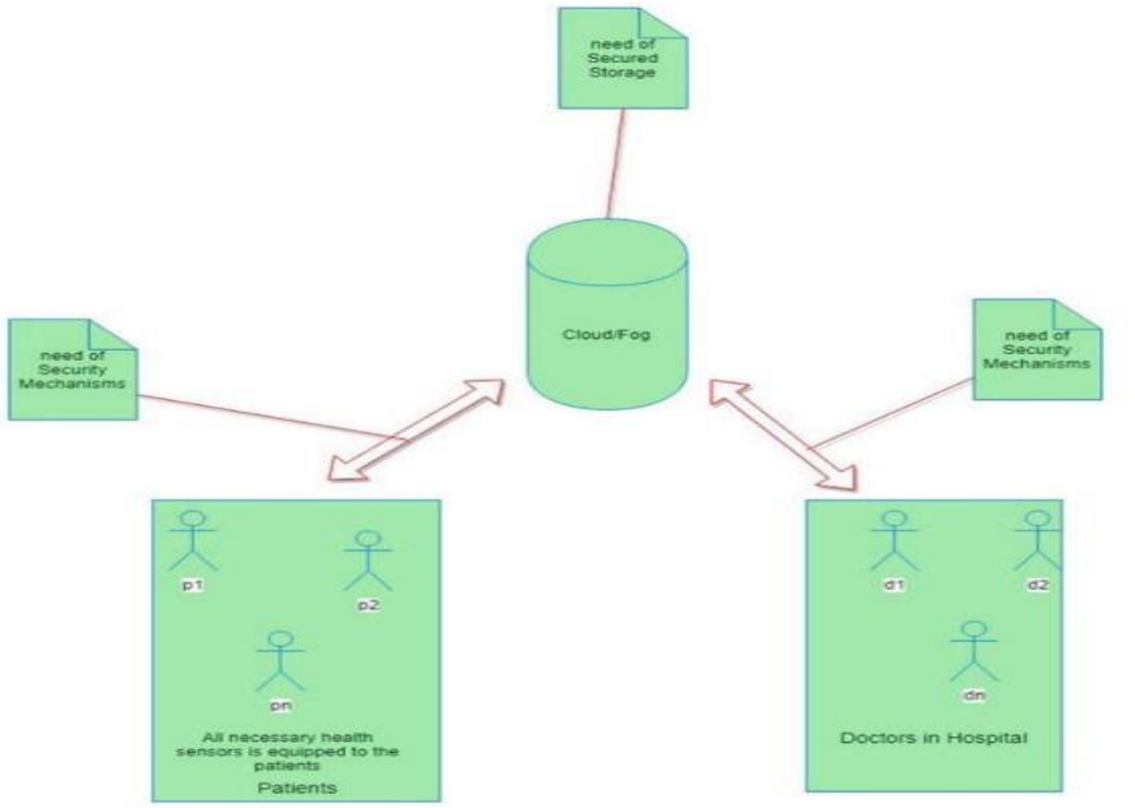

need of Secured Storage
Cloud/Fog
need of Security Mechanisms
need of Security Mechanisms
p1
p2
pn
All necessary health sensors is equipped to the patients
Patients
d1
d2
dn
Doctors in Hospital

Fig. 7.2 Novel security architecture for the IoT-enabled personalized healthcare systems

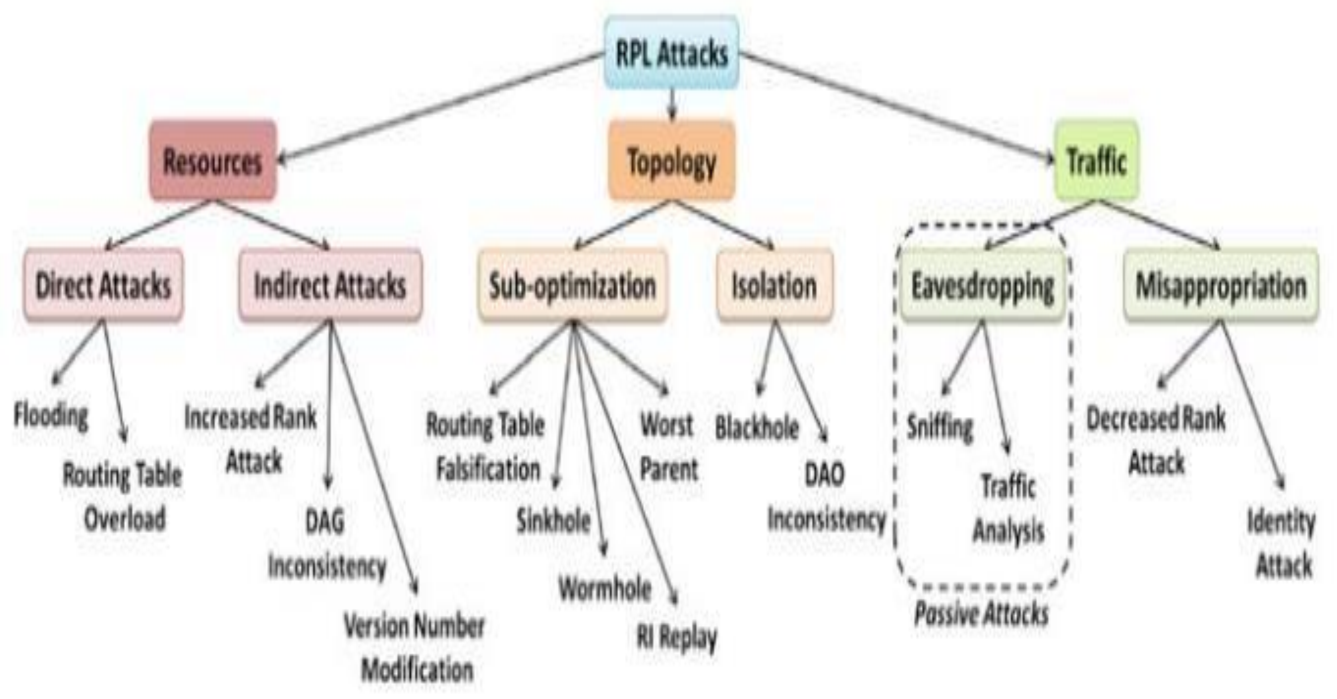


Fig. 7.3 Taxonomy of attacks against RPL networks [11]

be equipped in the patient's body and patient's home for continuous monitoring the patient and patient'shealth condition. The continuous data obtained by that equipped sensors will be stored in the componentcloud. From that cloud component, the data can be accessed by the doctor module. The doctor's moduleconsists of the doctors and there have a facility to collect the data of a patient from the cloud. In betweenall the three modules, doctor, patient, and cloud, secured communication links should be established, i.e.,the security mechanisms module. For storing the data in cloud, the secured cloud storage component isused. In the security mechanisms module, with the communication technology, there is necessity tointegrate any one or more security mechanisms like the intrusion detection systems (IDS), authorizationmechanisms, cryptographic mechanisms, and secured routing protocols. Simulataneously in the securedstorage module need to use the secured data storage techniques like authorization and cryptographicalgorithms.

7.4Conclusion

The Internet of Things is the emerging technology, and it has various applications in all disciplines. Thispaperisfocusedonthepersonalizedhealthcaresystems,andnovelsecurityarchitecturefortheIoT-enabledpersonalized healthcare systems is proposed. This architecture leads to many research challenges likedeveloping novel IDS, cryptographic mechanisms, and secured routing protocols. In the future work, wedevelopthenovelintrusiondetectionsystemsfortheRPL-basednetworksformitigatingtheroutingattackslikethe black holeattack,wormholeattack, and rankattack.