

# IOT QUESTION BANK

**Q1) Define the term Embedded System. State its component.**

**Ans :**

Embedded means something that is attached to another thing. An embedded system can be thought of as a computer hardware system having software embedded in it. An embedded system can be an independent system or it can be a part of a large system. An embedded system is a microcontroller or microprocessor based system which is designed to perform a specific task. For example, a fire alarm is an embedded system; it will sense only smoke.

An embedded system has three components –

- **It has hardware.**
- **It has application software.**
- **It has Real Time Operating system (RTOS)** that supervises the application software and provide mechanism to let the processor run a process as per scheduling by following a plan to control the latencies. RTOS defines the way the system works. It sets the rules during the execution of application program. A small scale embedded system may not have RTOS.

So we can define an embedded system as a Microcontroller based, software driven, reliable, real-time control system.

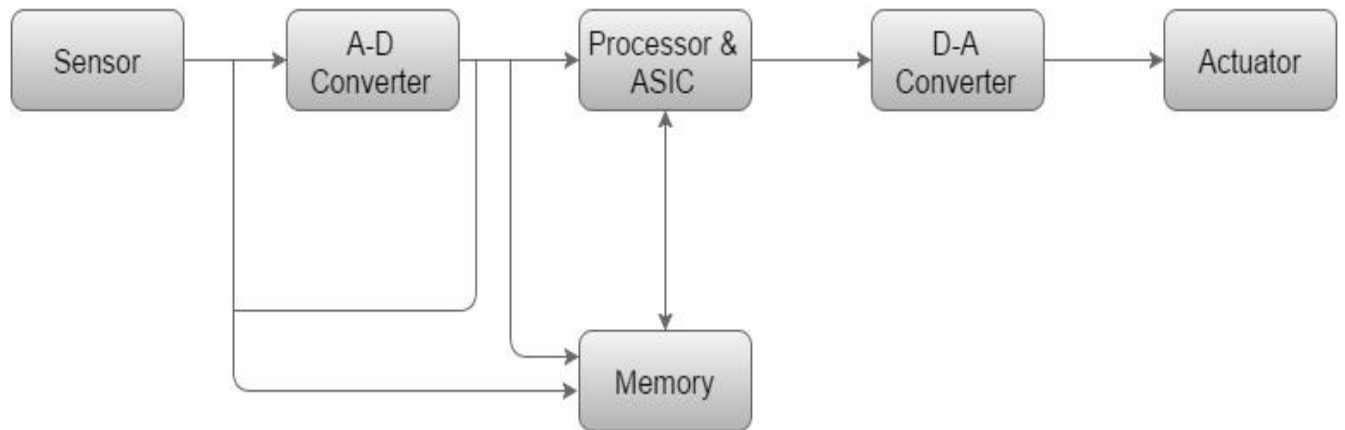
Advantages of Embedded System

- Easily Customizable
- Low power consumption
- Low cost
- Enhanced performance

Disadvantages of Embedded System

- High development effort
- Larger time to market

The following illustration shows the basic structure of an embedded system –



- **Sensor** – It measures the physical quantity and converts it to an electrical signal which can be read by an observer or by any electronic instrument like an A2D converter. A sensor stores the measured quantity to the memory.
- **A-D Converter** – An analog-to-digital converter converts the analog signal sent by the sensor into a digital signal.
- **Processor & ASICs** – Processors process the data to measure the output and store it to the memory.
- **D-A Converter** – A digital-to-analog converter converts the digital data fed by the processor to analog data
- **Actuator** – An actuator compares the output given by the D-A Converter to the actual (expected) output stored in it and stores the approved output.

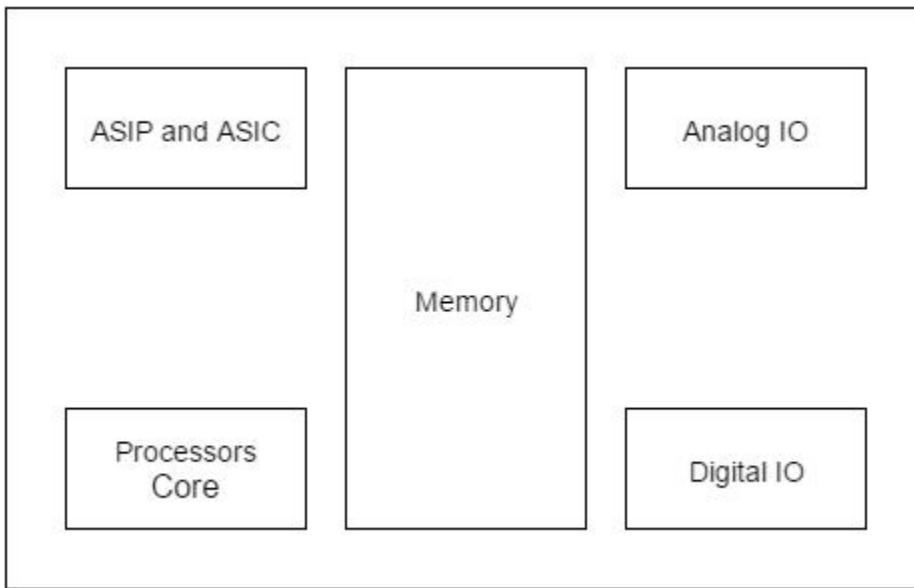
## Q2) State the characteristics of Embedded System.

Ans :

- **Single-functioned** – An embedded system usually performs a specialized operation and does the same repeatedly. For example: A pager always functions as a pager.
- **Tightly constrained** – All computing systems have constraints on design metrics, but those on an embedded system can be especially tight. Design metrics is a measure of an implementation's features such as its cost, size, power, and performance. It must be of a size to fit on a single chip, must perform fast enough to process data in real time and consume minimum power to extend battery life.
- **Reactive and Real time** – Many embedded systems must continually react to changes in the system's environment and must compute certain results in real time without any delay. Consider an example of a car cruise controller; it continually monitors and reacts

to speed and brake sensors. It must compute acceleration or de-accelerations repeatedly within a limited time; a delayed computation can result in failure to control of the car.

- **Microprocessors based** – It must be microprocessor or microcontroller based.
- **Memory** – It must have a memory, as its software usually embeds in ROM. It does not need any secondary memories in the computer.
- **Connected** – It must have connected peripherals to connect input and output devices.
- **HW-SW systems** – Software is used for more features and flexibility. Hardware is used for performance and security.



**Q3) Explain Microcontroller as SoC and draw its basic architecture.**

**Ans :**

A microcontroller is a single-chip VLSI unit (also called **microcomputer**) which, although having limited computational capabilities, possesses enhanced input/output capability and a number of on-chip functional units.

CPU	RAM	ROM
I/O Port	Timer	Serial COM Port

Microcontrollers are particularly used in embedded systems for real-time control applications with on-chip program memory and devices.

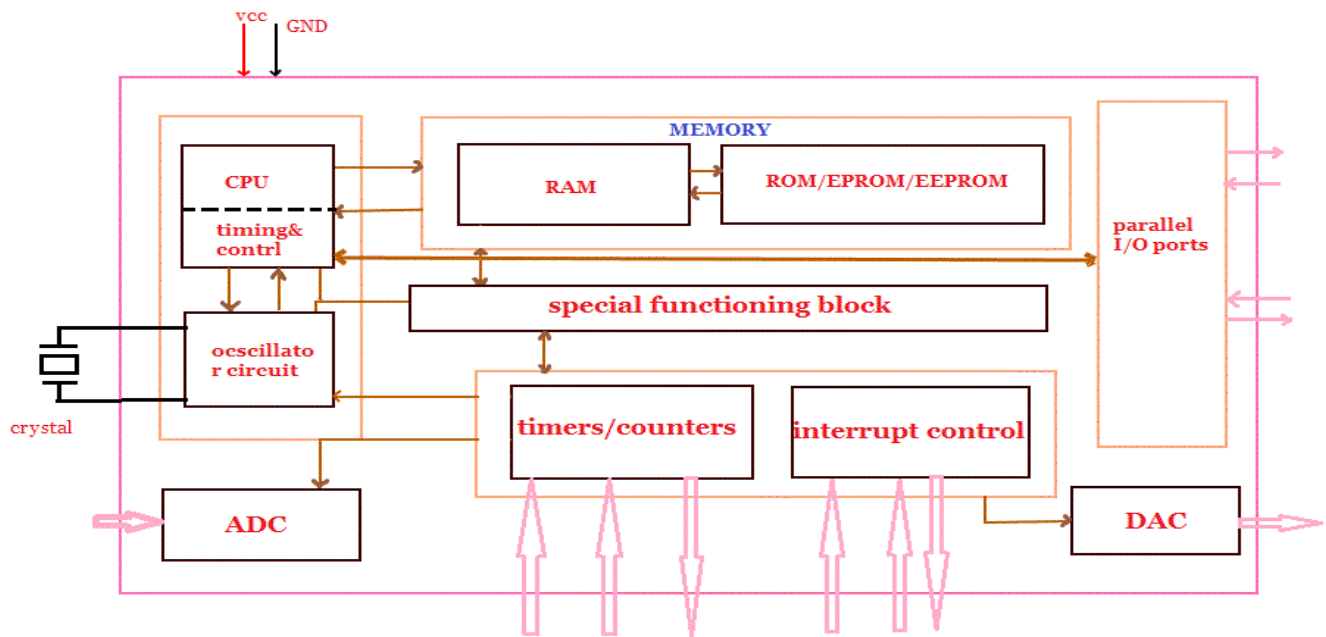
It is a Single task oriented. For example, a washing machine is designed for washing clothes only.

RAM, ROM, I/O Ports, and Timers cannot be added externally. These components are to be embedded together on a chip and are fixed in numbers.

Fixed number for memory or I/O makes a microcontroller ideal for a limited but specific task.

Microcontrollers are lightweight and cheaper.

A microcontroller-based system consumes less power and takes less space.



- **CPU**

CPU is the brain of a microcontroller. CPU is responsible for fetching the instruction, decodes it, then finally executed. CPU connects every part of a microcontroller into a single system. The primary function of CPU is fetching and decoding instructions. The instruction fetched from program memory must be decoded by the CPU.

- **Memory**

The function of memory in a microcontroller is the same as a microprocessor. It is used to store data and program. A microcontroller usually has a certain amount of RAM and ROM (EEPROM, EPROM, etc) or flash memories for storing program source codes.

- **Parallel input/output ports**

Parallel input/output ports are mainly used to drive/interface various devices such as LCD'S, LED'S, printers, memories, etc to a microcontroller.

- ***Serial ports***

Serial ports provide various serial interfaces between a microcontroller and other peripherals like parallel ports.

- ***Timers/counters***

This is the one of the useful function of a microcontroller. A microcontroller may have more than one timer and counters. The timers and counters provide all timing and counting functions inside the microcontroller. The major operations of this section are performed clock functions, modulations, pulse generations, frequency measuring, making oscillations, etc. This also can be used for counting external pulses.

- ***Analog to Digital Converter (ADC)***

ADC converters are used for converting the analog signal to digital form. The input signal in this converter should be in analog form (e.g. sensor output) and the output from this unit is in digital form. The digital output can be used for various digital applications (e.g. measurement devices).

- ***Digital to Analog Converter (DAC)***

DAC perform reversal operation of ADC conversion. DAC converts the digital signal into analog format. It usually used for controlling analog devices like DC motors, various drives, etc.

- ***Interrupt control***

The interrupt control used for providing interrupt (delay) for a working program. The interrupt may be external (activated by using interrupt pin) or internal (by using interrupt instruction during programming).

- ***Special functioning block***

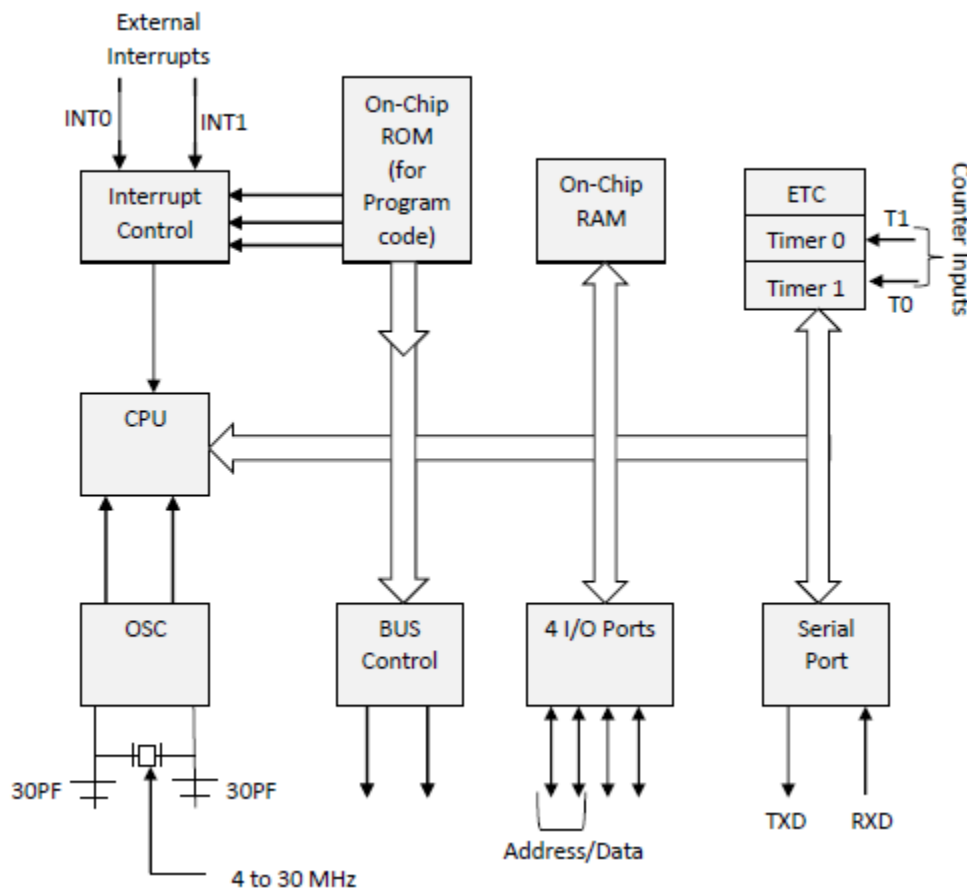
Some microcontrollers used only for some special applications (e.g. space systems and robotics) these controllers containing additional ports to perform such special operations. This considered as special functioning block.

**Q4) Draw 8051 Microcontroller as SoC as functional block diagram and explain important modules of it.**

**Ans:**

8051 microcontroller is designed by Intel in 1981. It is an 8-bit microcontroller. It is built with 40 pins DIP (dual inline package), 4kb of ROM storage and 128 bytes of RAM storage, 2 16-bit timers. It consists of are four parallel 8-bit ports, which are programmable as well as addressable as per the requirement. An on-chip crystal oscillator is integrated in the microcontroller having crystal frequency of 12 MHz.

In the following diagram, the system bus connects all the support devices to the CPU. The system bus consists of an 8-bit data bus, a 16-bit address bus and bus control signals. All other devices like program memory, ports, data memory, serial interface, interrupt control, timers, and the CPU are all interfaced together through the system bus.



There are some modules of 8051

- Memory –

A Microcontroller needs program memory to store program/instructions to perform defined tasks. This memory is termed as ROM. Furthermore the Microcontroller also requires data

memory to store the operands/data on a temporary basis. This memory is known as RAM. The 8051 Microcontroller is built with 4 Kb on-chip Read Only Memory (ROM) and 128 bytes Random Access Memory (RAM).

- Address Bus –

A bus of the Microcontroller can be defined as a group of wire which can act as a medium for the transfer of data. There are two buses present in the 8051 Microcontroller. While we are already aware of the Data Bus, let us know about the Address Bus of the 8051 Microcontroller. The address bus, which is used to address memory locations, is 16-bit wide. Furthermore, the address bus can also be used to transfer data from the CPU (Central Processing Unit) to the memory. Hence, for obvious reasons the address bus is unidirectional.

- Interrupts –

The most powerful attribute of the 8051 Microcontroller is the concept of Interrupts. The interrupt is a mechanism to –

- **Temporarily suspend the ongoing program,**
- **Pass the control to a subroutine,**
- **Execute the subroutine,**
- **Resume the ongoing/main program.**

Interrupts can be of various types, such as, Software and Hardware interrupts, Non-maskable and maskable interrupts, etc. Now the 8051 Microcontroller incorporates five interrupts. These are :

1. **INT0** – External Hardware Interrupt.
2. **TFO** – Timer 0 Overflow Interrupt.
3. **INT1** – External Hardware Interrupt.
4. **TF1** – Timer 1 Overflow Interrupt.
5. **R1/T1** – Serial communication Interrupt.

- Input/Output Ports –

The 8051 Microcontroller needs to be connected to the peripheral devices in order to control their operations. The I/O Ports are responsible for the connection of the Microcontroller to its peripheral devices. There are total Four 8-bit Input/Output Ports present in this Microcontroller.

Additionally, these are some important features of 8051 microcontroller given as follows :

1. **Two 16-bit Timers and Counters.**
2. **A Data Pointer and a Program Counter of 16-bit each.**

3. **128 User defined Flags.**
4. **Four Register banks.**
5. **31 General Purpose Registers which are of 8-bit each.**

**Q5) Define the following**

- a) **INTERRUPT** -> An interrupt is a signal sent to the [processor](#) that interrupts the current [process](#). It may be generated by a hardware device or a software program. A hardware interrupt is often created by an [input device](#) such as a [mouse](#) or [keyboard](#). For example, if you are using a [word processor](#) and press a key, the program must process the input immediately. Software interrupts are used to handle errors and [exceptions](#) that occur while a program is running.
- b) **HARDWARE** -> Computer **hardware** is the collection of all the parts you can physically touch. Computer hardware is the collection of physical parts of a computer system. This includes the computer case, monitor, keyboard, and mouse. It also includes all the parts inside the computer case, such as the hard disk drive, motherboard, video card, and many others. Computer hardware is what you can physically touch.
- c) **SOFTWARE** -> Software is a set of instructions, data or [programs](#) used to operate [computers](#) and execute specific tasks.
- d) **FIRMWARE** -> Firmware is a software program or set of instructions programmed on a hardware device. It provides the necessary instructions for how the device communicates with the other computer hardware. Firmware can be thought of as "semi-permanent" since it remains the same unless it is updated by a firmware updater.

**RTOS** -> A Real Time Operating System, commonly known as an RTOS, is a software component that rapidly switches between tasks, giving the impression that multiple programs are being executed at the same time on a single processing core.

In actual fact the processing core can only execute one program at any one time, and what the RTOS is actually doing is rapidly switching between individual programming threads (or Tasks) to give the impression that multiple programs are executing simultaneously.

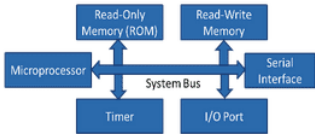
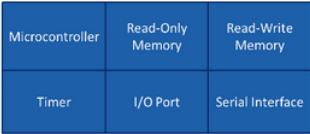
**HARD REAL TIME OS** => hard real-time system (also known as an immediate real-time system) is hardware or software that must operate within the confines of a stringent deadline. The application may be considered to have failed if it does not complete its function within the



allotted time span. Examples of hard real-time systems include components of pacemakers, anti-lock brakes and aircraft control systems.

- e) **SOFT REAL TIME OS** -> In soft real time system, the meeting of deadline is not compulsory for every time for every task but process should get processed and give the result. Even the soft real time systems cannot miss the deadline for every task or process according to the priority it should meet the deadline or can miss the deadline. If system is missing the deadline for every time the performance of the system will be worse and cannot be used by the users. Best example for soft real time system is personal computer, audio and video systems, etc.
- f) **SYSTEM** -> Is a collection of entities(hardware,software and liveware) that are designed to receive, process, manage and present information in a meaningful format.

## Q.6 Explain difference between microprocessor and microcontroller

Microprocessor	Micro Controller
	
Microprocessor is heart of Computer system.	Micro Controller is a heart of embedded system.
It is just a processor. Memory and I/O components have to be connected externally	Micro controller has external processor along with internal memory and i/o components
Since memory and I/O has to be connected externally, the circuit becomes large.	Since memory and I/O are present internally, the circuit is small.
Cannot be used in compact systems and hence inefficient	Can be used in compact systems and hence it is an efficient technique
Cost of the entire system increases	Cost of the entire system is low
Due to external components, the entire power consumption is high. Hence it is not suitable to used with devices running on stored power like batteries.	Since external components are low, total power consumption is less and can be used with devices running on stored power like batteries.
Most of the microprocessors do not have power saving features.	Most of the micro controllers have power saving modes like idle mode and power saving mode. This helps to reduce power consumption even further.
Since memory and I/O components are all external, each instruction will need external operation, hence it is relatively slower.	Since components are internal, most of the operations are internal instruction, hence speed is fast.
Microprocessor have less number of registers, hence more operations are memory based.	Micro controller have more number of registers, hence the programs are easier to write.
Microprocessors are based on von Neumann model/architecture where program and data are stored in same memory module	Micro controllers are based on Harvard architecture where program memory and Data memory are separate
Mainly used in personal computers	Used mainly in washing machine, MP3 players

### **Q.7 What do you mean by SoC and explain its modules.**

A system on a chip (SoC) combines the required electronic circuits of various computer components onto a single, integrated chip (IC). SoC is a complete electronic substrate system that may contain analog, digital, mixed-signal or radio frequency functions. Its components usually include a graphical processing unit (GPU), a central processing unit (CPU) that may be multi-core, and system memory (RAM).

Because SoC includes both the hardware and software, it uses less power, has better performance, requires less space and is more reliable than multi-chip systems. Most system-on-chips today come inside mobile devices like smartphones and tablets.

An SoC usually contains various components such as:

- Operating system
- Utility software applications
- Voltage regulators and power management circuits
- Timing sources such as phase lock loop control systems or oscillators
- A microprocessor, microcontroller or digital signal processor
- Peripherals such as real-time clocks, counter timers and power-on-reset generators
- External interfaces such as USB, FireWire, Ethernet, universal asynchronous receiver-transmitter or serial peripheral interface bus
- Analog interfaces such as digital-to-analog converters and analog-to-digital converters
- RAM and ROM memory

### **Q.8 Advantages and Disadvantages of SoC.**

#### **Advantages of an SoC**

- An SoC consumes less power. Usually 90% of power consumption is in data and bus address cabling. Since all the components are on the same chip and internally connected, and their size is also very small, the power consumption is hugely decreased.
- A smaller size means it is lightweight and of small size.
- Overall, the cost of an SoC is small due to advancements in VLSI technology. As mentioned in the first point, cabling is not much required and so the cost of cabling is conserved.
- An SoC provides greater design security at hardware and firmware levels.
- An SoC provides faster execution due to high speed processor and memory.

### **Disadvantages of an SoC**

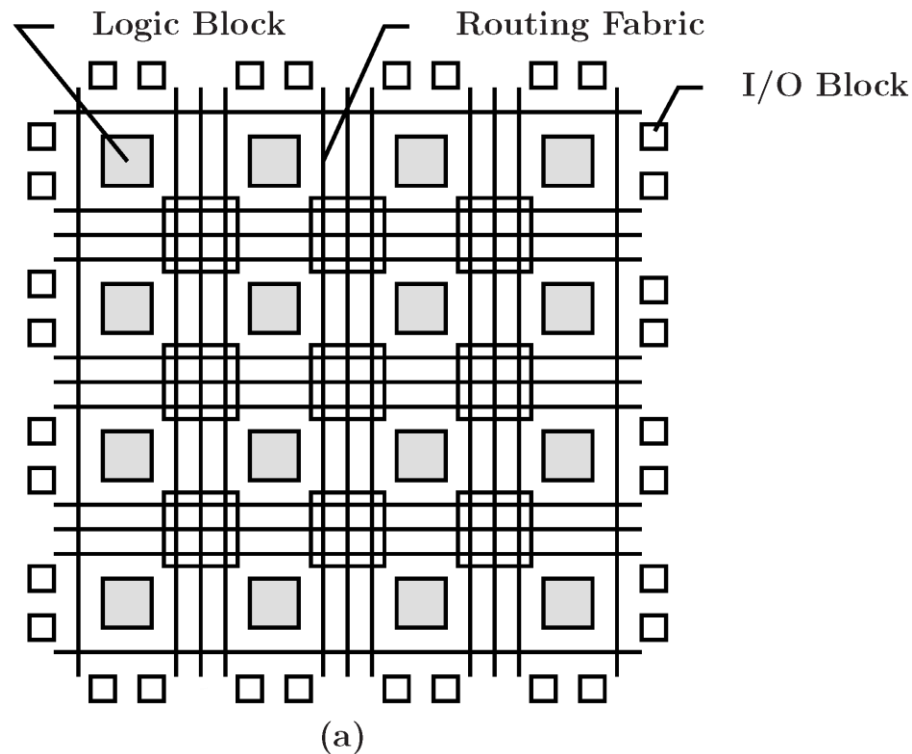
- Initial cost of design and development is very high. If the number of SoCs is small, the cost per SoC will be very high.
- Even a single transistor or system damage may prove to be very costly as the complete board has to be replaced, and its servicing is very expensive.
- Integrating all systems on single chip increases complexity.
- It is not suitable for power-intensive applications.

### **Q9) List various complexity issues related with SoC.**

- Functionality
- Compatibility
- Security
- Reliability
- Throughput
- Robustness
- Technology Churn
- Performance
- Availability
- Capacity
- Cost

### **Q10) Explain features of FPGA along with neat diagram.**

1. Easily configurable
2. Programmable Logic device
3. Thousands of Input Outputs
4. High Speed with Synchronous Circuitry up to 50 MHz
5. Low Power Consumption
6. Programmable at Job site to get required function
7. Programming at Behavioural level
8. System On Chip - ADC, DAC, Microcontroller, Microprocessor, DSP etc
9. FPGA is used in developing Soft Microprocessor, Software Defined Radios, Aerospace & Defence, High Performance Computing Systems like Servers, Super Computers etc.
10. Programming can be done using VHDL, Verilog, SystemVerilog, MyHDL, MATLAB, C, C++, SystemC, etc.



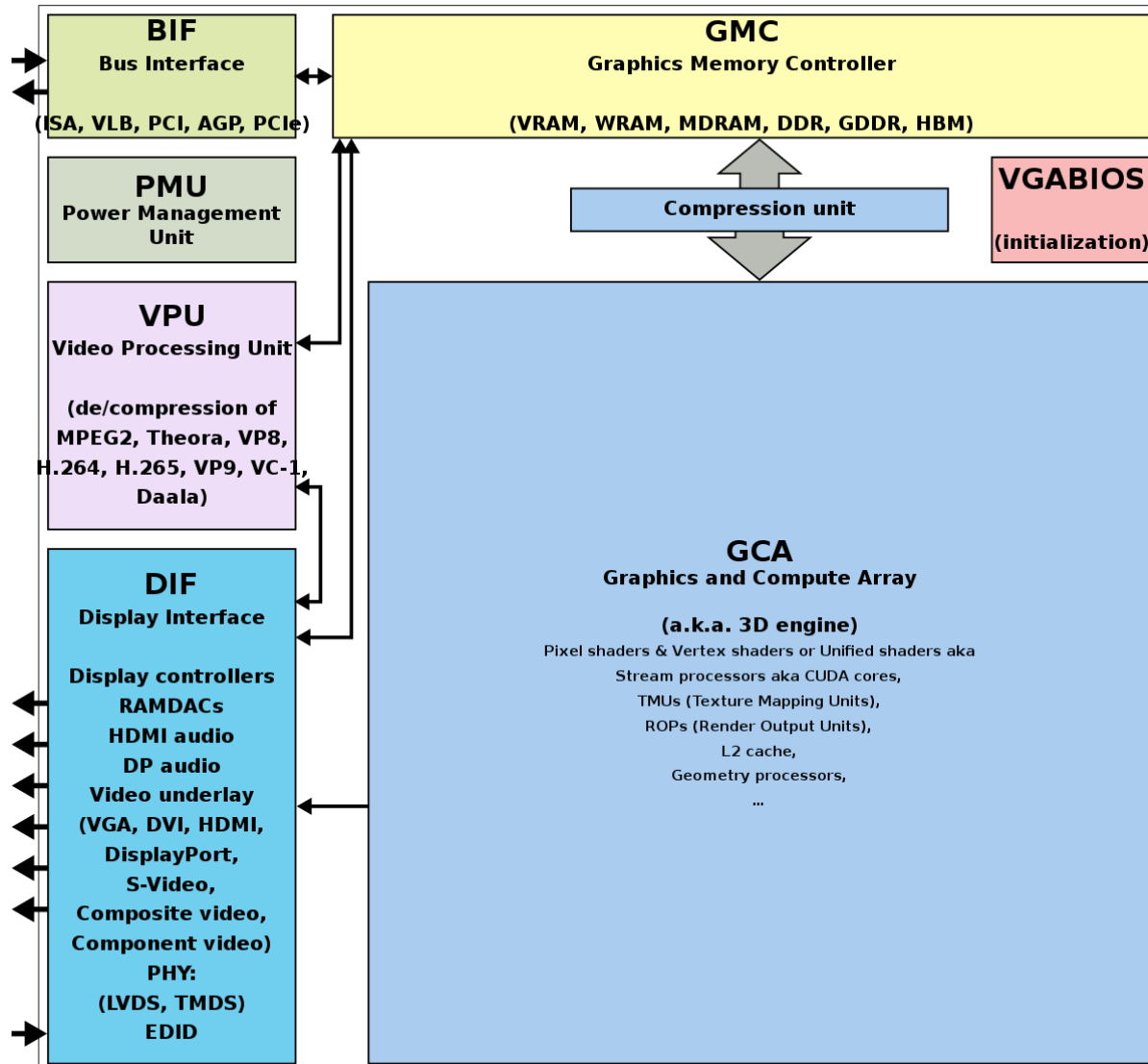
**Q11) State and explain in brief applications of FPGA.**

FPGAs have gained rapid growth over the past decade because they are useful for a wide range of applications. Specific application of an FPGA includes digital signal processing, bioinformatics, device controllers, software-defined radio, random logic, ASIC prototyping, medical imaging, computer hardware emulation, integrating multiple SPLDs, voice recognition, cryptography, filtering and communication encoding and many more.

Usually, FPGAs are kept for particular vertical applications where the production volume is small. For these low-volume applications, the top companies pay in hardware costs per unit. Today, the new performance dynamics and cost have extended the range of viable applications.

Some more common FPGA Applications are: Aerospace and Defence, Medical Electronics, ASIC Prototyping, Audio, Automotive, Broadcast, Consumer Electronics, Distributed Monetary Systems, Data Centre, High Performance Computing, Industrial, Medical, Scientific Instruments, Security systems, Video & Image Processing, Wired Communications, Wireless Communications.

**Q12) Draw and explain structure of GPU.**



**PMU** – It's responsible for:

- Telling the computer when to turn on, turn off, go to sleep, and wake up.
- Maintaining the system's PRAM (Parameter Random Access Memory).
- Managing system resets from various types of commands.
- Managing the real-time clock (date and time).

**VPU** - is a specialised processor which takes video stream as input and has the capability to perform highly complex processes on the input stream. Processing a video stream means performing calculation on each pixel value of multiple frames of the video, which is a huge amount of data, which isn't possible without a VPU.

**DIF** - continuously fetches the frame buffer data (the buffer in the system memory that contains the image to be displayed) and transmit the same to the display.

**GMC** – used to read and write data from and into various memories, related to display and graphics requirements.

### **Q.13 Compare CPU and GPU?**

**CPU (Central Processing Unit)** is a device primarily acts as the brain for every embedded system. It consists of an ALU (Arithmetic Logic Unit) used to temporarily store the data and perform calculations and a CU (Control Unit) which performs instruction sequencing and branching. It also interacts to the other units of the computer such as memory, input and output, for executing the instruction from the memory this is the reason an interface is also a crucial part of the CPU. The I/O interface is sometimes included in the control unit.

It provides address, data and control signal while receives instructions, data, status signal and interrupt which is processed with the help of the system bus. A system bus is a group of various busses such as address, control and data bus. The CPU assigns more hardware unit to fast cache while low to computation, unlike GPU.

The **GPU (Graphics Processing Unit)** is a processor specifically designed for computing the graphical displays. It is typically incorporated with CPU for sharing RAM with CPU which is good for the most computing task. It is needed for the high-end graphics intensive processing. The discrete GPU unit contains its own RAM known as VRAM for video RAM. The advanced GPU system cooperatively works with the multi-core CPUs. At first, the graphics unit was introduced by the Intel and IBM in the 1980s. These cards were enabled with simple functionalities such as area filling, alteration of simple images, shape drawing and so on.

The modern graphics are capable of performing the research and analysis task, often surpassing CPUs because of its extreme parallel processing. In the GPU the several processing units are stripped together where no cache coherency exist.

<b>BASIS FOR COMPARISON</b>	<b>CPU</b>	<b>GPU</b>
Stands For	Central Processing Unit	Graphics Processing Unit

Focuses On	Low-Latency	High Throughput
Good At	Processing serial instructions.	Processing parallel instructions.
Contains	Fewer powerful cores.	A lot of weaker cores.
Feature	Control logic for out-of-order and speculative executions.	Architecture is tolerant of memory latency.
Speed	Effective	Can be higher than CPU's.
Memory Consumption	High	Low

**Q14.List and explain Various Types of GPU.**

1) Dedicated Graphics Card.

- a) The GPU's of most powerful class typically interface with the motherboards by means of an expansion slot such as PCI Express.
- b) A dedicated GPU is not necessarily removable, nor does it necessarily interface with the motherboard in standard fashion.
- c) The term dedicated refers to the dedicated RAM that these GPU's have for the cards use.

2) Integrated GPU.

a) Integrated GPU utilize a portion of Computer's systems RAM rather than having its own dedicated memory.

b) IGPU's can be integrated with the motherboard as a part of chipset or on same Disc with CPU.

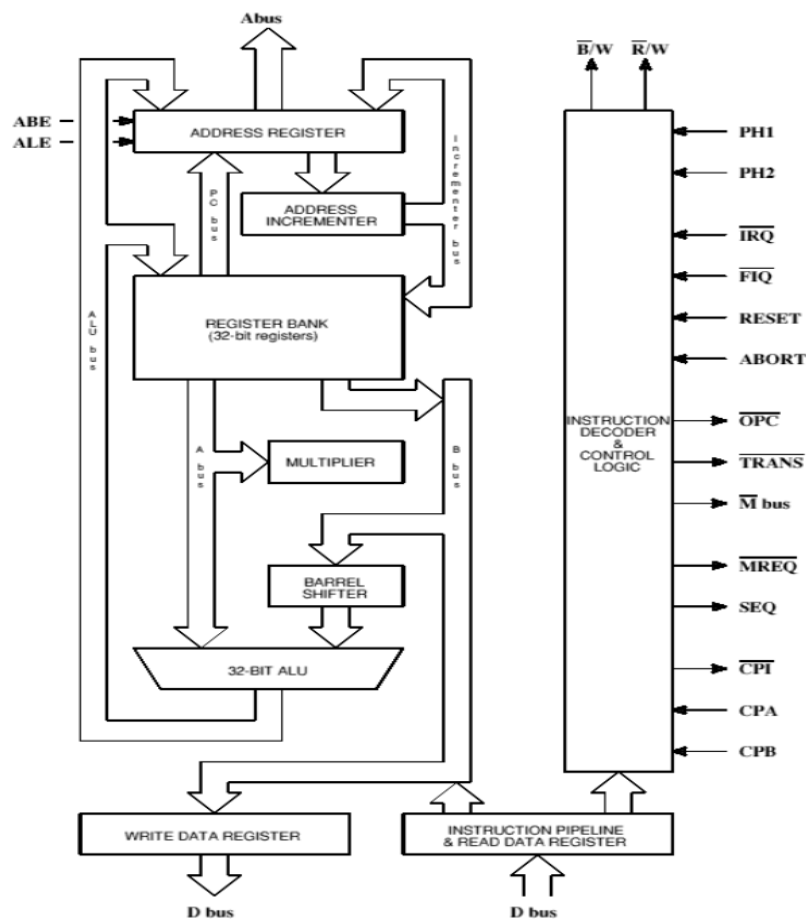
### 3) Hybrid Graphics Processing Unit.

a) These newer class of GPUs compete with IGPUs in the low end Desktop markets.

b) Hybrid Cards are somewhat more expensive then dedicated cards.

c) These share memory with the system and have a small dedicated cache to make up for the high latency of the System RAM.

### Q15.Draw ARM and explain its main parts





## The ARM Architecture

- Arithmetic Logic Unit
  - The ALU has two 32-bits inputs. The primary comes from the register file, whereas the other comes from the shifter. Status registers flags modified by the ALU outputs. The V-bit output goes to the V flag as well as the Count goes to the C flag. Whereas the foremost significant bit really represents the S flag, the ALU output operation is done by NOR to get the Z flag. The ALU has a 4-bit function bus that permits up to 16 opcodes to be implemented.
- Booth Multiplier Factor
  - The multiplier factor has 3 32-bit inputs and the inputs return from the register file. The multiplier output is barely 32-Least Significant Bits of the merchandise. The entity representation of the multiplier factor is shown in the above block diagram. The multiplication starts whenever the beginning 04 input goes active. Fin of the output goes high when finishing.
- Barrel Shifter
  - The barrel shifter features a 32-bit input to be shifted. This input is coming back from the register file or it might be immediate data. The shifter has different control inputs coming back from the instruction register. The Shift field within the instruction controls the operation of the barrel shifter. This field indicates the kind of shift to be performed (logical left or right, arithmetic right or rotate right). The quantity by which the register ought to be shifted is contained in an immediate field within the instruction or it might be the lower 6 bits of a register within the register file.
- Control Unit
  - For any microprocessor, control unit is the heart of the whole process and it is responsible for the system operation, so the control unit design is the most important part within the whole design. The control unit is sometimes a pure combinational circuit design. Here, the control unit is implemented by easy state machine. The processor timing is additionally included within the control unit. Signals from the control unit are connected to each component within the processor to supervise its operation.

#### **Q16 State features of ARM.**

- Load/store architecture.
- An orthogonal instruction set.
- Mostly single-cycle execution.
- Enhanced power-saving design.
- 64 and 32-bit execution states for scalable high performance.
- Hardware virtualization support.
- The simplified design of ARM processors enables more efficient multi-core processing and easier coding for developers.
- The ARM 7 core uses a three-stage pipeline to increase the flow of instructions to the processor. This allows multiple simultaneous operations to take place.
- The ARM can be configured to treat stored words in either big-endian or little-endian format.

#### **Q17. What are the different CPU modes available in RAM.**

CPU modes refer to the various ways that the processor creates an operating environment for itself. Specifically, the processor mode controls how the processor sees and manages the system memory and the tasks that use it. There are three different modes of operation:

##### **Real Mode:**

Real mode is characterized by a 20-bit segmented memory address space (giving exactly 1 MiB of addressable memory) and unlimited direct software access to all addressable memory, I/O addresses and peripheral hardware. Real mode provides no support for memory protection, multitasking, or code privilege levels.

##### **Protected Mode:**

The name of this mode comes from its primary use, which is by multitasking operating systems. Each program that is running has its own assigned memory locations, which are protected from

conflict with other programs. If a program tries to use a memory address that it isn't allowed to, a "protection fault" is generated. The advantages of protected mode (compared to real mode) are:

- Full access to all of the system's memory. There is no 1 MB limit in protected mode.
- Ability to multitask, meaning having the operating system manage the execution of multiple programs simultaneously.
- Support for virtual memory, which allows the system to use the hard disk to emulate additional system memory when needed.
- Faster (32-bit) access to memory, and faster 32-bit drivers to do I/O transfers

Virtual Real Mode:

It emulates real mode from within protected mode. A protected mode operating system such as Windows can create multiple virtual real mode machines, each of which appear to the software running them as if they are the only software running on the machine. Each virtual machine gets its own 1 MB address space, an image of the real hardware BIOS routines, etc.

**Q18. List and explain various components of Robot acting as SoC.**

- **Controller** – A microcontroller is a computer which is placed on a single integrated circuit chip. It consists of memory, a processor, as well as input-output interfaces. Microcontrollers are programmed to run a certain task, which means, if there is a need to change or enhance its functionality, one must install a new program on the chip.
- **Actuators** – Actuators are the energy conversion device used inside a robot. The major function of actuators is to convert energy into movement.
- **Electric motors (DC/AC)**- Motors are electromechanical component used for converting electrical energy into its equivalent mechanical energy. In robots motors are used for providing rotational movement.
- **Sensors** – Sensors provide real time information on the task environment. Robots are equipped with tactile sensor it imitates the mechanical properties of touch receptors of human fingerprints and a vision sensor is used for computing the depth in the environment.

- Power Supply – The working power to the robot is provided by batteries, hydraulic, solar power, or pneumatic power sources.

### **Q19. In what capacity SoC plays a role in construction of Robot?**

#### Artificial Intelligence

Artificial intelligence (AI) chips are specialized silicon chips, which incorporate AI technology and are used for machine learning. AI is a highly useful tool in robotic assembly applications. When combined with advanced vision systems, AI can help with real-time course correction, which is particularly useful in complex manufacturing sectors like aerospace. AI can also be used to help a robot learn on its own which paths are best for certain processes while it's in operation.

#### Control theory

Control theory in control systems engineering is a subfield of mathematics that deals with the control of continuously operating dynamical systems in engineered processes and machines. In the context of robotics control plays a fundamental role, particularly in mechanical robots, where actuators must be manipulated in an organized and even intelligent way by algorithms. Those algorithms are basically controls.

#### Robot Learning

Robot learning is a research field at the intersection of machine learning and robotics. It studies techniques allowing a robot to acquire novel skills or adapt to its environment through learning algorithms. The embodiment of the robot, situated in a physical embedding, provides at the same time specific difficulties (e.g. high-dimensionality, real time constraints for collecting data and learning) and opportunities for guiding the learning process .

#### Computer Vision

Computer vision is a theory and technology of creating machines that can detect and classify objects and their movement receiving information from the series of images.

Important issue of the artificial intelligence is an automatic planning or decision-making in systems which can perform mechanical actions, such as moving a robot through certain environment.

**Q20. Define the term Embedded System. State its components.**

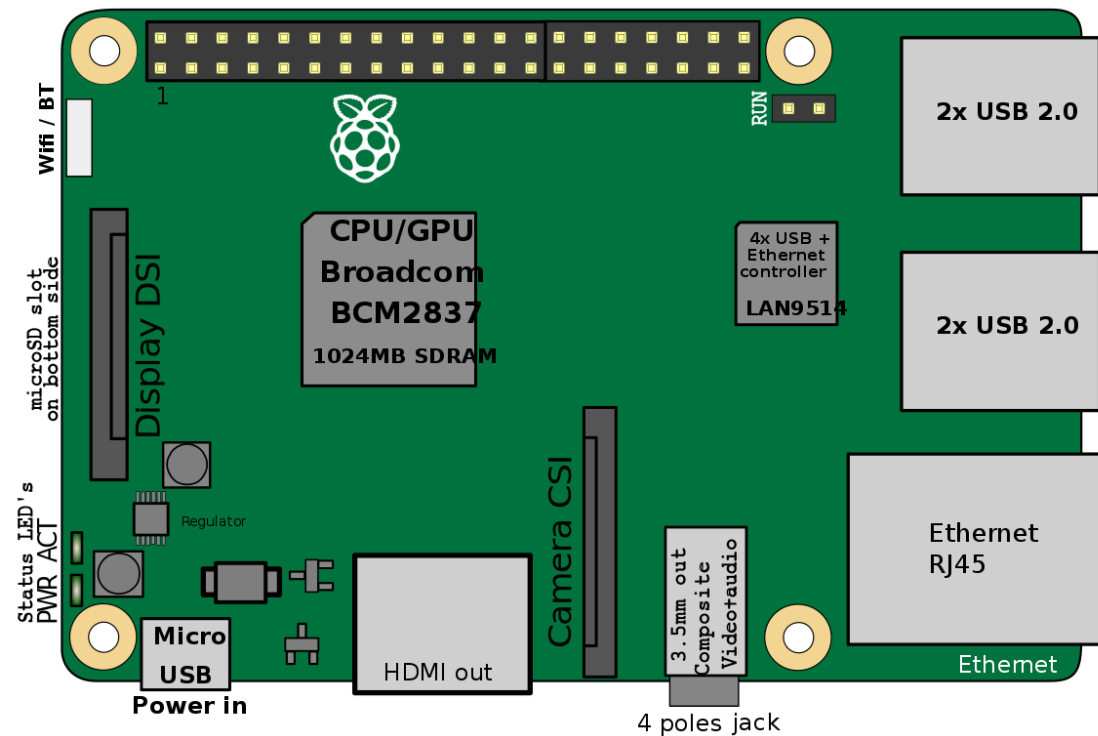
An embedded system is a computer system—a combination of a computer processor, computer memory, and input/output peripheral devices—that has a dedicated function within a larger mechanical or electrical system. It is embedded as part of a complete device often including electrical or electronic hardware and mechanical parts.

Components of Embedded System:

1. Processor : A microprocessor is a computer processor which incorporates the functions of a computer's central processing unit (CPU) on a single integrated circuit (IC), or at most a few integrated circuits. The microprocessor is a multipurpose, clock driven, register based, digital-integrated circuit which accepts binary data as input, processes it according to instructions stored in its memory, and provides results as output. (Can also be a microcontroller).
2. Timer / Counter : In Microprocessor System, Timer is used to provide a delay and counter which is used to count incoming pulses. In large systems, where microprocessor's wastage time is critical, a separate timer IC like IC8254 can be used. These counters can work as counter or can provide accurate time delays. Microcontrollers have these timers on-chip.
3. Interrupt Controller : The various peripheral controllers on-chip in case of a microcontroller need to be properly prioritized and controlled properly. The microprocessors require external interrupt controller, while most of the microcontrollers have on-chip interrupt controller.
4. Serial and Parallel ports : The embedded system needs to interface with the external devices for which it needs ports for interfacing. The serial ports give an advantage of long distance data transfer at lower costs, while parallel communication gives faster data transfer for shorter distance.

5. Memory : In an embedded system. ROM (Read only memory) is used for storing program or code while RAM (Random Access Memory) is used for storing data. The ROM used is normally the EEPROM (Electrically Erasable and Programmable ROM). Some data which is a constant value is also sometimes stored in ROM instead of RAM. There are flash memories that are high speed memory.
6. Power Supply: The power supply provides power to various circuits in the system. Various voltages are required according to the application, most of the microcontrollers require +6V or +3V as Vcc.
7. Oscillator: Generally oscillator circuit generates clock pulses for internal operations to be synchronized. Some Microcontroller have on chip oscillator and only crystal is to be connected externally or entire oscillator circuit is external.
8. Reset Circuit: The Reset pin can be considered as an interrupt as its signal cannot be blocked. The Reset pin for 8051 microcontroller is active HIGH. Whenever power is switched ON. Positive going pulse should be present for two machine cycles on this pin.

Draw and explain different features of Raspberry Pi.



Features of Raspberry PI Model A

- The Model A raspberry pi features mainly includes
- 256 MB SDRAM memory
- Single 2.0 USB connector
- Dual Core Video Core IV Multimedia coprocessor
- HDMI (rev 1.3 & 1.4) Composite RCA (PAL and NTSC) Video Out
- 3.5 MM Jack, HDMI, Audio Out
- SD, MMC, SDIO Card slot on board storage
- Linux Operating system
- Broadcom BCM2835 SoC full HD multimedia processor
- 8.6cm\*5.4cm\*1.5cm dimensions

List and explain various components used in Raspberry Pi.

- **ARM CPU/GPU** -- This is a Broadcom BCM2835 System on a Chip (SoC) that's made up of an ARM central processing unit (CPU) and a Videocore 4 graphics processing unit (GPU). The CPU handles all the computations that make a computer work (taking input, doing calculations and producing output), and the GPU handles graphics output.
- **GPIO** -- These are exposed general-purpose input/output connection points that will allow the real hardware hobbyists the opportunity to tinker.
- **RCA** -- An RCA jack allows connection of analog TVs and other similar output devices.

- **Audio out** -- This is a standard 3.55-millimeter jack for connection of audio output devices such as headphones or speakers. There is no audio in.
- **LEDs** -- Light-emitting diodes, for all of your indicator light needs.
- **USB** -- This is a common connection port for peripheral devices of all types (including your mouse and keyboard). Model A has one, and Model B has two. You can use a USB hub to expand the number of ports or plug your mouse into your keyboard if it has its own USB port.
- **HDMI** -- This connector allows you to hook up a high-definition television or other compatible device using an HDMI cable.
- **Power** -- This is a 5v Micro USB power connector into which you can plug your compatible power supply.
- **SD cardslot** -- This is a full-sized SD card slot. An SD card with an operating system (OS) installed is required for booting the device. They are available for purchase from the manufacturers, but you can also download an OS and save it to the card yourself if you have a Linux machine and the wherewithal.
- **Ethernet** -- This connector allows for wired network access and is only available on the Model B.

**Draw and explain pin diagram of Raspberry Pi.**

Raspberry Pi B+, 2, 3 & Zero

3V3	1	2	5V	Key
GPIO2	3	4	5V	+
GPIO3	5	6	GND	Ground
GPIO4	7	8	GPIO14	UART
GND	9	10	GPIO15	I2C
GPIO17	11	12	GPIO18	SPI
GPIO27	13	14	GND	GPIO
GPIO22	15	16	GPIO23	Pin Number
3V3	17	18	GPIO24	
GPIO10	19	20	GND	
GPIO9	21	22	GPIO25	
GPIO11	23	24	GPIO8	
GND	25	26	GPIO7	
DNC	27	28	DNC	
GPIO5	29	30	GND	
GPIO6	31	32	GPIO12	
GPIO13	33	34	GND	
GPIO19	35	36	GPIO16	
GPIO26	37	38	GPIO20	
GND	39	40	GPIO21	



- **GPIO** is your standard pins that simply be used to turn devices on and off. For example, a LED.
- **I2C** (Inter-Integrated Circuit) pins allow you to connect and talk to hardware modules that support this protocol (I2C Protocol). This protocol will typically take up two pins.
- **SPI** (Serial Peripheral Interface Bus) pins can be used to connect and talk to SPI devices. Pretty much the same as I2C but makes use of a different protocol.
- **UART** (Universal asynchronous receiver/transmitter) is the serial pins used to communicate with other devices.
- **DNC** stands for do not connect.
- The **power pins** pull power directly from the Raspberry Pi.
- **GND** are the pins you use to ground your devices. It doesn't matter which pin you use as they are all connected to the same line.

## 25.Application areas in which raspberry pi can be used.

### 1.Desktop PC

The Raspberry Pi can make a useful desktop computer if set up properly. To turn your raspberry pi into a desktop PC you'll need other gadgets than RPi itself such as a screen, a mouse, a keyboard and if you want, an extra storage device.

You also need to install an operating system like Raspbian or any other operating systems available for RPi. Some apps such as email and web browsing apps are included and many other are available for RPi.

### 2.Media Center

You can have your media center application for your TV running on your Raspberry Pi. Media center allows you to organize and play media, including pictures, music and videos. You just need to use the Raspberry Pi and Kodi software.

### 3.Web Server

The Raspberry Pi board is a great alternative to launch lightweight Web Server. It can handle a small amount of traffic and you can learn web programming languages such as HTML, CSS, PHP and MySQL. It can even handle WordPress, if you want to launch your own blog/web site you can easily do it.

#### 4.Home Automation System

The Raspberry Pi is capable of hosting a powerful home automation application. You can attach sensors, a camera, relays, etc. And you can monitor and control your house remotely. To extend its capabilities you can also add Arduinos or other similar boards. This is what someone made and documented over at Instructables.

#### 5.VPN

A virtual private network (VPN). A VPN extends your own private network into public places, so even if you're using Starbucks' Wi-Fi connection, your Internet browsing stays encrypted and secure.

#### 6.Robotics

Build and control awesome robots with the Raspberry Pi. Some cool projects out there include control robotic arms (example), drones (check out this example), humanoid robots (check this example), etc. There are many examples of robots out there that you can do or adapt to build your own original robot.

#### 7.Game Server

Raspbian, default OS of pi comes with a special version of Minecraft game pre-installed. But, the applications of Raspberry Pi can be used as a game server as well. It is an excellent game server for Minecraft. If multiple Raspberry Pis are used, making one as a dedicated server, a great gaming experience can be achieved.

Other multiplayer network games can be set up on the Raspberry Pi.

## 8.Retro Gaming

Raspberry Pi is ideal as a retro gaming machine. It fits as one of the lightest components of a machine. Particularly, it's a version, The Raspberry Pi Zero can fit into small spaces for gaming projects. There are two main options, Recalbox and RetroPie. Other platforms can be emulated too. Classic MS-DOS PC gaming and Commodore 64 can also be set-up and also many other popular 16-bit games consoles.

## 26.Write python code to blink LED with raspberry Pi.

1. Import RPi.GPIO as GPIO # Import Raspberry Pi GPIO library
2. From time import sleep # Import the sleep function from the time module
3. GPIO.setwarnings(False) # Ignore warning for now
4. GPIO.setmode(GPIO.BOARD) # Use physical pin numbering
5. GPIO.setup(8, GPIO.OUT, initial=GPIO.LOW) # Set pin 8 to be an output pin and set initial value to low (off)
6. While True: # Run forever
7. GPIO.output(8, GPIO.HIGH) # Turn on
8. Sleep(1) # Sleep for 1 second
9. GPIO.output(8, GPIO.LOW) # Turn off
10. Sleep(1) # Sleep for 1 second

## 27.How to play sound using raspberry pi?Provide python code.

Depending on your version of Raspbian, you may or may not have to install the pygame package (e.g. Raspbian Lite does not come with some Python packages pre-installed). In a terminal, enter the following:

Sudo apt-get update

Sudo apt-get install python3-pygame

In a new file, enter the following code:

```

import time

import RPi.GPIO as GPIO

from pygame import mixer

# Pins definitions

Btn_pin = 4

# Set up pins

GPIO.setmode(GPIO.BCM)

GPIO.setup(btn_pin, GPIO.IN)

# Initialize pygame mixer

Mixer.init()

# Remember the current and previous button states

Current_state = True

Prev_state = True

# Load the sounds

Sound = mixer.Sound('applause-1.wav')

# If button is pushed, light up LED

Try:

    While True:

        Current_state = GPIO.input(btn_pin)

        If (current_state == False) and (prev_state == True):

            Sound.play()

            Prev_state = current_state

# When you press ctrl+c, this will be called

Finally:

    GPIO.cleanup()

```

Save the file (e.g. applause.py), and start the program with `python applause.py`. Push the button, and you should hear some congratulatory sounds!

## **28. Draw Circuit Diagram for connecting LED with Pi module and explain.**

The first step in this project is to design a simple LED circuit. Then we will make the LED circuit controllable from the Raspberry Pi by connecting the circuit to the general purpose input/output (GPIO) pins on the Raspberry Pi.

A simple LED circuit consists of a LED and resistor. The resistor is used to limit the current that is being drawn and is called a current limiting resistor. Without the resistor the LED would run at too high of a voltage, resulting in too much current being drawn which in turn would instantly burn the LED, and likely also the GPIO port on the Raspberry Pi.

When hooking up the circuit note the polarity of the LED. You will notice that the LED has a long and short lead. The long lead is the positive side also called the anode, the short lead is the negative side called the cathode. The long should be connected to the resistor and the short lead should be connected to ground via the blue jumper wire and pin 6 on the Raspberry Pi as shown on the diagram.

To find the pin number refer to this diagram showing the physical pin numbers on the Raspberry Pi.

## **29. Define the following with respect to PWM(Pulse Width Modulation):-**

TON, TOFF, period, duty cycle.

## **30. Provide python code for controlling the brightness of LED**

**Sol:-**

```
Import RPi.GPIO as GPIO
```

```
Import time
```

```
# Led1 on my Board
```

```
Led = 11
```

```
GPIO.setmode( GPIO.BOARD)
```

```
GPIO.setup( led, GPIO.OUT)
```

```
# 50Hz PWM Frequency
```

```

Pwm_led = GPIO.PWM( led, 50)
# Full Brightness, 100% Duty Cycle
Pwm_led.start(100)
Try:
While True:
    Duty_s = raw_input("Enter Brightness Value (0 to 100):")
    # Convert into Integer Value
    Duty = int(duty_s)
    Pwm_led.ChangeDutyCycle(duty)
    Time.sleep(0.5)
Except KeyboardInterrupt:
    Print "Exiting Program"
except:
    print "Error Occurs, Exiting Program"
finally:
    GPIO.cleanup()

```

### **31.Explain the purpose of following linux commands.**

ls Command

ls command is used to list contents of a directory. It works more or less like dir command.

The -l option enables long listing format like this.

```
$ ls -l file1
```

mkdir Command

mkdir command is used to create single or more directories, if they do not already exist (this can be overridden with the -p option).

```
$ mkdir tecmint-files
```

OR

```
$ mkdir -p tecmint-files
```

cd

cd command is used to change the present working directory.

```
$ cd Documents
```

pwd Command

pwd command displays the name of current/working directory as below.

```
$ pwd
```

echo Command

echo command prints a text of line provided to it.

```
$ echo "SYBSc CS"
```

cp Command

cp command is used for copying files and directories from one location to another.

```
$ cp /home/tecmint/file1 /home/tecmint/Personal/
```

mv Command

mv command is used to rename files or directories. It also moves a file or directory to another location in the directory structure.

```
$ mv test.sh sysinfo.sh
```

rm command

rm command is used to remove files or directories as shown below.

```
$ rm file1
```

```
$ rm -rf my-files
```

grep Command

grep command searches for a specified pattern in a file (or files) and displays in output lines containing that pattern as follows.

```
$ grep 'tecmint' domain-list.txt
```

cat Command

cat command is used to view contents of a file or concatenate files, or data provided on standard input, and display it on the standard output.

```
$ cat file.txt
```

#### chmod Command

chmod command is used to change/update file access permissions like this.

```
$ chmod +x sysinfo.sh
```

#### chown Command

chown command changes/updates the user and group ownership of a file/directory like this.

```
$ chmod -R www-data:www-data /var/www/html
```

#### ssh Command

ssh (SSH client) is an application for remotely accessing and running commands on a remote machine. It is designed to offer a secure encrypted communications between two untrusted hosts over an insecure network such as the Internet.

```
$ ssh tecmint@192.168.56.10
```

#### tar Command

tar command is a most powerful utility for archiving files in Linux.

```
$ tar -czf home.tar.gz .
```

#### man Command

man command is used to view the on-line reference manual pages for commands/programs like so.

```
$ man du
```

```
$ man df
```

#### df Command

df command is used to show file system disk space usage as follows.

```
$ df -h
```

#### head Command

head command is used to show first lines (10 lines by default) of the specified file or stdin to the screen:

```
# ps -eo pid,ppid,cmd,%mem,%cpu --sort=-%mem | head
```

#### tail Command



tail command is used to display the last lines (10 lines by default) of each file to standard output.

If there more than one file, precede each with a header giving the file name. Use it as follow (specify more lines to display using -n option).

```
$ tail long-file
```

OR

```
$ tail -n 15 long-file
```

touch Command

touch command changes file timestamps, it can also be used to create a file as follows.

```
$ touch file.txt
```

sed Command

sed command is used as a stream editor.

```
$ sed s/linux/*geekfile.txt
```

### **32.List and explain various features of python.**

**Sol:**

**Python provides lots of features that are listed below.**

#### **1) Easy to Learn and Use**

**Python is easy to learn and use. It is developer-friendly and high level programming language.**

#### **2) Expressive Language**

**Python language is more expressive means that it is more understandable and readable.**

#### **3) Interpreted Language**

**Python is an interpreted language i.e. interpreter executes the code line by line at a time. This makes debugging easy and thus suitable for beginners.**

#### **4) Cross-platform Language**

**Python can run equally on different platforms such as Windows, Linux, Unix and Macintosh etc. So, we can say that Python is a portable language.**

#### **5) Free and Open Source**

Python language is freely available at official web address. The source-code is also available. Therefore it is open source.

#### 6) Object-Oriented Language

Python supports object oriented language and concepts of classes and objects come into existence.

#### 7) Extensible

It implies that other languages such as C/C++ can be used to compile the code and thus it can be used further in our python code.

#### 8) Large Standard Library

Python has a large and broad library and provides rich set of module and functions for rapid application development.

#### 9) GUI Programming Support

Graphical user interfaces can be developed using Python.

#### 10) Integrated

It can be easily integrated with languages like C, C++, JAVA etc.

**Q33. Explain the purpose of following files in Linux file system.**

`\boot \bin \dev \etc \home \lib \media  
\mnt \opt \proc \sys \sbin \tmp \usr \var`

Ans:

1. In Linux, and other Unix-like operating systems, the /boot/ directory holds files used in booting the operating system. The usage is standardized in the Filesystem Hierarchy Standard.  
It contains :  
`vmlinux` – the Linux kernel  
`initrd.img` – a temporary file system, used prior to loading the kernel  
`System.map` – a symbol lookup table
2. `bin` file is a self extracting binary file for Linux and Unix-like operating systems. Before executing a `.bin` file you need to give it executive permissions. ... If you like to practise terminal commands then open a new terminal and run the following command to give the file execute permissions.
3. `/dev` is the location of special or device files. It is a very interesting directory that highlights one important aspect of the Linux filesystem - everything is a file or a directory.
4. `/etc` - Usually contain the configuration files for all the programs that run on your Linux/Unix system. `/opt` - Third party application packages which does not conform to the standard Linux file hierarchy can be installed here.

5. A standard subdirectory of the root directory, /home has the sole purpose of containing users' home directories. The root directory, which is designated by a forward slash ( / ), is the directory that contains all other directories and their subdirectories as well as all files on the system.
6. The lib folder is a library files directory which contains all helpful library files used by the system. In simple terms, these are helpful files which are used by an application or a command or a process for their proper execution. The commands in /bin or /sbin dynamic library files are located just in this directory.
7. /media – Removable Media: The /media directory contains subdirectories where removable media devices inserted into the computer are mounted. For example, when you insert a CD into your Linux system, a directory will automatically be created inside the /media directory.
8. The /mnt directory and its subdirectories are intended for use as the temporary mount points for mounting storage devices, such as CDROMs, floppy disks and USB (universal serial bus) key drives. /mnt is a standard subdirectory of the root directory on Linux and other Unix-like operating systems, along with directories such as /bin, /boot, /dev, /etc, /home, /proc, /root, /sbin, /usr and /var.
9. /opt is for “the installation of add-on application software packages”.
10. Proc file system (procfs) is virtual file system created on fly when system boots and is dissolved at time of system shut down. It contains the useful information about the processes that are currently running, it is regarded as control and information centre for kernel.
11. /sys is an interface to the kernel. Specifically, it provides a filesystem-like view of information and configuration settings that the kernel provides, much like /proc . Writing to these files may or may not write to the actual device, depending on the setting you're changing.
12. /sbin is a standard subdirectory of the root directory in Linux and other Unix-like operating systems that contains executable (i.e., ready to run) programs. They are mostly administrative tools, that should be made available only to the root (i.e., administrative) user.
13. The /tmp directory contains mostly files that are required temporarily, it is used by different programs to create lock files and for temporary storage of data. Many of these files are important for currently running programs and deleting them may result in a system crash.
14. The /usr/local hierarchy is for use by the system administrator when installing software locally. It needs to be safe from being overwritten when the system software is updated. It may be used for programs and data that are shareable amongst a group of hosts, but not found in /usr.
15. /var is a standard subdirectory of the root directory in Linux and other Unix-like operating systems that contains files to which the system writes data during the course of its operation

**Q34. Explain the following with GPIO pins:**

1. Need
2. Use of GPIO
3. Features of GPIO

Ans:

1. Stands for "General Purpose Input/Output." GPIO is a type of pin found on an integrated circuit that does not have a specific function. While most pins have a dedicated purpose, such as sending a signal to a certain component, the function of a GPIO pin is customizable and can be controlled by software. They are also used by system-on-chip (SOC) circuits, which include a processor, memory, and external interfaces all on a single chip. GPIO pins allow these chips to be configured for different purposes and work with several types of components.

2. The Raspberry Pi's GPIO Pins: GPIO stands for General Purpose Input Output. It is a way the Raspberry Pi can control and monitor the outside world by being connected to electronic circuits. The Raspberry Pi is able to control LEDs, turning them on or off, or motors, or many other things. GPIO is your standard pins that can be used to turn devices on and off. For example, a LED. I2C (Inter-Integrated Circuit) pins allow you to connect and talk to hardware modules that support this protocol (I2C Protocol). This protocol will typically take up two pins.
3. GPIO has the following user-configurable features:
  - Up to 32 GPIO
  - 8 GPIO with Analog channels for SAADC, COMP or LPCOMP inputs
  - Configurable output drive strength
  - Internal pull-up and pull-down resistors
  - Wake-up from high or low level triggers on all pins
  - Trigger interrupt on state changes on any pin
  - All pins can be used by the PPI task/event system
  - One or more GPIO outputs can be controlled through PPI and GPIOTE channels
  - All pins can be individually mapped to interface blocks for layout flexibility
  - GPIO state changes captured on SENSE signal can be stored by LATCH register

**Q35. In order to setup Raspberry pi what all components are needed? (hint: hardware /software/OS)**

Ans: Here are the various components needed to setup Raspberry Pi:

- ARM CPU/GPU -- This is a Broadcom BCM2835 System on a Chip (SoC) that's made up of an ARM central processing unit (CPU) and a Videocore 4 graphics processing unit (GPU). The CPU handles all the computations that make a computer work (taking input, doing calculations and producing output), and the GPU handles graphics output.
- GPIO -- These are exposed general-purpose input/output connection points that will allow the real hardware hobbyists the opportunity to tinker.
- RCA -- An RCA jack allows connection of analog TVs and other similar output devices.
- Audio out -- This is a standard 3.55-millimeter jack for connection of audio output devices such as headphones or speakers. There is no audio in.
- LEDs -- Light-emitting diodes, for all of your indicator light needs.
- USB -- This is a common connection port for peripheral devices of all types (including your mouse and keyboard). Model A has one, and Model B has two. You can use a USB hub to expand the number of ports or plug your mouse into your keyboard if it has its own USB port.
- HDMI -- This connector allows you to hook up a high-definition television or other compatible device using an HDMI cable.
- Power -- This is a 5v Micro USB power connector into which you can plug your compatible power supply.

- SD cardslot -- This is a full-sized SD card slot. An SD card with an operating system (OS) installed is required for booting the device. They are available for purchase from the manufacturers, but you can also download an OS and save it to the card yourself if you have a Linux machine and the wherewithal.
- Ethernet -- This connector allows for wired network access and is only available on the Model B.

**Q36. Write a code to record a video in camera.**

Ans:

```
import picam
import time
picam.recordVideoWithDetails('./foo.h264',640,480,10000)
```

**37) Write a short note on cross compilation.**

**Cross-compilation** is the act of **compiling** code for one computer system (often known as the target) on a different system, called the host. It's a very useful technique, for instance when the target system is too small to host the **compiler** and all relevant files.

A **cross compiler** is a [compiler](#) capable of creating [executable](#) code for a [platform](#) other than the one on which the compiler is running. For example, a compiler that runs on a [Windows 7 PC](#) but generates code that runs on [Android smartphone](#) is a cross compiler.

The fundamental use of a cross compiler is to separate the build environment from target environment. This is useful in several situations:

- [Embedded computers](#) where a device has extremely limited resources. For example, a microwave oven will have an extremely small computer to read its touchpad and door sensor, provide output to a digital display and speaker, and to control the machinery for cooking food. This computer will not be powerful enough to run a compiler, a file system, or a development environment. Since debugging and testing may also require more resources than are available on an embedded system, cross-compilation can be less involved and less prone to errors than native compilation.
- Compiling for multiple machines. For example, a company may wish to support several different versions of an operating system or to support several different operating systems. By using a cross compiler, a single build environment can be set up to compile for each of these targets.

### 38) Compare I2C and SPI.

I2C	SPI
I2C can be multi-master and multi-slave, which means there can be more than one master and slave attached to the I2C bus	SPI can be multi-slave but does not a multi-master serial protocol, that means there can be only one master attached to SPI bus.
I2C is half-duplex communication protocol.	SPI is a full duplex communication protocol.
I2C has the feature of clock stretching, that means if the slave cannot able to send fast data as fast enough then it suppresses the clock to stop the communication.	Clock stretching is not the feature of SPI.
I2C is used only two wire for the communication, one wire is used for the data and the second wire is used for the clock.	SPI needs three or four wire for communication ((depends on requirement), MOSI, MISO, SCL and Chip-select pin.
I2C is slower than SPI.	In comparison to I2C, SPI is faster.
I2C draws more power than SPI.	Draws less power as compared to I2C.

I2C is less susceptible to noise than SPI	SPI is more susceptible to noise than I2C.
I2C is cheaper to implement than the SPI communication protocol.	Costly as compare to I2C.
I2C work on wire and logic and it has a pull-up resistor.	There is no requirement of pull-up resistor in case of the SPI.
In I2C communication we get the acknowledgment bit after each byte.	Acknowledgment bit is not supported by the SPI communication protocol.
I2C ensures that data sent is received by the slave device.	SPI does not verify that data is received correctly or not.
I2C support the multi-master communication.	SPI does not support multi -master communication.
I2C is a multi-master communication protocol that's why it has the feature of arbitration.	SPI is not a multi-master communication protocol, so it does not consist the properties of arbitration.
I2C is the address base bus protocol, you have to send the address of the slave for the communication.	In case of the SPI, you have to select the slave using the slave select pin for the communication.
I2C has some extra overhead due to start and stop bits.	SPI does not have a start and stop bits.

---

I2C supports multiple devices on the same bus without any additional select lines (work on the basis of device address).

SPI requires additional signal (slave select lines) lines to manage multiple devices on the same bus.

---

I2C is better for long distance.

SPI is better for the short distance.

---

I2C is developed by NXP.

SPI is developed by Motorola.

### 39)&40) Steps to connect motor to pi.

To get a motor running, you will need:

- A Raspberry Pi with SD card preinstalled with Raspbian
- A Breadboard to connect everything on
- An L293 or SN755410 motor driver chip (I will refer both as L293D in this tutorial)
- Jumper cables to connect everything up (Male to male and female to male)
- One or two DC motors rated for 6v
- 4x AA batteries and holder

There are 26 pins grouped in two rows of 13, and these collectively are called the *General Purpose Input Output header* or *GPIO* for short. These are a mix of four power pins, five ground pins and 17 data pins.

### Assembling the Circuit

#### Adding Power and Ground

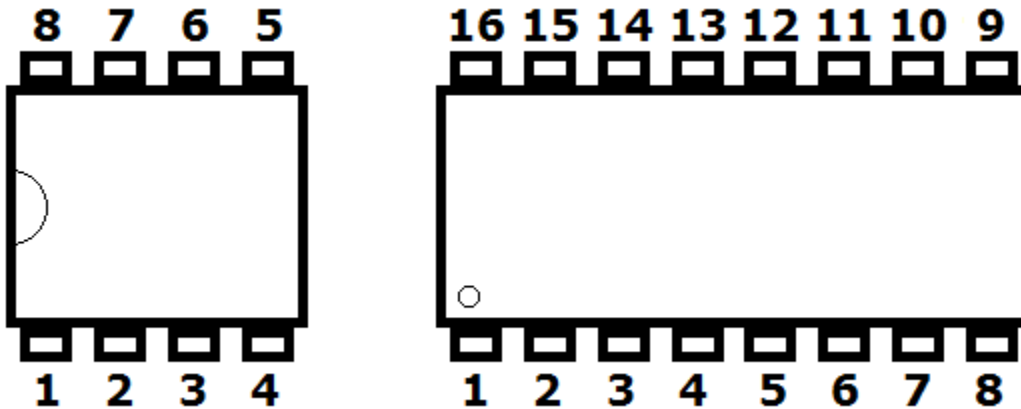
It is important to do this while the power to the Raspberry Pi is off, or disconnected, as you want to avoid shorting any connectors by mistake.

The first thing you need to do is connect up the power and ground wires. As with most electronics projects, everything that connects together will require a common ground.

The ground on the Raspberry Pi is physical **pin 6**. Starting at the top left with **pin 3V3**, counting left to right so **5V** is **pin 2**, **GPIO 2** (labelled **2**) is **pin 3** and so on.

Reading pin numbers on Integrated Circuit (IC) chips is easily done by having the notch or dot to the left then starting from bottom left gives us **pin 1**.





Pin 1 is at the bottom left

Adding the Data Wires

Now add three wires from the **GPIO** pins to the **L293D**.

- GPIO 25–Pin 22 > L293D–Pin 1
- GPIO 24–Pin 18 > L293D–Pin 2
- GPIO 23–Pin 16 > L293D–Pin 7

Add the motor:

- Motor–wire 1 > L293D–pin 3
- Motor–wire 2 > L293D–pin 6

It is extremely important that you double-check every connection before adding the batteries. Only when you are happy that everything is in place, connect the battery wires to the power rails of the breadboard.

# 1. Explain commands which are required for controlling/configuring camera.

- Open up your Raspberry Pi Camera module.

-Install the Raspberry Pi Camera module by inserting the cable into the Raspberry Pi. The cable slots into the connector situated between the Ethernet and HDMI ports, with the silver connectors facing the HDMI port.

- Boot up your Raspberry Pi.

-From the prompt, run "**sudo raspi-config**".

-If the "camera" option is not listed, you will need to run a few commands to update your Raspberry Pi.

-Run `sudo apt-get update` and `sudo apt-get upgrade`.

-Run `sudo raspi-config` again - you should now see the "camera" option.



-Navigate to the "camera" option, and enable it. Select "Finish" and reboot your Raspberry Pi.

-`raspistill` is a command line application that allows you to capture images with your camera module.

-To capture an image in jpeg format, type `raspistill -o image.jpg` at the prompt, where "image" is the name of your image.

## 2. What is SPI? Explain the working.

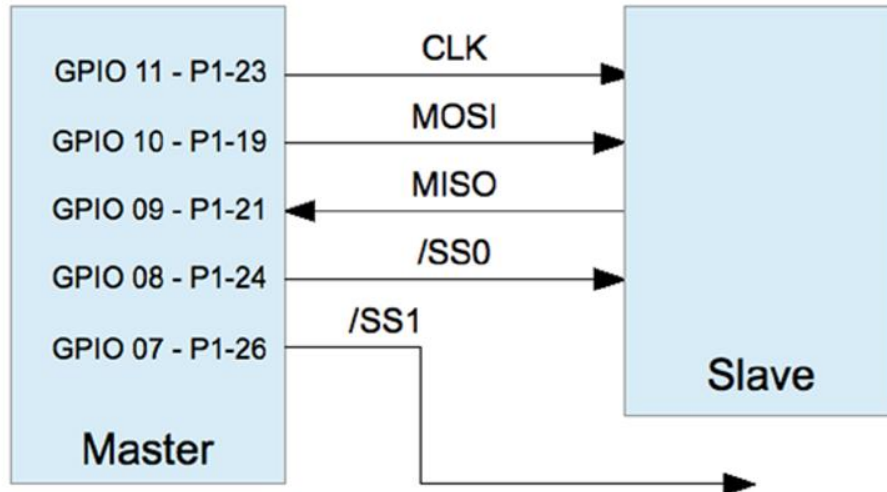
-The Serial Peripheral Interface bus, known affectionately as spy, is a synchronous serial interface that was named by Motorola.

- The SPI protocol operates in full-duplex mode, allowing it to send and receive data simultaneously.

-Devices on the SPI bus communicate on a master/slave basis.

-Multiple slaves coexist on a given SPI bus, with each slave being selected for communication by a slave select signal (also known as chip select).

-The following figure shows the Raspberry Pi as the master communicating with a slave:



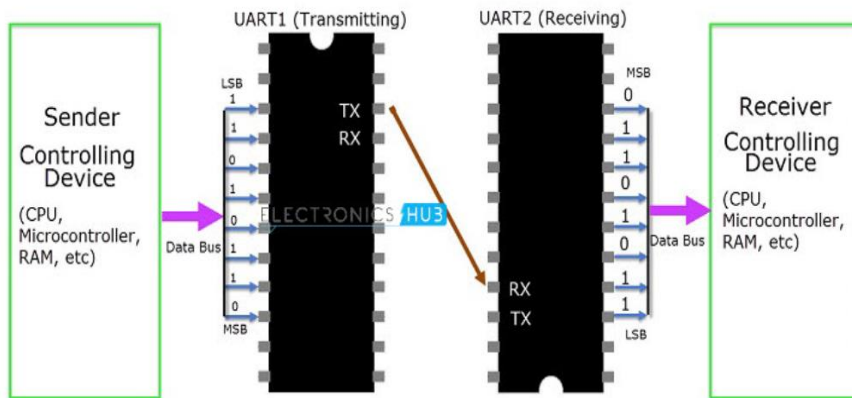
- Many SPI devices support only 8-bit transfers, while others are more flexible.
- The SPI bus is a de facto standard, meaning that there is no standard for data transfer width and SPI mode.
- The SPI controller can also be configured to transmit the most significant or the least significant bit first.

---Working:

- Data is transmitted from the master to the slave by using the MOSI line (master out, slave in).
- As each bit is being sent out by the master, the slave sends data bits on the MISO line (master in, slave out).
- Bits are shifted out of the master and into the slave.
- Simultaneously, bits are shifted out of the slave and into the master.
- Both transfers occur to the beat of the system clock (CLK).

### 3. What UART is needed in Pi? What all operations it can perform?

- UART stands for Universal Asynchronous Receiver/Transmitter.
- It is a physical circuit in a microcontroller, or a stand-alone IC.
- A UART's main purpose is to transmit and receive serial data.
- One of the best things about UART is that it only uses two wires to transmit data between devices.
- In UART communication, two UARTs communicate directly with each other.



-The Raspberry Pi supports two UARTs- UART0 and UART1

----operations:

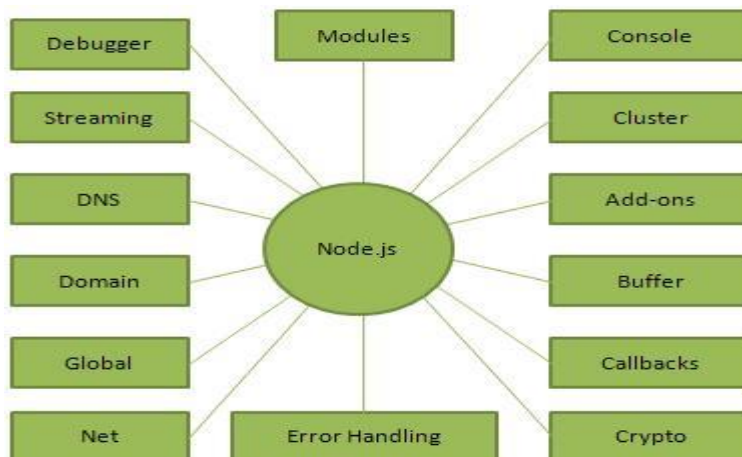
- The transmitting UART converts parallel data from a controlling device like a CPU into serial form, transmits it in serial to the receiving UART, which then converts the serial data back into parallel data for the receiving device.

-Data flows from the Tx pin of the transmitting UART to the Rx pin of the receiving UART

#### 4. State and Explain concept of Node.js. Write features as well.

##### Concepts

The following diagram depicts some important parts of Node.js.



##### BUFFER:

Pure JavaScript is Unicode friendly, but it is not so for binary data. While dealing with TCP streams or the file system, it's necessary to handle octet streams. Node provides Buffer class which provides instances to store raw data similar to an array of integers but corresponds to a

raw memory allocation outside the V8 heap. Buffer class is a global class that can be accessed in an application without importing the buffer module.

### CALLBACKS:

Callback is an asynchronous equivalent for a function. A callback function is called at the completion of a given task. Node makes heavy use of callbacks. All the APIs of Node are written in such a way that they support callbacks.

### STREAMS:

Streams are objects that let you read data from a source or write data to a destination in continuous fashion. In Node.js, there are four types of streams –

- **Readable** – Stream which is used for read operation.
- **Writable** – Stream which is used for write operation.
- **Duplex** – Stream which can be used for both read and write operation.
- **Transform** – A type of duplex stream where the output is computed based on input.

### GLOBAL:

Node.js global objects are global in nature and they are available in all modules. We do not need to include these objects in our application, rather we can use them directly. These objects are modules, functions, strings and object itself.

### MODULES:

There are several utility modules available in Node.js module library. These modules are very common and are frequently used while developing any Node based application.

Sr.No.	Module Name & Description
1	<u>OS Module</u> : Provides basic operating-system related utility functions.
2	<u>Path Module</u> : Provides utilities for handling and transforming file paths.

3	<u>Net Module</u> : Provides both servers and clients as streams. Acts as a network wrapper.
4	<u>DNS Module</u> : Provides functions to do actual DNS lookup as well as to use underlying operating system name resolution functionalities.
5	<u>Domain Module</u> : Provides ways to handle multiple different I/O operations as a single group.

## Features of Node.js

Following are some of the important features that make Node.js the first choice of software architects.

- **Asynchronous and Event Driven** – All APIs of Node.js library are asynchronous, that is, non-blocking. It essentially means a Node.js based server never waits for an API to return data. The server moves to the next API after calling it and a notification mechanism of Events of Node.js helps the server to get a response from the previous API call.
- **Very Fast** – Being built on Google Chrome's V8 JavaScript Engine, Node.js library is very fast in code execution.
- **Single Threaded but Highly Scalable** – Node.js uses a single threaded model with event looping. Event mechanism helps the server to respond in a non-blocking way and makes the server highly scalable as opposed to traditional servers which create limited threads to handle requests. Node.js uses a single threaded program and the same program can provide service to a much larger number of requests than traditional servers like Apache HTTP Server.
- **No Buffering** – Node.js applications never buffer any data. These applications simply output the data in chunks.
- **License** – Node.js is released under the MIT license.

---

## Q) Why is Linux used in a Raspberry Pi?

---

- Linux is an operating system used in almost all Raspberry Pi projects. This is because Linux is loaded with rich features such as portability, modularity, scalability which allows users to add or remove different functional blocks, scale the software according to the user's needs.

- Linux provides an open source ecosystem to the Raspberry pi, allowing the Raspberry Pi Foundation to contribute to various open source projects as well as releasing much of its own software as open source. These projects can often reduce the cost of hardware by taking advantage of multi-tasking feature of Linux.
- Linux is versatile, durable, secure. It allows unlimited modifications in the source code. i.e we can re-use components multiple times. Software is of free cost as the utilities are provided by Linux. There are many forums, blogs available for support. Most of the vendors provide support to Linux based embedded products.

“Raspbian” is a version of Linux built specifically for the Raspberry Pi. It comes packed with all the software you’ll need for every basic task with a computer. You’ll get an office suite, a web browser, email program, and some tools to learn programming.

---

### Q) Draw and explain the pin diagram of Raspberry pi

---

Raspberry Pi B+, 2, 3 & Zero

3V3	1	2	5V	<table><tr><th>Key</th></tr><tr><td>+</td></tr><tr><td>Ground</td></tr><tr><td>UART</td></tr><tr><td>I2C</td></tr><tr><td>SPI</td></tr><tr><td>GPIO</td></tr><tr><td>Pin Number</td></tr></table>	Key	+	Ground	UART	I2C	SPI	GPIO	Pin Number
Key												
+												
Ground												
UART												
I2C												
SPI												
GPIO												
Pin Number												
GPIO2	3	4	5V									
GPIO3	5	6	GND									
GPIO4	7	8	GPIO14									
GND	9	10	GPIO15									
GPIO17	11	12	GPIO18									
GPIO27	13	14	GND									
GPIO22	15	16	GPIO23									
3V3	17	18	GPIO24									
GPIO10	19	20	GND									
GPIO9	21	22	GPIO25									
GPIO11	23	24	GPIO8									
GND	25	26	GPIO7									
DNC	27	28	DNC									
GPIO5	29	30	GND									
GPIO6	31	32	GPIO12									
GPIO13	33	34	GND									
GPIO19	35	36	GPIO16									
GPIO26	37	38	GPIO20									
GND	39	40	GPIO21									

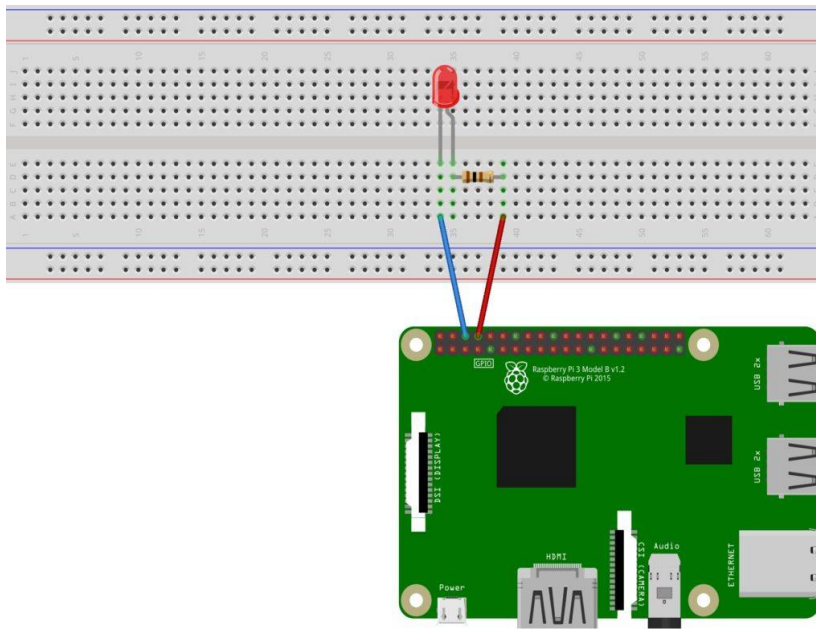
- GPIO is your standard pins that can be used to turn devices on and off. For example, a LED.
- I2C (Inter-Integrated Circuit) pins allow you to connect and talk to hardware modules that support this protocol (I2C Protocol). This protocol will typically take up two pins.

- SPI (Serial Peripheral Interface Bus) pins can be used to connect and talk to SPI devices. Pretty much the same as I2C but makes use of a different protocol.
- UART (Universal asynchronous receiver/transmitter) is the serial pins used to communicate with other devices.
- DNC stands for do not connect, this is pretty self-explanatory.
- The power pins pull power directly from the Raspberry Pi.
- GND are the pins you use to ground your devices. It doesn't matter which pin you use as they are all connected to the same line.

---

Q) Write python code to blink a LED.

---



- Raspberry Pi 3 setup with monitor and USB Mouse & Keyboard
- Solderless breadboard
- Jumper wires for easy hookup
- Resistor pack
- Red LED

Python code: **#initialise gpio points and turn LED on and off in 1 second interval**

import RPi.GPIO as GPIO **# Import Raspberry Pi GPIO library**



```
from time import sleep # Import the sleep function from the time module

GPIO.setwarnings(False) # Ignore warning for now

GPIO.setmode(GPIO.BOARD) # Use physical pin numbering

GPIO.setup(8, GPIO.OUT, initial=GPIO.LOW) # Set pin 8 to be an output pin and set initial
value to low (off)

# set the output pin to either high (on) or low (off). We do this inside a infinite loop so our
program keep executing until we manually stop it.

while True: # Run forever

    GPIO.output(8, GPIO.HIGH) # Turn on

    sleep(1) # Sleep for 1 second

    GPIO.output(8, GPIO.LOW) # Turn off

    sleep(1)
```

---

### **Q) Why is UART needed in a Raspberry Pi, what operations can it perform?**

---

UART (Universal Asynchronous Receiver/Transmitter) is a serial communication protocol in which data is transferred serially i.e. bit by bit.

UART serial communication protocol uses a defined frame structure for their data bytes. Frame structure in Asynchronous communication consists:

- **START bit:** It is a bit with which indicates that serial communication has started and it is always low.
- **Data bits packet:** Data bits can be packets of 5 to 9 bits. Normally we use 8-bit data packet, which is always sent after the START bit.
- **STOP bit:** This usually is one or two bits in length. It is sent after data bits packet to indicate the end of frame. Stop bit is always logic high.

Asynchronous transmission allows data to be transmitted without the sender having to send a clock signal to the receiver. Instead, the sender and receiver agree on timing parameters in advance and special bits called 'start bits' are added to each word and used to synchronize the sending and receiving units.

UART is commonly used on the Pi as a convenient way to control it over the GPIO, or access the kernel boot messages from the serial console (enabled by default). It can also be used as a way to interface an Arduino, etc with your Pi.

Function:

1. The transmitting UART receives data in parallel from the data bus, adds start bit, stop bit to data frame.
2. The entire packet is sent serially from the transmitting UART to the receiving UART.
3. The receiving UART discards the start bit and stop bit from the data frame and converts and transfers to data bus on receiving end.

### **Q1) Why one needs to use Linux as OS for Raspberry Pi?**

- It is an open source OS which gives a great advantage to the programmers as they can design their own custom operating systems.
- It gives you a lot of option of programs having some different features so you can choose according to your need.
- A global development community look at different ways to enhance its security, hence it is highly secured and robust so you don't need an anti-virus to scan it regularly. Companies like Google, Amazon and Facebook use Linux in order to protect their servers as it is highly reliable and stable.
- Portability: Portability doesn't mean it is smaller in file size or can be carried in pen drives or memory cards. It means that it supports different types of hardware.
- Generally, for Raspberry Pi a lighter OS is required as hardware is limited and as you can tailor Linux to your needs it is a perfect solution.

- Linux can be made free of code bloat which reduces the load on raspberry pi.
- Systems that run on Raspberry Pi must be compatible with its ARM processors, such as Linux.

**Q2) List and Explain various features of Python.**

- Python provides lots of features that are listed below.
- Python is easy to learn and use. It is developer-friendly and high-level programming language.
- Python language is more expressive means that it is more understandable and readable.
- Python is an interpreted language i.e. interpreter executes the code line by line at a time. This makes debugging easy and thus suitable for beginners.
- Python can run equally on different platforms such as Windows, Linux, Unix and Macintosh etc. So, we can say that Python is a portable language.
- Python language is freely available at official web address. The source-code is also available. Therefore, it is open source.

- Python supports object-oriented language and concepts of classes and objects come into existence.
- It implies that other languages such as C/C++ can be used to compile the code and thus it can be used further in our python code.
- Python has a large and broad library and provides rich set of module and functions for rapid application development.
- Graphical user interfaces can be developed using Python.
- It can be easily integrated with languages like C, C++, JAVA etc.

#### **Q4) List and Explain various components used in Raspberry Pi.**

Here are the various components on the Raspberry Pi board:

- ARM CPU/GPU – This is a Broadcom BCM2835 System on a Chip (SoC) that's made up of an ARM central processing unit (CPU) and a Video core 4 graphics processing unit (GPU). The CPU handles all the computations that make a computer work (taking input, doing calculations and producing output), and the GPU handles graphics output.
- GPIO – These are exposed general-purpose input/output connection points that will allow the real hardware hobbyists the opportunity to tinker.
- RCA – An RCA jack allows connection of analogue TVs and other similar output devices.
- Audio out – This is a standard 3.55-millimeter jack for connection of audio output devices such as headphones or speakers. There is no audio in.
- LEDs – Light-emitting diodes, for all of your indicator light needs.
- USB – This is a common connection port for peripheral devices of all types (including your mouse and keyboard). Model A has one, and Model B has two. You can use a USB hub to expand the number of ports or plug your mouse into your keyboard if it has its own USB port.

- HDMI – This connector allows you to hook up a high-definition television or other compatible device using an HDMI cable.
- Power – This is a 5v Micro USB power connector into which you can plug your compatible power supply.
- SD card slot – This is a full-sized SD card slot. An SD card with an operating system (OS) installed is required for booting the device. They are available for purchase from the manufacturers, but you can also download an OS and save it to the card yourself if you have a Linux machine and the wherewithal.
- Ethernet – This connector allows for wired network access and is only available on the Model B.

Many of the features that are missing, such as Wi-Fi and audio in, can be added using the USB port(s) or a USB hub as needed.

### **1. How do you play sound using raspberry pi? Provide python code.**

Required Tools:

- Raspberry Pi Setup with monitor, USB mouse and keyboard.
- Solderless breadboard.
- Jumper wires for easy hookup.
- Register pack.
- Speaker.

Recommended Reading:-

- Amixer – We will be using the amixer Linux tool to adjust the volume on our Raspberry Pi
- Pygame – Pygame is a framework that is used for making simple games in Python. Raspbian comes pre-loaded with Pygame, which means we can use it to play sounds.

Hardware Connections:-

- Connect GPIO12 (pin 32) to the 330Ω resistor, and the resistor to the LED
- Connect GPIO4 (pin 7) to the button
- Make the power (3.3 V) and ground (GND) connections as shown in the diagram.

Code:-

Import time

Import RPi.GPIO as GPIO

From pygame import mixer

```

# Pins definitions

Btn_pin = 4

# Set up pins

GPIO.setmode(GPIO.BCM)

GPIO.setup(btn_pin, GPIO.IN)

# Initialize pygame mixer

Mixer.init()

# Remember the current and previous button states

Current_state = True

Prev_state = True

# Load the sounds

Sound = mixer.Sound('applause-1.wav')

# If button is pushed, light up LED

Try:

    While True:

        Current_state = GPIO.input(btn_pin)

        If (current_state == False) and (prev_state == True):

            Sound.play()

            Prev_state = current_state

# When you press ctrl+c, this will be called

Finally:

    GPIO.cleanup()

```

## **2. State and explain the concept of Node.js. Write features of the same.**

Node.js is an open source, cross-platform runtime environment for developing server-side and networking applications. Node.js applications are written in JavaScript, and can be run within the Node.js runtime on OS X, Microsoft Windows, and Linux.

Node.js also provides a rich library of various JavaScript modules which simplifies the development of web applications using Node.js to a great extent.

Node.js is a platform built on Chrome's JavaScript runtime for easily building fast and scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.

### **Features of Node.js**

Following are some of the important features that make Node.js the first choice of software architects.

- **Asynchronous and Event Driven** – All APIs of Node.js library are asynchronous, that is, non-blocking. It essentially means a Node.js based server never waits for an API to return data. The server moves to the next API after calling it and a notification mechanism of Events of Node.js helps the server to get a response from the previous API call.
- **Very Fast** – Being built on Google Chrome's V8 JavaScript Engine, Node.js library is very fast in code execution.
- **Single Threaded but Highly Scalable** – Node.js uses a single threaded model with event looping. Event mechanism helps the server to respond in a non-blocking way and makes the server highly scalable as opposed to traditional servers which create limited threads to handle requests. Node.js uses a single threaded program and the same program can provide service to a much larger number of requests than traditional servers like Apache HTTP Server.
- **No Buffering** – Node.js applications never buffer any data. These applications simply output the data in chunks.
- **License** – Node.js is released under the MIT license.

### **3. Draw a connection diagram for connecting LED with PI. Write python code**

**To blink LED with a duration of 4 units.**

Code:-

#To initialize the GPIO ports on the Raspberry Pi we need to first import the Python library, the initialize the library and setup pin 8 as an output pin.

```
Import RPi.GPIO as GPIO # Import Raspberry Pi GPIO library
```

```
From time import sleep # Import the sleep function from the time module
```

```
GPIO.setwarnings(False) # Ignore warning for now
```

```
GPIO.setmode(GPIO.BOARD) # Use physical pin numbering
```

```
GPIO.setup(8, GPIO.OUT, initial=GPIO.LOW) # Set pin 8 to be an output pin and set initial value to low (off)
```

#Next we need to turn the LED on and off in 4 second intervals by setting the output pin to either high (on) or low (off). We do this inside a infinite loop so our program keep executing until we manually stop it.

```
While True: # Run forever
```

```
GPIO.output(8, GPIO.HIGH) # Turn on
```

```
Sleep(4) # Sleep for 4 second
```

```
GPIO.output(8, GPIO.LOW) # Turn off
```

```
Sleep(4) # Sleep for 4 second
```

#### **4. Priyank has joined MacroSYS organization as a Jr. programmer in the domain**

Of raspberry pi. On the very first day he received following task which needs to

Be done in Linux. Help him by giving solutions.

I) Creation of user called “ramu”

```
$sudo useradd ramu
```

```
$sudo passwd ramu
```

II) Extract first 15 lines from a /etc.

```
$head -15 /etc/passwd
```

III) List the information of all users connected to a system.

```
$who -H -a
```

IV) Find out the location where he is working currently.

```
$pwd
```

V) To create a folder called project.

```
$mkdir project
```



**Q. What do you mean by GPU? Explain the role of following with respect to GPU:**

- **Vertex Shader**
- **Geometry Shader**
- **Pixel Shader**

**Answer:**

Graphics Processing Unit (GPU):

1. A programmable logic chip (processor) specialized for display functions. The GPU renders images, animations and video for the computer's screen.
2. GPUs are located on plug-in cards, in a chipset on the motherboard or in the same chip as the CPU.
3. A GPU performs parallel operations. Although it is used for 2D data as well as for zooming and panning the screen, a GPU is essential for smooth decoding and rendering of 3D animations and video. The more sophisticated the GPU, the higher the resolution and the faster and smoother the motion in games and movies.
4. GPUs on stand-alone cards include their own memory, while GPUs built into the chipset or CPU chip share main memory with the CPU.

Vertex Shader:

1. The purpose is to transform each vertex's 3D position in virtual space to the 2D coordinate at which it appears on the screen.
2. The vertex shader is used to transform the attributes of vertices (points of a triangle) such as color, texture, position and direction from the original color space to the display space.
3. It allows the original objects to be distorted or reshaped in any manner.

Geometry Shader:

1. This type of shader can generate new graphics primitives, such as points, lines, and triangles.
2. Geometry shader programs are executed after vertex shaders. They take as input a whole primitive, possibly with adjacency information.
3. For example, when operating on triangles, the three vertices are the geometry shader's input. The shader can then emit zero or more primitives, which are rasterized and their fragments ultimately passed to a pixel shader.

Pixel Shader:

1. Pixel shaders are specialized shaders that are executed for each pixel of a bitmap. They are typically used to implement per-pixel effects.
2. Pixel shader effects in WPF are effects that one can apply to a UI element.
3. Pixel shader effects allow you to add adjustments such as glow, pixel brightness, red eye removal, and shadows, to rendered objects.

**Q. Currently world is moving towards M:M technology. Justify the impact of such technology in human life by providing at least three examples.**

**Answer:**

#### 1. MANUFACTURING

Every manufacturing environment—whether it's food processing or general product manufacturing—relies on technology to ensure costs are managed properly and processes are executed efficiently. Automating manufacturing processes within such a fast-paced environment is expected to improve processes even more. In the manufacturing world, this could involve highly automated equipment maintenance and safety procedures.

For example, M2M tools allow business owners to be alerted on their smartphones when an important piece of equipment needs servicing, so they can address issues as quickly as they arise. Sophisticated networks of sensors connected to the Internet could even order replacement parts automatically.

#### 2. HOME APPLIANCES

IoT already affects home appliance connectivity through platforms like Nest. However, M2M is expected to take home-based IoT to the next level. Manufacturers like LG and Samsung are already slowly unveiling smart home appliances to help ensure a higher quality of life for occupants.

For example, an M2M-capable washing machine could send alerts to the owners' smart devices once it finishes washing or drying, and a smart refrigerator could automatically order groceries from Amazon once its inventory is depleted. There are many more examples of home automation that can potentially improve quality of life for residents, including systems that allow members of the household to remotely control HVAC systems using their mobile devices. In situations where a homeowner decides to leave work early, he or she could contact the home heating system before leaving work to make sure the temperature at home will be comfortable upon arrival.

### 3. HEALTHCARE DEVICE MANAGEMENT

One of the biggest opportunities for M2M technology is in the realm of health care. With M2M technology, hospitals can automate processes to ensure the highest levels of treatment. Using devices that can react faster than a human healthcare professional in an emergency situation make this possible. For instance, when a patient's vital signs drop below normal, an M2M-connected life support device could automatically administer oxygen and additional care until a healthcare professional arrives on the scene. M2M also allows patients to be monitored in their own homes instead of in hospitals or care centers. For example, devices that track a frail or elderly person's normal movements can detect when he or she has had a fall and alert a healthcare worker to the situation.

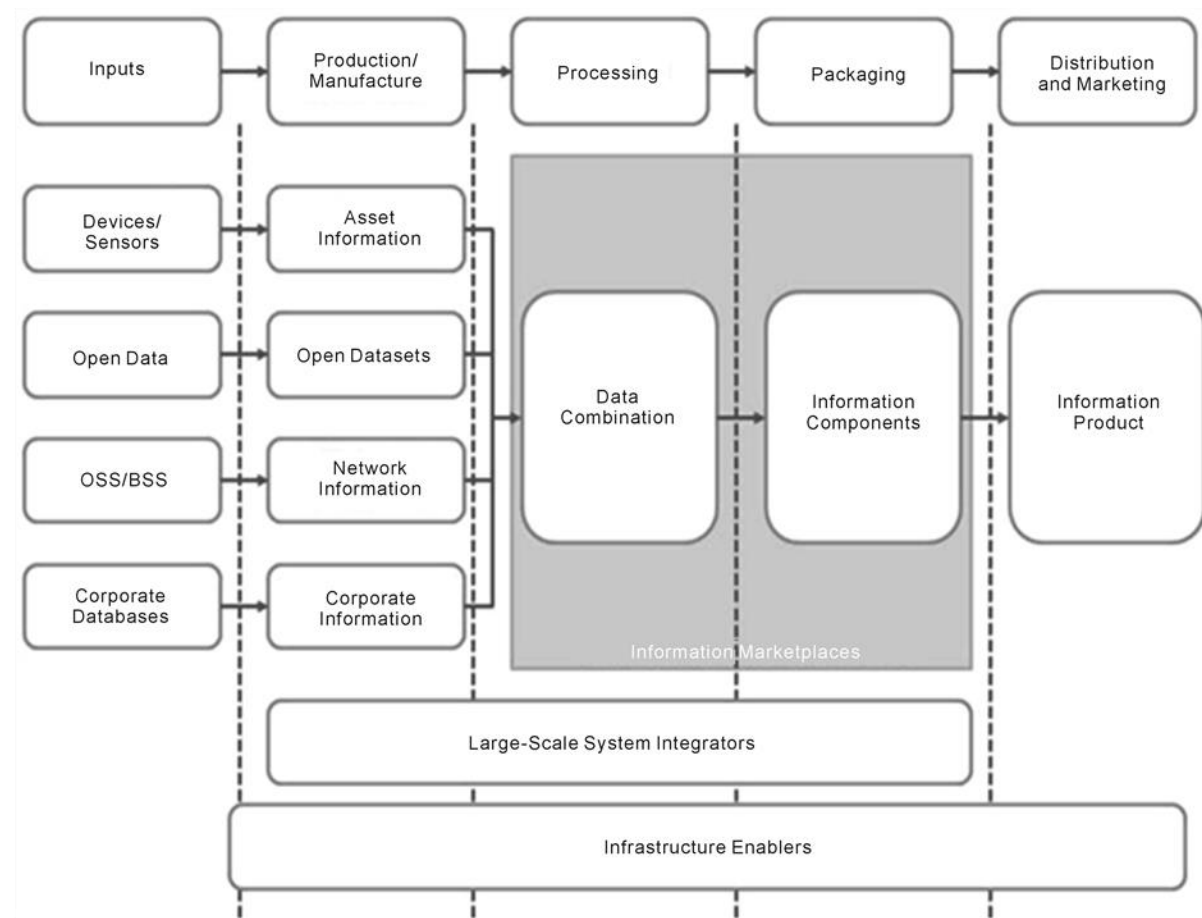
**Q. Justify the role of RFID in the field of IoT which helps in tracking inventory and improving customer satisfaction.**

**Answer:**

1. Improved visibility and faster scanning: Since RFID tags do not require a "line-of-sight" scan like barcodes, it is possible to read them at a distance for fast inventory processing. They can also be read in any orientation and give you improved visibility into your inventory with the potential for more frequent updates and scanning locations.
2. Reduced labor costs: With labor costs accounting for as much as 50-80% of distribution center costs, RFID offers potential benefits in this area. Inventory check-in, counting, and shipment verification can be done very quickly and automatically in a few scans without the need for multiple employees to process them. These savings must be weighed against the cost of investing in an RFID inventory solution, which we'll discuss in more detail below.
3. Tracking of returnable assets: For those companies that utilize a returnable fleet of assets such as containers and pallets, there is often a significant capital investment to protect. Utilizing RFID allows you to track these assets through the entire supply chain loop and provide increased visibility on inventory locations. This has the added benefit of improving returns and reducing theft or neglect.

**Q. Draw block diagram for Information-Driven Value Chain for IoT and explain.**

**Answer:**



**Q. What are the challenges for IOT implementation now and in future ?**

**1. Compatibility and Interoperability of Different IoT systems**

- As per the market analysts at McKinsey, 40% to 60% of the total values lies on our ability to achieve interoperability between different IoT systems. With numerous vendors, OEMs, and service providers, it becomes really difficult to maintain interoperability between different IoT systems.
- Sensors and Networking are the integral components of IoT. But not every machine is equipped with advanced sensors and networking capabilities to effectively communicate and share data. Besides, sensors of different power consumption

capabilities and security standards inbuilt in legacy machines may not be capable to provide the same results.

- A quick workaround could be to add external sensors, but this is also challenging because determining which function and which part will communicate and share data with the network is complex.

## 2. Identification and Authentication of Technologies

- According to a report, there are around 20 billion connected devices at present, and to connect all the devices involves a lot of security risks and not just complexity. Bringing along a large number of connected devices on one platform needs formalization and system architecture that can identify and authenticate those devices.

## 3. Integration of IoT Products with IoT Platforms

- For the successful implementation of IoT application, enterprises need to integrate various IoT connected products with right IoT platforms. Lack of proper integration could lead to abnormalities in functions and efficiency to deliver value to the customers.
- Research vice president at Gartner, Benoit Lheureux, says “Through 2018, half the cost of implementing IoT solutions will be spent integrating various IoT components with each other and back-end systems. It is vital to understand integration is a crucial IoT competency.”
- The major challenge here is too many IoT endpoints and asserts that need to be connected to aggregate the sensor data and transmit it to an IoT platform. Only with deep integration, companies can mine the huge data through Big Data technique to generate insight and to predict the outcomes.

## 4. Connectivity

- It is the part of networking challenges, as the Internet is still not available everywhere at the same speed. A global mobile satellite company Inmarsat revealed that 24% finds connectivity issue as the one of the biggest challenges in IoT deployment.
- Specifically, Logistics and Oil & Gas companies engaged in remote operations require robust communication networks to collect data in tough conditions and transmit back to the centre for analysis.
- The quality of signals collected by the sensors and to transmit over to the Networks largely depend upon the routers, LAN, MAN, and WAN.
- These networks have to be well-connected through different technologies to facilitate quick and quality communication. But the number of connected devices is growing at a much higher rate than the network coverage, which creates monitoring and tracking problems.

## 5. Delivering Value

- According to Forbes Insights Survey, 29% executives feel major challenge in building IoT capabilities is the quality of IoT technology.
- This data reveals the struggle of IoT application development companies in bringing the value for their consumers. So, before plunging into the development of IoT applications, an enterprise must clearly define what value they are going to deliver through what capabilities. And how their solution will enhance the efficiency and productivity, while also generating customer-satisfaction.
- As the IoT is all about “connected things”, the IoT projects also require a high level of assistance throughout the way. Around 50% of companies with IoT initiatives are strongly involved with IT service providers or consulting firms, relying on them to help across solution delivery and provide business advice.
- Connect with an IoT Development Company that thinks engineering beyond design and work on integrating all the components of IoT in a manner that is focused on connectivity, gaining insight, and maintaining accuracy at all the stages.
- Nevertheless, keep a scope for enhancements of product capabilities to successfully improve the functional efficiency of the product or service based on the latest technology.

## 6. Data Capturing Capabilities

- The purpose of capturing data is to transform the information collected from various sources in a standard format that can be analyzed and automated.
- As IoT is mainly about dependence on sensors for signals and networks for the distribution, chances are that due to certain anomalies in runtime, such as a shutdown of power, incorrect data may get recorded.

## 7. Intelligent Analytics

- At this stage, we are at the very purpose of IoT i.e. translating data into meaningful information. A flaw in data or data model could lead to false positives and false negatives. We have to understand the data in itself is not an insight, rather right questions have to be asked from the precise data to gain the insight.
- Legacy systems such as traditional analytics software where not all data can be loaded at a time can limit the capabilities to manage real-time data. Here’s the list of challenges that deter intelligent analytics:
  - Unpredictable action of the machine during an incident
  - Traditional analytics software
  - Slow adoption of the latest technology due to the high cost
  - Lack of skilled professionals in data mining, algorithms, machine learning, and complex event processing

## 8. Data Security and Privacy Issues

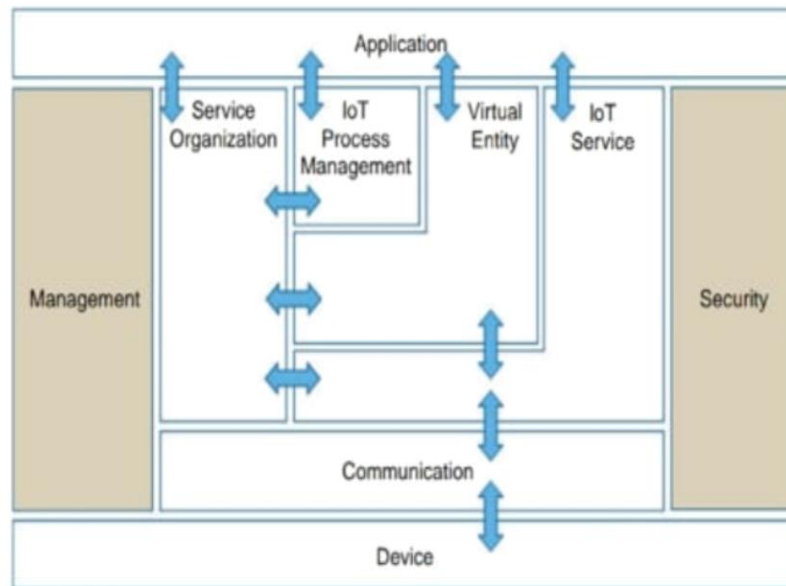
- Even top companies like Apple, known for big security claims, and visionaries like Elon Musk have not been spared by hackers. Recent cases of ransomware attacks have also challenged the confidence of corporate.
- A latest research claims that by 2020, 25% of cyber attacks will target IoT devices.
  - Malware infiltration: 24%
  - Phishing attacks: 24%
  - Social engineering attacks: 18%
  - Device misconfiguration issues: 11%
  - Privilege escalation: 9%
  - Credential theft: 6%
- When it comes to cyber security, lapses could be from both company and consumer side, so it is essential for each party to take necessary measures to improve security.
- A study revealed that 54% IoT device owners do not use any third party security tool and 35% out of these do not even change default password on their devices.
- Here, it should be a collaborative effort between companies and customers to plan and implement collaborative data security policies for successful IoT implementation.

#### 9. Consumer Awareness

- Many people are not aware of IoT, but they understand the dependence on Smart Apps like news apps, stocks applications, entertainment applications. It is not actually important for the consumers to how things work technically, but lack of basic awareness can create a fear of security and cost, which could lead to the slow adoption of technology.
- According to a survey of 3,000 U.S. and Canadian consumers conducted by Cisco, 53% consumers would not prefer to get their data collected, irrespective of the device. This shows the fear among users to share their data, which can act as a deterrent to the IoT

#### **Q. Explain Functional model and need for various functional groups with respect to IOT?**

The IoT Functional Model aims at describing mainly the Functional Groups (FG) and their interaction with the ARM, while the Functional View of a Reference Architecture describes the functional components of an FG, interfaces, and interactions between the components. The Functional View is typically derived from the Functional Model in conjunction with high level requirements.



**FIGURE 7.11**

IoT-A Functional Model.

(Carrez et al. 2013)

#### 1. Device functional group

- The Device FG contains all the possible functionality hosted by the physical Devices that are used for instrumenting the Physical Entities. This Device functionality includes sensing, actuation, processing, storage, and identification components, the sophistication of which depends on the Device capabilities.

#### 2. Communication functional group

- The Communication FG abstracts all the possible communication mechanisms used by the relevant Devices in an actual system in order to transfer information to the digital world components or other Devices.
- Examples of such functions include wired bus or wireless mesh technologies through which sensor Devices are connected to Internet Gateway Devices. Communication technologies used between Applications and other functions such as functions from the IoT Service FG are out of scope because they are the typical Internet technologies.

#### 3. IoT Service functional group

- The IoT Service FG corresponds mainly to the Service class from the IoT Domain Model, and contains single IoT Services exposed by Resources hosted on Devices or in the Network (e.g. processing or storage Resources). Support functions such as directory services, which allow discovery of Services and resolution to Resources, are also part of this FG.

#### 4. Virtual Entity functional group



- The Virtual Entity FG corresponds to the Virtual Entity class in the IoT Domain Model, and contains the necessary functionality to manage associations between Virtual Entities with themselves as well as associations between Virtual Entities and related IoT Services, i.e. the Association objects for the IoT Information Model.
- Associations between Virtual Entities can be static or dynamic depending on the mobility of the Physical Entities related to the corresponding Virtual Entities. An example of a static association between Virtual Entities is the hierarchical inclusion relationship of a building, floor, room/corridor/open space, i.e. a building contains multiple floors that contain rooms, corridors, and open spaces. An example of a dynamic association between Virtual Entities is a car moving from one block of a city to another (the car is one Virtual Entity while the city block is another).
- A major difference between IoT Services and Virtual Entity Services is the semantics of the requests and responses to/from these services. Referring back to the parking lot example, the Parking Sensor Service provides as a response only a number “0” or “1” given the identifier of a Loop Sensor (e.g. #11). The Virtual Entity Parking Spot #01 responds to a request about its occupancy status as “free.” The IoT Service provides data or information associated to specific Devices or Resources, including limited semantic information (e.g. Parking sensor #11, value5“0”, units 5 none); the Virtual IoT Service provides information with richer semantics (“Parking spot #01 is free”), and is closer to being human-readable and understandable.

#### 5. IoT Service Organization functional group

- The purpose of the IoT Service Organization FG is to host all functional components that support the composition and orchestration of IoT and Virtual Entity services. Moreover, this FG acts as a service hub between several other functional groups such as the IoT Process Management FG when, for example, service requests from Applications or the IoT Process Management are directed to the Resources implementing the necessary Services.
- Therefore, the Service Organization FG supports the association of Virtual Entities with the related IoT Services, and contains functions for discovery, composition, and choreography of services. Simple IoT or Virtual Entity Services can be composed to create more complex services, e.g. a control loop with one Sensor Service and one Actuator service with the objective to control the temperature in a building. Choreography is the brokerage of Services so that Services can subscribe to other services in a system.

#### 6. IoT Process Management functional group

- The IoT Process Management FG is a collection of functionalities that allows smooth integration of IoT-related services (IoT Services, Virtual Entity Services, Composed Services) with the Enterprise (Business) Processes.

#### 7. Management functional group

- The Management FG includes the necessary functions for enabling fault and performance monitoring of the system, configuration for enabling the system to be flexible to changing User demands, and accounting for enabling subsequent billing for the usage of the system. Support functions such as management of ownership, administrative domain, rules and rights of functional components, and information stores are also included in the Management FG.

#### 8. Security functional group

- The Security FG contains the functional components that ensure the secure operation of the system as well as the management of privacy. The Security FG contains components for Authentication of Users (Applications, Humans), Authorization of access to Services by Users, secure communication (ensuring integrity and confidentiality of messages) between entities of the system such as Devices, Services, Applications, and last but not least, assurance of privacy of sensitive information relating to Human Users. These include privacy mechanisms such as anonymization of collected data, anonymization of resource and Service accesses (Services cannot deduce which Human User accessed the data), and un-linkability (an outside observer cannot deduce the Human User of a service by observing multiple service requests by the same User).

#### 9. Application functional group

- The Application FG is just a placeholder that represents all the needed Logic for creating an IoT application. The applications typically contain custom logic tailored to a specific domain such as a Smart Grid. An application can also be a part of a bigger ICT system that employs IoT services such as a supply chain system that uses RFID readers to track the movement of goods within a factory in order to update the Enterprise Resource Planning (ERP) system.

#### **Q. Compare IOT and M2M with respect to following points.**

- **Application and Services**
- **Business**
- **Technology**

Aspect	M2M	IoT
Applications and services	Point problem driven	Innovation driven
	Single application - single device	Multiple applications - multiple devices
	Communication and device centric	Information and service centric
	Asset management driven	Data and information driven
Business	Closed business operations	Open market place
	Business objective driven	Participatory community driven
	B2B	B2B, B2C
	Established value chains	Emerging ecosystems
	Consultancy and Systems Integration enabled	Open Web and as-a-Service enabled
	In-house deployment	Cloud deployment
Technology	Vertical system solution approach	Horizontal enabler approach
	Specialized device solutions	Generic commodity devices
	De facto and proprietary	Standards and open source
	Specific closed data formats and service descriptions	Open APIs and data specifications
	Closed specialized software development	Open software development
	SOA enterprise integration	Open APIs and web development

**Q. Discuss various inputs and outputs of an M2M Value Chain.**



**Inputs:** Inputs are the base raw ingredients that are turned into a product.

Examples could be cocoa beans for the manufacture of chocolate or data from an M2M device that will be turned into a piece of information.

**Production/Manufacture:** Production/Manufacture refers to the process that the raw inputs are put through to become part of a value chain.

For example, cocoa beans may be dried and separated before being transported to overseas markets. Data from an M2M solution, meanwhile, needs to be verified and tagged for provenance.

**Processing:** Processing refers to the process whereby a product is prepared for sale. For example, cocoa beans may now be made into cocoa powder, ready for use in chocolate bars. For an M2M solution, this refers to the aggregation of multiple data sources to create an information component something that is ready to be combined with other data sets to make it useful for corporate decision-making.

**Packaging:** Packaging refers to the process whereby a product can be branded as would be recognizable to end-user consumers. For example, a chocolate bar would now be ready to eat and have a red wrapper with

the words “KitKatt” on it. For M2M solutions, the data will have to be combined with other information from internal corporate databases, for example, to see whether the data received requires any action. This data would be recognizable to the end-users that need to use the information, either in the form of visualizations or an Excel spreadsheet.

**Distribution/Marketing:** This process refers to the channels to market for products. For example, a chocolate bar may be sold at a supermarket, a kiosk, or even online. An M2M solution, however, will have produced an Information Product that can be used to create new knowledge within a corporate environment examples include more detailed scheduling of maintenance based on real-world information or improved product design due to feedback from the M2M solution.

As mentioned previously, M2M value chains are internal to one company and cover one solution. IoT Value Chains, meanwhile, are about the use and reuse of data across value chains and across solutions.

**What do you mean by M2M communication? Provide any 2 applications for the same.**

#### M2M communication

M2M refers to those solutions that allow communication between devices of the same type and a specific application, all via wired or wireless communication networks.

M2M solutions allow end-users to capture data about events from assets, such as temperature or inventory levels.

Typically, M2M is deployed to achieve productivity gains, reduce costs, and increase safety or security

#### Applications

*Telematics for cars and vehicles.*

Typical applications include navigation, remote vehicle diagnostics, pay-as-you-drive insurance schemes, road charging, and stolen vehicle recovery

*Metering applications*

include primarily remote meter management and data collection for energy consumption in the electricity utility sector, but also for gas and water consumption.

*Remote monitoring*

is more generalized monitoring of assets, and includes remote patient monitoring as one prime example.

*Fleet management*

includes a number of different applications, like data logging, goods and vehicle positioning, and security of valuable or hazardous goods.

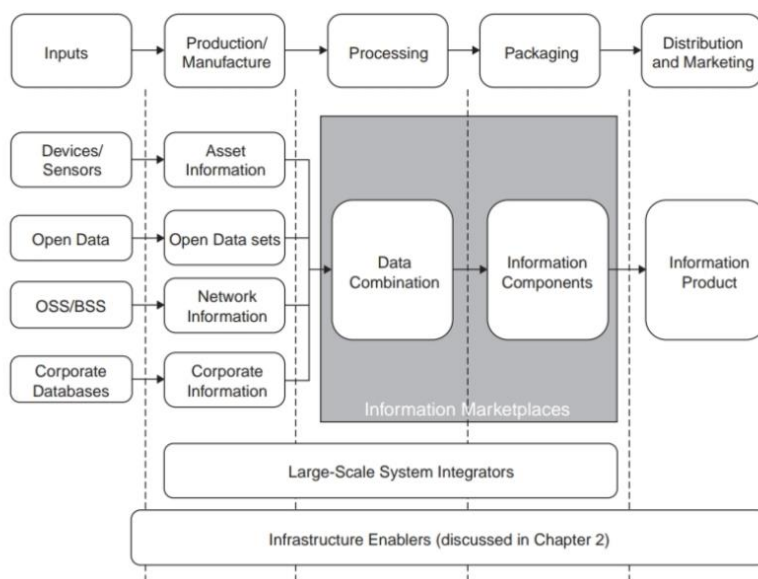
### *Security applications*

are mainly those related to home alarms and small business surveillance solutions

### *Automated Teller Machines (ATM) and Point of Sales (POS) terminals*

ATM and POS devices are connected to a centralized secure environment.

**Draw block diagram for Information-Driven Value Chain for IoT and explain the same.**



**FIGURE 3.3**

An Information-Driven Value Chain for IoT.

- ❖ **Inputs:** The first thing that is apparent for an IoT value chain is that there are significantly more inputs than for an M2M solution
  - *Devices/Sensors:* the manner in which the data from these devices and sensors is used provides a different and much broader marketplace than M2M does
  - *Open Data:* Open data is an increasingly important input to Information Value Chains. A broad definition of open data defines it as: “A piece of data is open if anyone is free to use, reuse, and redistribute it subject only, at most, to the requirement to attribute and/or share-alike” Open data requires a license stating that it is open data

- *OSS/BSS*: The Operational Support Systems and Business Support Systems of mobile operator networks are being used increasingly in tightly closed information marketplaces that allow operators to deliver services to enterprises
- *Corporate Databases*: Companies of a certain size generally have multiple corporate databases covering various functions, including supply chain management, payroll, accounting, etc.
- ❖ Production/Manufacture: In the production and manufacturing processes for data in an IoT solution, the raw inputs described above will undergo initial development into information components and products. Irrespective of input type described above, this process will need to include tagging and linking of relevant data items in order to provide provenance and traceability across the information value chain. Some examples are as follows
  - *Asset Information*: this relates to whatever the sensor/device has been developed to monitor.
  - *Open Data Sets*: Open data sets may include maps, rail timetables, or demographics about a certain area in a country or city.
  - *Network Information*: Network information relates to information such as GPS data, services accessed via the mobile network, etc.
  - *Corporate Information*: Corporate information may be, for example, the current state of demand for a particular product in the supply chain at a particular moment in time.
- ❖ Processing: the data from the various inputs from the production and manufacture stage are combined together to create information. This process involves the extensive use of data analytics for M2M and IoT solutions
- ❖ Packaging: packaging section of the information value chain creates information components. These components could be produced as charts or other traditional methods of communicating information to end-users
- ❖ Distribution/Marketing: The final stage of the Information Value Chain is the creation of an Information Product. A broad variety of such products may exist, but they fall into two main categories:
  - Information products for improving internal decision-making*
  - Information products for resale to other economic actors*

### **In what capacity modern Retailers can leverage IoT for Supply Chain Management?**

to leverage the following features of IoT in supply chain management :

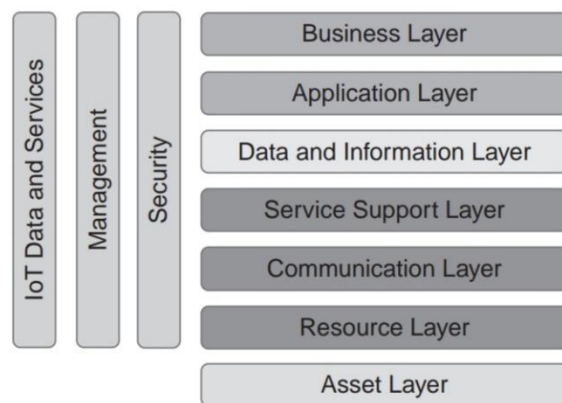
Product tracking: Real-time monitoring and accurate tracking of the supply throughout the product life cycle using location based services (Geofencing, telematics) is bound to help you ensure timely and quality delivery of services and improve customer satisfaction.

Improve Transactional Efficiency: Most retailers struggle with consumers complaining about delayed product delivery, damaged or misplaced order.

These instances could be prevented using the power of cognitive computing to analyze data and offer deep insights.

Efficient Inventory management: With the help of IoT, the supply chain stake holders can build an efficient inventory system. In such a system, the information related to finished products, volume and time of orders, raw material availability, inventory and manufacturing capacity – all are captured using intelligent IoT devices.

**Explain an IoT architecture showing Functional layers and capabilities of an IoT solution with a block diagram**



**FIGURE 4.3**

Functional layers and capabilities of an IoT solution.

- **Asset Layer:** Assets are instrumented with embedded technologies that bridge the digital realm with the physical world, and that provide the capabilities to monitor and control the assets as well as providing identities to the assets.
- **Resource Layer:** provides the main functional capabilities of sensing, actuation, and embedded identities. This is also where gateways of different types are placed that can provide aggregation or other capabilities that are closely related to these basic resources
- **Communication Layer:** provide the means for connectivity between the resources on one end and the different computing infrastructures that host and execute service support logic and application logic on the other end. Different types of networks realize the connectivity, and it is customary to differentiate between the notion of a Local Area Network (LAN) and a Wide Area Network (WAN)

- **Service Support Layer:**IoT applications benefit from simplification by relying on support services that perform common and routine tasks. These support services are provided by the Service Support Layer and are typically executing in data centers or server farms inside organizations or in a cloud environment. These support services can provide uniform handling of the underlying devices and networks, thus hiding complexities in the communications and resource layers.
- **Data and Information Layer:** provides a more abstract set of functions as its main purposes are to capture knowledge and provide advanced control logic support. Key concepts here include data and information models and knowledge representation in general, and the focus is on the organization of information.
- **Application Layer:**provides the specific IoT applications.There is an open-ended array of different applications, and typical examples include smart metering in the Smart Grid, vehicle tracking, building automation, or participatory sensing (PS).
- **Business Layer:**focuses on supporting the core business or operations of any enterprise, organization, or individual that is interested in IoT applications. This is where any integration of the IoT applications into business processes and enterprise systems takes place

In addition to the functional layers, three functional groups cross the different layers,

- **Management:**deals with management of various parts of the system solution related to its operation, maintenance, administration, and provisioning. This includes management of devices, communications networks, and the general Information Technology (IT) infrastructure as well as configuration and provisioning data, performance of services delivered, etc.
- **Security:** is about protection of the system, its information and services,from external threats or any other harm.Trust and identity management, and authentication and authorization, are key capabilities
- **Data and Services:**Data and Service processing can, from a topological perspective, be done in a very distributed fashion and at different levels of complexity



## **Q) WHAT ARE THE CHALLENGES FOR IOT IMPLEMENTATIONS NOW AND IN THE FUTURE?**

**A.**

### **CHALLENGES FOR IOT IMPLEMENTATIONS NOW:**

#### **1. The “high” investment cost**

Moving from one end of the maturity curve to the other may require a substantial investment. Companies shouldn't try to make the leap from beginning to end in one step. A grand vision may be persuasive, but its cost may prevent management from giving the go-ahead.

To manage risk and mitigate cost, several successive “bite-sized” IoT projects implementations with concrete milestones and reasonable costs are recommended. Start small with pilot technologies and then invest in foundational pieces rolled-out in phases. To control costs further, make use of public infrastructure and software-as-a-service in lieu of more expensive private or on-premise installations.

#### **2. Security**

Posting data to—or transferring data via—the internet seems to be the source of many information technology (IT) department nightmares, and rightfully so. Hacking is an international industry producing frequent announcements of security breaches. Putting data online—particularly data related to critical equipment—may seem dangerous. Many IoT platforms consider security a core element and work to ensure that any potential leaks are stopped before hackers find them.

#### **3. Technology infrastructure**

Often, clients have instruments tied into SCADA that generate the data needed to provide analytics and insights. Or, even without power monitoring equipment, SCADA's network potentially could provide the communication infrastructure needed to connect new instrumentation. Yet, almost universally when seeking to tie into SCADA, IT replies, “Our network is super secure and cannot be used to send information to an IoT platform”—and rightfully so.

As discussed under the security of in-flight data, the most secure networks rely on one-way, outbound-only communication. SCADA, being a supervisory control network, necessarily must handle control signals going to the equipment.

#### **4. Communications infrastructure**

Using a cellular gateway to connect IoT instruments sounds great, but users don't get phone reception at some remote sites. Building an infrastructure would be too costly. Although LTE-M and LTE-NB use existing cellular towers, these low-powered, wide-area networks provide much

broader coverage. Even if the user doesn't get a strong-enough signal for voice calls or 4G-LTE data, he or she may still be able to access LTE-M.

## **5. Immaturity of IoT standards**

Understandably, nobody wants to invest in IoT's version of Betamax. Analysts equated protocols emerging from the early IoT industry as a "cacophony of discordant musicians." Waiting to see which standard or protocol would win results in delayed IoT investments. While some IoT standards are still in development, and there's still a lot of fragmentation in the market, standards affecting currently available devices were mostly ironed out in 2016 and 2017.

## **6. Procuring IoT**

Implementing IoT often involves procuring devices and services that don't have IoT in their name, such as instrumentation, communication networks, storage, and data management consultants. The complexity of procuring these services and the lack of the IoT label can make it difficult for stakeholders to see how the multitude of pieces fit together.

The right plan can help streamline this complexity and help communicate each piece's importance to the overall project and make it work.

## **CHALLENGES FOR IOT IMPLEMENTATIONS FUTURE:**

### **Security**

IoT has already turned into a serious security concern that has drawn the attention of prominent tech firms and government agencies across the world. The hacking of baby monitors, smart fridges, thermostats, drug infusion pumps, cameras and even assault rifles are signifying a security nightmare being caused by the future of IoT. So many new nodes being added to networks and the internet will provide malicious actors with innumerable attack vectors and possibilities to carry out their evil deeds, especially since a considerable number of them suffer from security holes.

### **Connectivity**

Connecting so many devices will be one of the biggest challenges of the future of IoT, and it will defy the very structure of current communication models and the underlying technologies. At present, we rely on the centralized, server/client paradigm to authenticate, authorize and connect different nodes in a network.

## Compatibility and Longevity

IoT is growing in many different directions, with many different technologies competing to become the standard. This will cause difficulties and require the deployment of extra hardware and software when connecting devices.

Other compatibility issues stem from non-unified cloud services, lack of standardized M2M protocols and diversities in firmware and operating systems among IoT devices.

## Standards

*Technology standards* which include network protocols, communication protocols, and data-aggregation standards, are the sum of all activities of handling, processing and storing the data collected from the sensors. This aggregation increases the value of data by increasing, *the scale, scope, and frequency* of data available for analysis.

## Intelligent Analysis & Actions

The last stage in IoT implementation is extracting insights from data for analysis, where analysis is driven by *cognitive technologies* and the accompanying models that facilitate the use of cognitive technologies.

## Q) WHAT IS WIRELESS HOME AUTOMATION? DRAW THE BLOCK DIAGRAM FOR THE SAME AND EXPLAIN THE ROLE OF IOT IN WIRELESS HOME AUTOMATION.

A.

### WIRELESS HOME AUTOMATION:

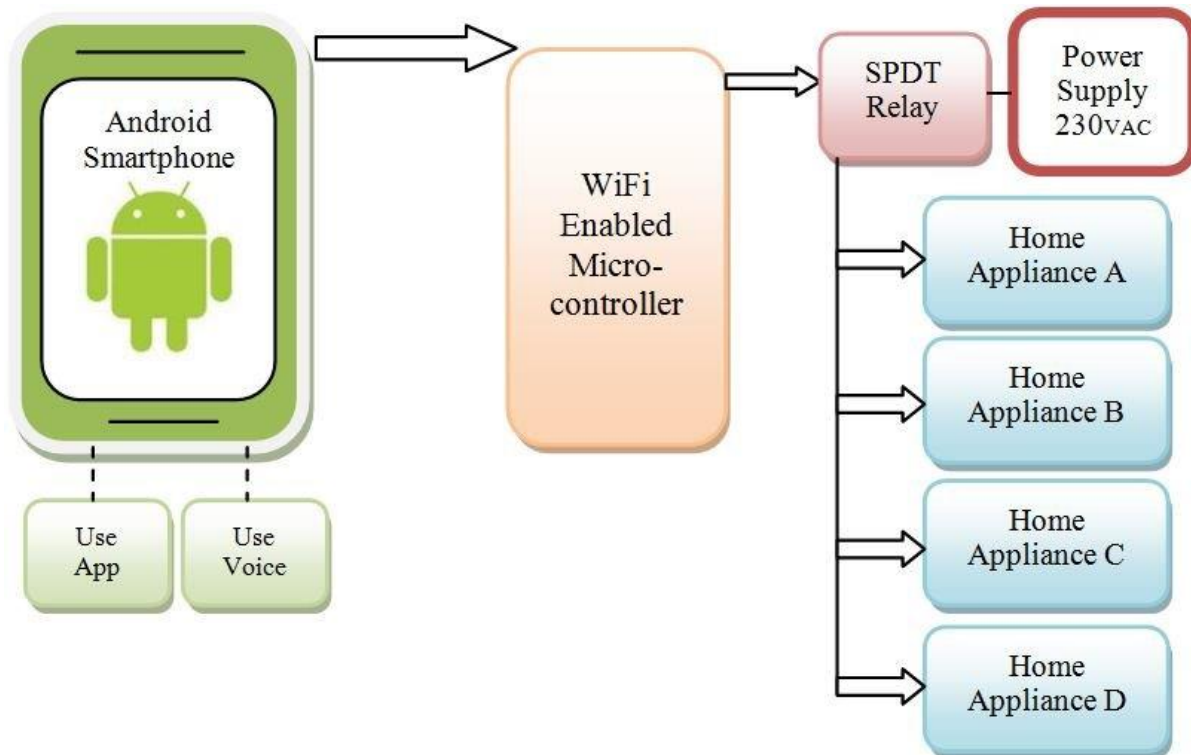
“Home automation” refers to the automatic and electronic control of household features, activity, and appliances. In simple terms, it means you can easily control the utilities and features of your home via the Internet to make life more convenient and secure, and even spend less on household bills.

Home automation systems offer a variety of services and functions. Some of the more common features available through these platforms include:

- Fire and carbon monoxide monitoring
- Remote lighting control
- [Thermostat control](#)
- Appliance control
- Home automation security systems and cameras

- [Live video surveillance](#)
- Alarm systems
- Real-time text and email alerts
- Digital personal assistant integration
- Keyless entry
- Voice-activated control

#### BLOCK DIAGRAM:



#### ROLE OF IOT:

The IoT based Home Automation will enable the user to use a Home Automation System based on Internet of Things (IoT). The modern homes are automated through the internet and the home appliances are controlled. The user commands over the internet will be obtained by the Wi-Fi modems. The Microcontroller has an interface with this modem. The system status is displayed through the LCD display, along with the system data. This is a typical IoT based Home Automation system, for controlling all your home appliances. The smart home market is taking off as IoT device prices come down and the general public comes to understand the benefits of these products. And from smart homes, the next logical step is smart cities, which would take the IoT to the next level. And yet, smart homes are just one small part of our daily lives that the Internet of Things will transform in the coming years.

**Q) EXPLAIN GVC WITH PROPER DIAGRAM.**

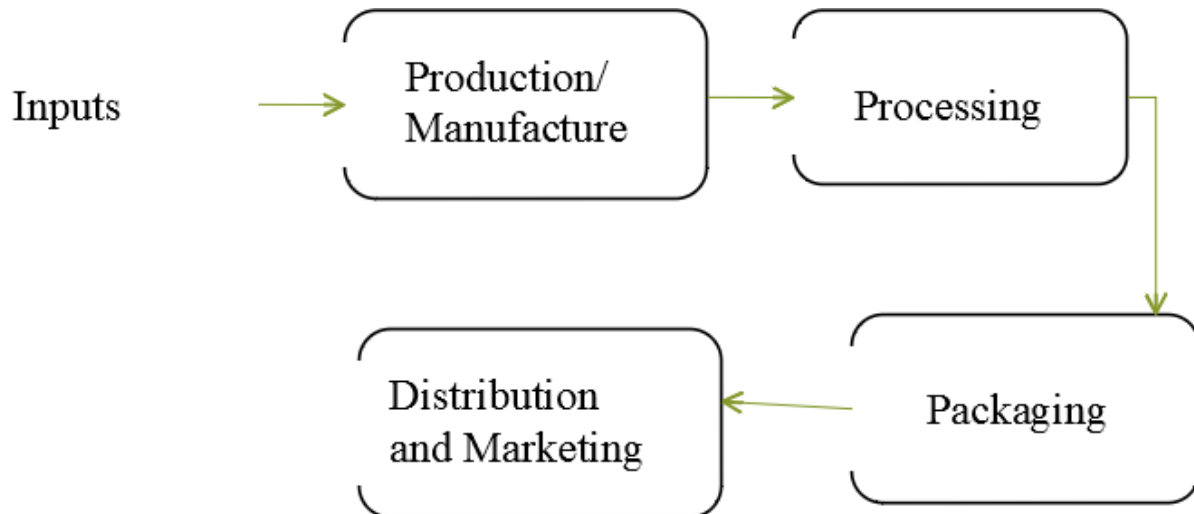
**A.**

***Global value chains***

A value chain describes the full range of activities that firms and workers perform to bring a product from its conception to end use and beyond, including design, production, marketing, distribution, and support to the final consumer

Analyzing an industry from a global value chain (GVC) perspective permits understanding of the context of globalization on the activities contained within them by “focusing on the sequences of tangible and intangible value-adding activities, from conception and production to end use. GVC analysis therefore provides a holistic view of global industries both from the top down and from the bottom up”.

***M2M value chain***



<b>Input</b>	Inputs are the base raw ingredients that are turned into a product.
<b>Example</b>	cocoa beans for the manufacture of chocolate
<b>M2M example</b>	data from an M2M device that will be turned into a piece of information.
<b>Production/ Manufacture:</b>	Production/Manufacture refers to the process that the raw inputs are put through to become part of a value chain.
<b>Example</b>	cocoa beans may be dried and separated before being transported to overseas markets.
<b>M2M example</b>	Data from an M2M, needs to be verified and tagged for provenance.
<b>Processing:</b>	Processing refers to the process whereby a product is prepared for sale.
<b>Example</b>	cocoa beans may now be made into cocoa powder, ready for use in chocolate bars.
<b>M2M example</b>	M2M refers to the aggregation of multiple data sources to create an information component something that is ready to be combined with other data sets to make it useful for corporate decision-making.

<b>Packaging:</b>	Packaging refers to the process whereby a product can be branded as would be recognizable to end-user consumers.
<b>Example</b>	A chocolate bar would now be ready to eat and have a red wrapper with the words “KitKatt” on it.
<b>M2M example</b>	M2M data will have to be combined with other information from internal corporate databases, for example, to see whether the data received requires any action. This data would be recognizable to the end-users that need to use the information, either in the form of visualizations or an Excel spreadsheet.
<b>Distribution/Marketing:</b>	This process refers to the channels to market for products.
<b>Example</b>	a chocolate bar may be sold at a supermarket, a kiosk, or even online.
<b>M2M example</b>	will have produced an Information Product that can be used to create new knowledge within a corporate environment examples include more detailed scheduling of maintenance based on real-world information or improved product design due to feedback from the M2M solution.

#### Q) WHAT IS M2M AND HOW IS IT DIFFERENT FROM COMPARED TO IOT?

A.

##### Machine to machine

---

**Machine to machine (M2M)** is direct communication between devices using any [communications channel](#), including [wired](#) and [wireless](#).<sup>[1][2]</sup> Machine to machine communication can include industrial instrumentation, enabling a sensor or meter to communicate the information it records (such as temperature, inventory level, etc.) to application [software](#) that can use it (for example, adjusting an industrial process based on temperature or placing orders to replenish inventory).<sup>[3]</sup> Such communication was originally accomplished by having a remote network of machines relay information back to a central hub for analysis, which would then be rerouted into a system like a [personal computer](#).<sup>[4]</sup>

More recent machine to machine communication has changed into a system of networks that transmits data to personal appliances. The expansion of [IP](#) networks around the world has made machine to machine communication quicker and easier while using less power.<sup>[5]</sup> These networks also allow new business opportunities for consumers and suppliers.<sup>[6]</sup>

## M2M versus the IoT

M2M	IoT
M2M is about direct communication between machines.	The IoT is about sensors automation and Internet platform.
It supports point-to-point communication.	It supports cloud communication.
Devices do not necessarily rely on an Internet connection.	Devices rely on an Internet connection.
M2M is mostly hardware-based technology.	The IoT is both hardware- and software-based technology.
Machines normally communicate with a single machine at a time.	Many users can access at one time over the Internet.
A device can be connected through mobile or other network.	Data delivery depends on the Internet protocol (IP) network.

### Q What are smart and connected products ? explain in brief?

#### Connected Devices

1. Strictly speaking, connected products have been around for some time—as long as technology such as the Internet has been available to link them up
2. Devices such as laptops and printers were among the very first to be connected.
3. However, in modern lingo the term “connected device” primarily refers to a product with a real-world function that’s connected to the Internet in order to transmit data or be controlled remotely. The term “Internet of Things” (IoT) is used to describe the massive network of connected devices and systems that collect and exchange data.
4. sensors on manufacturing equipment that detect temperature anomalies to toasters that print the weather forecast on your bread, there’s no shortage of connected devices on the market for consumer and enterprise use. Gartner estimates that by 2020, the IoT will consist of 26 billion connected devices—a figure which is even more impressive because it doesn’t include computers, tablets, or smartphones.

#### Smart Devices



1. There's some confusion and debate about what exactly constitutes a "smart" device. Of course, it doesn't help that manufacturers often label their products "smart" for marketing purposes when there's nothing particularly intelligent about them.
2. Although connected devices may take some simple actions based on what they detect in the environment, their primary purpose is to send and receive data. Smart devices, on the other hand, have some kind of intelligent behavior that enables them to react to real-world situations and even predict the needs of their users.
3. Smart devices often have a much more complex hardware architecture, including things such as sensors, microprocessors, data storage, controls, and embedded operating systems.
4. Most smart devices are connected to the Internet (although they don't have to be), but not all connected devices are smart.

### **Q Explain functional model and its various functional groups?**

Ans: 1. The IoT Functional Model aims at describing mainly the FGs and their interaction with the ARM, while the Functional View of a Reference Architecture describes the Functional Components (FCs) of an FG, interfaces, and interactions between the components.

2. The Functional View is typically derived from the Functional Model in conjunction with high-level requirements.

#### **Device Functional Group**

1. The Device FG contains all the possible functionality hosted by the physical Devices that are used for instrumenting the Physical Entities.

2. This Device functionality includes sensing, actuation, processing, storage, and identification components, the sophistication of which depends on the Device capabilities.

#### **Communication Functional Group**

1. The Communication FG abstracts all the possible communication mechanisms used by the relevant Devices in an actual system in order to transfer information to the digital world components or other Devices.

2. Examples of such functions include wired bus or wireless mesh technologies through which sensor Devices are connected to Internet Gateway Devices. Communication technologies used between Applications and other functions such as functions from the IoT Service FG are out of scope because they are the typical Internet technologies.

#### **IoT Service Functional Group**

1.The IoT Service FG corresponds mainly to the Service class from the IoT Domain Model and contains single IoT Services exposed by Resources hosted on Devices or in the Network (e.g., processing or storage Resources).

2.Support functions such as directory services, which allow discovery of Services and resolution of Resources, are also part of this FG.

#### Virtual Entity Functional Group

1.The Virtual Entity FG corresponds to the Virtual Entity class in the IoT Domain Model and contains the necessary functionality to manage associations between Virtual Entities with themselves as well as associations between Virtual Entities and related IoT Services, i.e., the Association objects for the IoT Information Model.

2. Associations between Virtual Entities can be static or dynamic depending on the mobility of the Physical Entities related to the corresponding Virtual Entities.

#### IoT Service Organization Functional Group

The purpose of the IoT Service Organization FG is to host all FCs that support the composition and orchestration of IoT and Virtual Entity services. Moreover, this FG acts as a service hub between several other FGs such as the IoT Process Management FG when, for example, service requests from Applications or the IoT Process Management are directed to the Resources implementing the necessary Services.

#### IoT Process Management Functional Group

The IoT Process Management FG is a collection of functionalities that allows smooth integration of IoT-related services (IoT Services, Virtual Entity Services, Composed Services) with the Enterprise (Business) Processes.

#### Management Functional Group

1.The Management FG includes the necessary functions for enabling fault and Performance Monitoring of the system, configuration for enabling the system to be flexible to changing User demands and accounting for enabling subsequent billing for the usage of the system.

2.Support functions such as management of ownership, administrative domain, rules and rights of FCs, and information stores are also included in the Management FG.

#### Security Functional Group

1 .The Security FG contains the FCs that ensure the secure operation of the system as well as the management of privacy.

2 The Security FG contains components for Authentication of Users (Applications, Humans), Authorization of access to Services by Users, se

cure communication (ensuring integrity and confidentiality of messages) between entities of the system such as Devices, Services, and Applications, and last but not least, assurance of privacy of sensitive information relating to Human Users.

#### Application Functional Group

The Application FG is just a placeholder that represents all the needed logic for creating an IoT application. The applications typically contain custom logic tailored to a specific domain such as a Smart Grid.

#### Q Role of IoT in healthcare with suitable example?

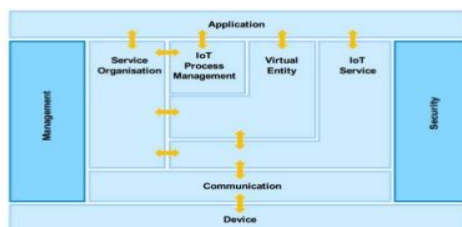
Ans:1. One thing is certain; IoT increases productivity.

2. Apart from lowering the risks, the device data can also be used for preventing machine failure, which is quite an advantage for life sciences, as this can help to improve reliability and quality when it comes to the patient's supply chain responsiveness.

3. Meaning, lower expenses in terms of production.

4. On that note, the beauty of IoT in healthcare and life science is personalized medicine. The IoT enables an integrated production process from sales order to workstation. Putting it simply, the industry can make a bunch of products suitable for personal therapeutic use because, during the product assembling stage, individual directions are passed onto a special smart software.

#### Functional Model



#### EXAMPLES

Organs-on-Chips

Yes, you have read this correctly and it's exactly what your first thought implied. Organs-on-Chips is truly a groundbreaking invention of creating human organ system on miniature micro-engineered chips. These chips help to better understand the functions of our organs, in addition to supporting research and development by reducing cost, data reporting, and increasing medicine efficiency. It is an advanced personalized treatment with the main role of accelerating new drug developments.

## **Wearables**

What is the deal with wearables one might ask? Well, these devices in their current form are only the beginning of something spectacular. So far, we have had wristbands that measure a user's pulse, daily routine, blood pressure, temperature, and oxygen levels. All these pieces of data are sent to the user's smartphone. We can only imagine what comes next because this type of health support can eliminate human error in versatile healthcare regions.

## **Q.State the role of following with respect to IGVC :-**

### **Data Factories**

Data factories are those entities that produce data in digital forms for use in other parts of the I-GVC. Many of these companies existed in the pre digital era; for example, Ordnance Survey (OS) in the UK has always collected map information from the field, and collated and produced maps for purchase. Previously, such data factories would create paper-based products and sell them to end-users via retailers. With the move to the digital era, however, these companies now also provide this data via digital means; for example, OS now makes maps and associated data available in digital format. Essentially, its business model has not changed significantly it still produces maps but its means of delivery of products has changed. Moreover, its products can now be combined, reused, and bundled together with other products by actors in the commodity chain as the foundation of other services. For example, maps from OS can be combined with other data from travel services such as TFL to provide detailed travel applications on mobile devices.

### **Service Provider/Data wholesalers**

Service Providers and Data wholesalers are those entities that collect data from various sources worldwide, and through the creation of massive databases, use it to either improve their own information products or sell information products in various forms. Many examples exist; several well-known ones are Twitter, Facebook, Google, etc.... Google "sells" its data assets through the development of extremely accurate, targeted, search-based advertising mechanisms that it is able to sell to companies wishing to reach a particular market. Twitter, meanwhile, through collating streams of "Tweets" from people worldwide, is able to collate customer sentiment about different products and world events, from service at a restaurant to election processes across the globe; through what Twitter refers to as a "data hose," companies and developers can access 50% of end-user Tweets for \$360,000 USD per annum.

### **Intermediaries**

In the emerging industrial structure of the I-GVC, there is a need for intermediaries that handle several aspects of the production of information products. As mentioned above, there are many privacy and regional issues associated with the collection of personal information. In Europe, the manner in which Facebook collects and uses the data of the individuals that participate in its service may actually be in contravention of European privacy law. The development of databases such as the ones created by Google, Facebook, and Twitter may therefore require the creation of entities that are able to “anonymise” data sufficiently to protect individuals’ privacy rights in relevant regional settings. These corporations will provide protection for the consumer that their data is being used in an appropriate manner, i.e. the manner in which the consumer has approved its usage.

## **Resellers**

Resellers are those entities that combine inputs from several different intermediaries, combine it together, analyze, and sell it to either end-users or to corporate entities. These resellers are currently rather limited in terms of the data that they are able to easily access via the converged communications platform, but they are indicative of the types of corporate entities that are forming within this space. One example is BlueKai, which tracks the online shopping behavior of Internet users and mines the data gathered for “purchasing intent” in order to allow advertisers to target buyers more accurately. BlueKai combines data from several sources, including Amazon, Ebay, and Alibaba

## **Q. Define the term IOT with suitable example**

The Internet of Things, or IoT, is essentially an ecosystem of physical devices, vehicles, appliances, and other things that have the ability to connect, collect and exchange data over a wired and wireless network, with little or no human-to-human or human-to-computer intervention. Allowing integration and data exchange between physical devices and the computer, this new wave of technology focuses on making human life more simplified and comfortable with the right mix of efficiency and productivity.

To be more specific, taking advantage of cutting-edge technologies like Machine Learning, Machine-to-Machine (M2M) Communication and Artificial Intelligence (AI), IoT aims at extending connectivity beyond standard Internet supported physical devices (smartphones, tablets, desktops, and laptops) to a wide spectrum of non-internet-enabled physical devices and everyday objects, such as coffee makers, washing machines, door locks, etc., so you can remotely monitor and control them with the help of a mobile or tablet device

### **IOT Example :**

#### **1) Nest Learning Thermostat**

Nest Learning Thermostat self-learning Wi-Fi-enabled smart thermostat that leverages Machine Learning to automatically optimize the heating and cooling of your home to conserve energy. You can also manually control your home’s temperature with just a few taps on your smartphone or tablet

#### **Amazon Go**

Amazon Go is one of its kind retail store that facilitates customers shopping with no checkout required. All you need to sign in the Amazon Go app to enter the store, then shop as you normally would and leave the store. No lines, no checkout, just walk out!

## Q.State and explain various components of IOT

**1. Smart devices & sensors** - Device connectivity - Sensors like Temperature sensors, Wifi Humidity sensors, Wifi Light sensor, Wifi Vibration sensor, Proximity detection, RFID tags etc. Basically these sensors are continuously collecting data from the environment and transmit the information to the next layer.

**2. Gateway** - IOT gateway manages the bidirectional data traffic between different networks and protocols. IOT gateway offers extreme level of security for the network and transmitted data with higher order encryption techniques.

**3. Cloud** - Basically cloud is an advanced high performance network of servers optimized to perform high speed data. Cloud system integrates billion of devices, sensors, gateway, protocols, data storage and provide predictive analytics.

**4. Analytics** - Analytics is the process of converting analog data from billions of smart devices and sensors into useful insights which can be interpreted and used for detailed analysis.

**5. User interface** - User interfaces are the visible, tangible part of the IoT system which can be accessible by users.

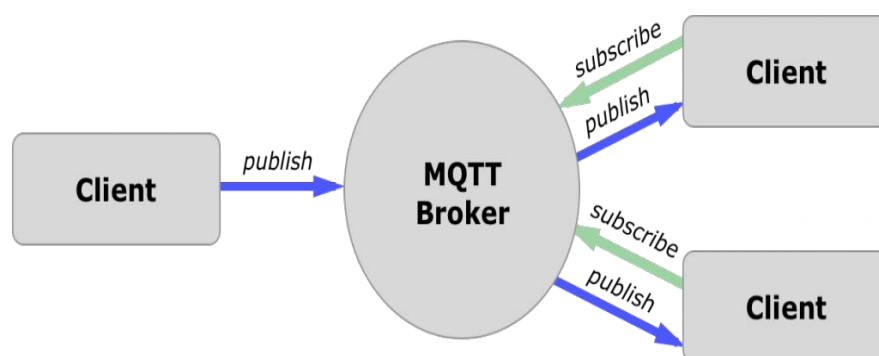
## Q.Explain in brief MQTT broker

MQTT is one of the most commonly used protocols in IoT projects. It stands for Message Queuing Telemetry Transport.

In addition, it is designed as a lightweight messaging protocol that uses publish/subscribe operations to exchange data between clients and the server. Furthermore, its small size, low power usage, minimized data packets and ease of implementation make the protocol ideal of the “machine-to-machine” or “Internet of Things” world.

### How MQTT works

Like any other internet protocol, MQTT is based on clients and a server. Likewise, the server is the guy who is responsible for handling the client's requests of receiving or sending data between each other.



MQTT server is called a broker and the clients are simply the connected devices.

So: When a device (a client) wants to send data to the broker, we call this operation a “publish”.

When a device (a client) wants to receive data from the broker, we call this operation a “subscribe”.

### MQTT Components:

That takes us to the MQTT components, which are 5 as follows:

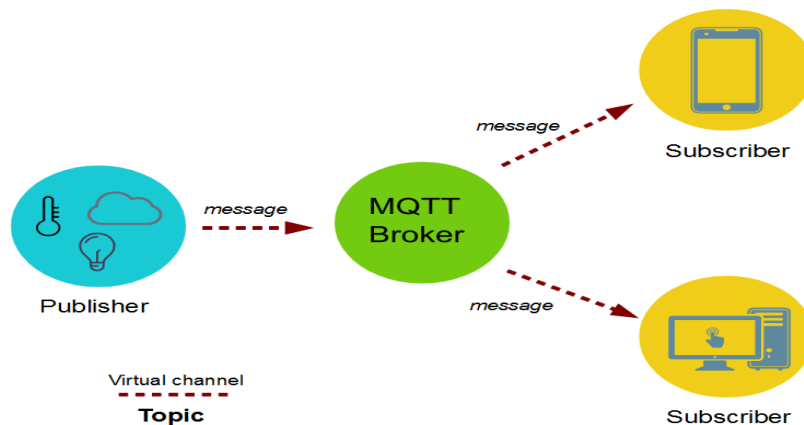
Broker, which is the server that handles the data transmission between the clients.

A topic, which is the place a device want to put or retrieve a message to/from.

The message, which is the data that a device receives “when subscribing” from a topic or send “when publishing” to a topic.

Publish, is the process a device does to send its message to the broker.

Subscribe, where a device does to retrieve a message from the broker.



### Q Explain various aspects of trust model in arm?

Ans:1.Building trusted IoT environments is of great importance to achieve the full benefits of smart applications.

2.Trust models are proposed to tackle behaviour-related issues.

- **Trust Model Domains:** Because ICT and IoT systems may include a large number of interacting entities with different properties, maintaining trust relationships for every pair of interacting entities may be prohibitive. Therefore, groups of entities with similar trust properties can define different trust domains.
- **Trust Evaluation Mechanisms:** These are well-defined mechanisms that describe how a trust score could be computed for a specific entity. The evaluation mechanism needs to take into account the source of information used for computing the trust level/score of an entity; two related aspects are the federated trust and trust anchor. A related concept is the IoT support for evaluation of the trust level of a Device, Resource, and Service.
- **Trust Behavior Policies:** These are policies that govern the behavior between interacting entities based on the trust level of these interacting entities; for example, how a User could use sensor measurements retrieved by a Sensor Service with a low trust level.
- **Trust Anchor:** This is an entity trusted by default by all other entities belonging to the same trust model, and is typically used for the evaluation of the trust level of a third entity.
- **Federation of Trust:** A federation between two or more Trust Models includes a set of rules that specify the handling of trust relationships between entities with different Trust Models. Federation becomes important in large-scale systems.

### Q. 1 State applications of IOT in real life.

Ans:

#### 1. Smart Home

- Smart Home has become the revolutionary ladder of success in the residential spaces and it is predicted Smart homes will become as common as smartphones.
- The cost of owning a house is the biggest expense in a homeowner's life. Smart Home products are promised to save time, energy and money.
- With Smart home companies like Nest, Ecobee, Ring and August, to name a few, will become household brands and are planning to deliver a never seen before experience.

#### 2. Smart Wearables

- Wearable devices are installed with sensors and softwares which collect data and information about the users. This data is later pre-processed to extract essential insights about user.
- These devices broadly cover fitness, health and entertainment requirements.



### 3. Connected Cars

- **A connected car is a vehicle which is able to optimize it's own operation, maintenance as well as comfort of passengers using onboard sensors and internet connectivity.**

### 4. Smart Cities

- **Smart city is another powerful application of IoT generating curiosity among world's population.**
- **Smart surveillance, automated transportation, smarter energy management systems, water distribution, urban security and environmental monitoring all are examples of internet of things applications for smart cities.**

### 5. IoT in agriculture

- **Smart farming is one of the fastest growing field in IoT.**
- **Farmers are using meaningful insights from the data to yield better return on investment.**
- **Sensing for soil moisture and nutrients, controlling water usage for plant growth and determining custom fertilizer are some simple uses of IoT.**

### 6. IOT in Healthcare

- **IoT in healthcare is aimed at empowering people to live healthier life by wearing connected devices.**
- **The collected data will help in personalized analysis of an individual's health and provide tailor made strategies to combat illness.**

### 7. Smart Retail

- **IoT provides an opportunity to retailers to connect with the customers to enhance the in-store experience.**
- **Smartphones will be the way for retailers to remain connected with their consumers even out of store.**

### 8. Industrial Internet

- **Industrial Internet is the new buzz in the industrial sector, also termed as Industrial Internet of Things ( IIoT ).**
- **It is empowering industrial engineering with sensors, software and big data analytics to create brilliant machines.**

**Q Explain CoAP protocol used in IOT.**

**Ans:**

- **Constrained Application Protocol (CoAP) is a specialized web transfer protocol for use with constrained nodes and constrained networks in the Internet of Things.**
- **CoAP is designed to enable simple, constrained devices to join the IoT even through constrained networks with low bandwidth and low availability.**
- **It is generally used for machine-to-machine (M2M) applications such as smart energy and building automation. The protocol was designed by the Internet Engineering Task Force ([IETF](#)).**
- **CoAP functions as a sort of [HTTP](#) for restricted devices, enabling equipment such as sensors or actuators to communicate on the IoT.**

- These sensors and actuators are controlled and contribute by passing along their data as part of a system.
- The protocol is designed for reliability in low bandwidth and high congestion through its low power consumption and low network overhead.
- CoAP relies on UDP security features to protect information.

**Q Discuss various characteristics/features due to which IOT is popular in market.**

**Ans:**

Internet of Things (IoT) is a technology of connected smart devices that has incremental use cases across industries.

### **1. Connectivity**

- In the case of IoT, the most important feature one can consider is connectivity.
- Without seamless communication among the interrelated components of the IoT ecosystems (i.e sensors, compute engines, data hubs, etc.) it is not possible to execute any proper business use case.
- IoT devices can be connected over Radio waves, Bluetooth, Wi-Fi, Li-Fi, etc.

### **2. Sensing**

- In the case of IoT in order to get the best of it, we need to read the analog signal, convert it in such a way that we can derive meaningful insights out of it.
- We use Electrochemical, gyroscope, pressure, light sensors, GPS, Electrochemical, pressure, RFID, etc. to gather data based on a particular problem.
- For example for automotive use cases, we use Light detection sensors along with pressure, velocity and imagery sensors.

### **3. Active Engagements**

- IoT device connects various products, cross-platform technologies and services work together by establishing an active engagement between them.
- In general, [we use cloud computing](#) in blockchain to establish active engagements among IoT components.

### **4. Scale**

- IoT devices should be designed in such a way that they can be scaled up or down easily on demand.
- In general, IoT is being used from smart home automation to automating large factories and work stations, so the use cases vary in scale.

### **5. Dynamic Nature**

- For any IoT use case, the first and foremost step is to collecting and converting data in such a way that means business decisions can be made out of it.
- In this whole process, [various components of IoT](#) need to change their state dynamically. For example, the input of a temperature sensor will vary continuously based on weather conditions, locations, etc.
- IoT devices should be designed this keeping in mind.

### **6. Safety**

- One of the main features of the IoT ecosystem is security.
- In the whole flow of an IoT ecosystem, sensitive information is passed from endpoints to the analytics layer via connectivity components.

- While designing an IoT system we need to adhere to proper safety, security measures, and firewalls to keep the data away from misuse and manipulations.

## 7. Integration

- IoT integrates various cross-domain models to enrich user experience.
- It also ensures proper trade-off between infrastructure and operational costs.

**Q Mention advantages and disadvantages of IOT.**

**Ans:**

### Advantages:

#### 1. Data:

The more the information, the easier it is to make the right decision. Knowing what to get from the grocery while you are out, without having to check on your own, not only saves time but is convenient as well.

#### 2. Tracking:

The computers keep a track both on the quality and the viability of things at home. Knowing the expiration date of products before one consumes them improves safety and quality of life. Also, you will never run out of anything when you need it at the last moment.

#### 3.

#### Time:

The amount of time saved in monitoring and the number of trips done otherwise would be tremendous.

#### 4. Money:

The financial aspect is the best advantage. This technology could replace humans who are in charge of monitoring and maintaining supplies.

### Disadvantages:

#### 1. Compatibility:

As of now, there is no standard for tagging and monitoring with sensors. A uniform concept like the USB or Bluetooth is required which should not be that difficult to do.

#### 2. Complexity:

There are several opportunities for failure with complex systems. For example, both you and your spouse may receive messages that the milk is over and both of you may end up buying the same. That leaves you with double the quantity required. Or there is a software bug causing the printer to order ink multiple times when it requires a single cartridge.

#### 3. Privacy/Security:

Privacy is a big issue with IoT. All the data must be encrypted so that data about your financial status or how much milk you consume isn't common knowledge at the work place or with your friends.

#### 4. Safety:

There is a chance that the software can be hacked and your personal information misused. The possibilities are endless. Your prescription being changed or your account details being hacked could put you at risk. Hence, all the safety risks become the consumer's responsibility.

**Q) Discuss the working of HTTP protocol.**

Ans)

The Hypertext Transfer Protocol is an application protocol for distributed, collaborative, hypermedia information systems that allows users to communicate data on the World Wide Web.

HTTP was invented alongside HTML to create the first interactive, text-based web browser: the original World Wide Web. Today, the protocol remains one of the primary means of using the Internet.

As a request-response protocol, HTTP gives users a way to interact with web resources such as HTML files by transmitting hypertext messages between clients and servers. HTTP clients generally use Transmission Control Protocol (TCP) connections to communicate with servers.

HTTP utilizes specific request methods in order to perform various tasks:

GET requests a specific resource in its entirety

HEAD requests a specific resource without the body content

POST adds content, messages, or data to a new page under an existing web resource

PUT directly modifies an existing web resource or creates a new URI if need be

DELETE gets rid of a specified resource

TRACE shows users any changes or additions made to a web resource

OPTIONS shows users which HTTP methods are available for a specific URL

CONNECT converts the request connection to a transparent TCP/IP tunnel

PATCH partially modifies a web resource

All HTTP servers use the GET and HEAD methods, but not all support the rest of these request methods.

**Q)How Upnp protocol is different than HTTP. Explain in brief.**

Ans)

Universal Plug and Play (UPnP) is a networking protocol or a set of networking protocols which enables devices like personal computers, WiFi, Mobile devices, printers etc. to discover each other and establish connections for sharing services and data and also entertainment purposes. UPnP is intended to be used on residential networks. UPnP can be considered as an extension of Plug and Play which enables users to connect devices directly to a computer without any manual configurations to the device or to the computer.

UPnP allows direct networking between home appliances like printers, personal computers, mobile devices, and many more. It uses established standard industry protocols like TCP/IP, XML, Simple Object Access Protocol (SOAP), UDP, DHCP (Dynamic Host Configuration Protocol), DNS (Domain Name System). UPnP technology was initially promoted by UPnP forum which was an initiative by various vendors.

Advantages of UPnP:

- 1.It can be used for NAT traversal or Firewall punching.
- 2.It allows real Plug and Play compatibility.

3.It is backed by various big vendors and companies like Microsoft and Intel, which makes it an industry standard.

4.It is an ideal architecture for home devices and networks.

**Q) Explain in breif how IoT can be helpful in Education and Government sector?**

Ans)

Education sector:-

a. Poster boards into IoT enabled boards

It is indeed very difficult to compare the older era presentation boards with present-day multimedia poster boards. Internet gear like Glogster has changed this ease and permits us to create digital posters without problems combining with the photos, audio, video, text, and hyperlinks.

b. Interactive gaining of knowledge

Getting to know these days is not restrained to the mixture of texts and pictures but beyond that. Most of the textbooks are paired with net-primarily based websites that consist of extra substances, films, exams, animations and different substances to support the mastering.

c. Learning at any time and anywhere

IoT plays an important position in constructing a network through the use of special internet-based systems. Advanced technology enables the academics to display the development of the scholars. IoT allows students and teachers to communicate via extraordinary method, checking messages and upcoming events at the same time when away from the classroom or even replying to posts. It is by far a very effective app that provides safe network and complete privacy. It also allows a user to save your specific thoughts and class undertaking without worrying and assure you full confidentiality.

d. Superior safety features

This Application of IoT in Education is important as enforcing the superior technology answers inside the school rooms and training area may be very useful. It includes emergency indicators, audio enhancement, wi-fi clocks and hearing impaired notifications that offer the scholars and body of workers with a feeling of security.

e. Bye Bye to Chalkboards

Students in recent times make use of a very powerful platform which includes smart boards. It facilitates the lecturers to provide an explanation for the lectures more without problems with the assist of online displays and films.

Government sector:-

1. Law enforcement

The enormity of the general population and complexity of the inhabited areas like cities and towns make it hard for government agencies to provide surveillance and protection with the relatively small number of personnel they have. Even the presence of a myriad of cameras is pointless if there aren't enough eyes to constantly analyze the footage recorded by them. IoT can help government law enforcement agencies in monitoring public safety through its smart network of sensing and scanning devices. Since it is impossible for law enforcement personnel to be present at all high-risk locations simultaneously, IoT enabled surveillance systems can provide continuous, real-time intelligence. These systems, powered by machine and deep learning algorithms, can not only gather video and audio footage but also analyze it for threats.

They can notify concerned personnel upon discovery of potentially harmful situations, which can be looked into and prevented in time.

IoT-enabled license plate scanning systems can help track down vehicles of criminals and traffic offenders. An intelligent, well connected IoT network can not only enable quick response to threats but can also minimize crime rates through preventive action.

## 2. Infrastructure management

Governments spend vast amounts of tax money on building and maintaining infrastructural facilities for the public. This includes the building of roads and bridges, power grids, water supply lines, gas supply, rail lines, airports, etc. which cost a lot more to maintain than to build. Any major damages or defects any of these systems can cause disruptions in civilian activities, which can have far-reaching ramifications. To ensure the effective functioning of these systems, governments can use IoT enabled maintenance and monitoring. For instance, IoT sensors can constantly analyze railway lines and engines for any defects or really sounds of failure and notify the authorities and the concerned staff for prompt repairs. They can also constantly monitor and regulate power supply through nation-wide power grids, and help in identifying impending break-downs.

In addition to helping governments manage infrastructure, IoT can help governments to make decisions regarding new infrastructure projects, such as laying new roads and railway lines.

## 3. Disaster management

Governments have the obligation of preventing man-made disasters and protecting the public from natural ones. IoT and big data enabled disaster prevention and management systems can help governments preserve human life during times of unforeseen disasters. In areas where forest fires are common, IoT sensors can be set up in forests to detect fires in their early stages to curb their spreading and the consequent devastating effects. IoT can also help in areas that are prone to flooding, by monitoring the water bodies for alerting authorities when the water levels rise at alarming rates.

IoT can also prevent man-made disasters by providing emergency response teams with continuous monitoring capability, which enables them to be prepared for contingencies.

Integrating IoT in government operations will be a long and effortful endeavor. However, the potential benefits far outweigh the initial friction. With greater penetration and propagation of IoT, governments can take a fully supervisory role, and ensure public welfare without obtrusion.

## **Q) Explain the working of protocol which is responsible for publishing and subscribing.**

Ans)

MQTT (MQ Telemetry Transport) is a lightweight messaging protocol that provides resource-constrained network clients with a simple way to distribute telemetry information. The protocol, which uses a publish/subscribe communication pattern, is used for machine-to-machine (M2M) communication and plays an important role in the internet of things (IoT). The MQTT protocol is a good choice for wireless networks that experience varying levels of latency due to occasional bandwidth constraints or unreliable connections.

The MQTT protocol surrounds two subjects: a client and a broker. An MQTT broker is a server, while the clients are the connected devices. When a device -- or client -- wants to send data to a

server -- or broker-- it is called a publish. When the operation is reversed, it is called a subscribe.

If the connection from a subscribing client to a broker is broken, then the broker will buffer messages and push them out to the subscriber when it is back online. If the connection from the publishing client to the broker is disconnected without notice, then the broker can close the connection and send subscribers a cached message with instructions from the publisher. While the TT in MQTT stands for Telemetry Transport, the MQ is in reference to a product called IBM MQ.

## **12.XMPP and clayster**

*Xmpp - the Extensible Messaging and Presence Protocol (XMPP) is widely used as a communication protocol. Based on Extensible Markup Language (XML), XMPP enables fast, near-real-time exchange of data between multiple entities on a network.* In contrast to most direct messaging protocols, XMPP is described in an open standard and uses an open systems approach of development and application, by which anyone may implement an XMPP service and interoperate with other organisations' implementations. Since XMPP is an open set of rules, implementations can be developed using any software licence, and many server, client, and library XMPP implementations are distributed as free and open source software. Numerous freeware and commercial software implementations also exist.

Clayster - Internet **Clayster** Include XMPP is the core into which we normalize data from different sources and make it available in unified fabric. The Include platform abstracts and transforms any data source to coexist in a data normalized infrastructure.

## **13.i>thinger.io –**

**Thinger.io** is a platform that allows connecting things to the Internet. It is Open Source, so you can take the code and build your own cloud if you want. It provides thing API discovery right out of the box, so you can code your things and interact easily from the web.

- **imple but Powerful:** Just a couple code lines to connect a device and start retrieving data or controlling it's functionalities with our web based Console, able to connect and manage thousands of devices in a simple way.
- **Hardware agnostic:** Any device from any manufacturer can be easily integrated with Thinger.io's infrastructure.
- **Open-Source:** most of the platform modules, libraries and APP source code are available in our github repository to be downloaded and modified with MIT license.
- **Customizable:** Fully white-labelable frontend allows customizing Thinger.io Platform with your brand colors, logotype and web domain.

13.ii> **Sense iot - Sense IoT** is a sensor data storage, visualisation and remote management platform offering leading cloud computing technologies to provide you excellent data scalability and easy visualisation. We support any web- connected third party device, sensor, or sensor network through a simple open API.

#### **14. Red programming language-**

First fullstack programming solution: combines in one tool, the ability to write high-level code (GUI apps, scripting and DSL) and fast low-level code (writing device drivers, operating systems, native interfacing, etc). Moreover, Red is also a both-sided technology (client & server).

Cross-platform native code compiler: from any platform the toolchain runs on, you can compile to about 15 other platforms, with a simple command-line option (-t Windows, -t Linux, -t Darwin, -t RPi, ...).

Extremely lightweight: Red is a 1MB, single-file, no install, no setup, toolchain. It takes typically a few seconds to download and you can immediatly start writing and running code, there's *\*nothing\** to setup (it's just terrible that this is the exception instead of being the norm...).

Batteries-included solution: it comes with a very rich runtime library, despite its tiny size, covering pretty much anything you need for common tasks.

DSL-oriented environment: Red comes with many embedded DSL addressing important needs (like GUI or system-programming). DSL are a very powerful way to reduce complexity arising from frameworks or API, while drastically increasing productivity. Red includes a DSL (called Parse) for constructing DSLs.

Red (like Rebol) is a Lisp derivative, but with a human-friendly syntax (no parenthesis hell). Red is its own data format. All code is treated as data until you evaluate it, code/data serialization comes for free. The minimal punctuation makes it easy on the eye.

#### **Q16):features of carriots?**

1.Carriots offers an end-to-end Internet of Things (IoT) platform designed for today's industry needs and tomorrow's innovations.

2.Carriots is a smart Platform as a Service (PaaS) designed for machine to machine (M2M) and digital twin projects.

3.Carriots accelerates your IoT application development and provides simple scalability as your projects and devices grow.



4. Carriots lets users collect & store data from connected devices, build powerful applications, deploy and scale from prototypes to thousands of devices.

**It's features are:**

- >Real-Time Interactive Visualization.
- >End-to-End Security Encryption.
- >User-Specified Data Engine Assignment.
- >Multi-Tenant Architecture.
- >Self-Service Interface.
- >Analytical Data Engine.
- >Multiple Data Source Reports.
- >Sparse Navigation
- >Embed Link Generation

**Q18):various tools available for security?**

We all know that data breaches are on the rise.

Which means that most people are increasing their cyber security IQ, right?

Unfortunately, that's not the case! According to a survey by Pew Research Center, the majority of people are still unclear about some critically important cyber security topics, terms and concepts.

**Four security tools that everyone should be using:**

**1.Firewalls**

A firewall is the first (of many) layers of defense against malware, viruses and other threats. It scrutinizes and filters both incoming and outgoing data.

Users can also customize rules and policies based on their needs.

For example, it's often necessary to create exceptions that allow certain apps to pass through the firewall so that they don't constantly trigger false alarms.

## **2.Antivirus Software**

Signature-based antivirus software scans files (from any source) to make sure that there aren't any hidden threats. And if it finds something shady or scary, it can often remove or quarantine the affected file. While antivirus software certainly isn't bulletproof — especially when it comes to zero-day threats (i.e. vulnerabilities that hackers have found before software vendors have a chance to patch them and/or users have a chance to install updates) — it's still a critical piece of the cyber security puzzle. There are many options to choose from that range in price from free to hundreds of dollars a year.

## **3.Anti-Spyware Software**

As the term implies, spyware secretly snoops on victims to see where they go online and, even more so, what they type — such as usernames and passwords, and any other confidential or personal data. That's where anti-spyware software fights back by (ideally) detecting and removing threats such as key loggers, password recorders, and so on.

## **4.Password Management Software**

Good password management software not only saves a great deal of time, but it strengthens security and prevents major mistakes, such as saving passwords in web browsers.

If you're looking for something to fit your needs and budget, here is a review of some popular options.

## **5.Also there are many more :**

>Wireshark (packet sniffer previously-known as Ethereal) ...

>Metasploit (exploit) ...

>Nessus (vulnerability scanner) ...

>Aircrack (WEP and WPA cracker) ...

>Snort (network intrusion detector) ...

>Cain and Abel (packet sniffer and password cracker) ...

>BackTrack (penetration tester) ...

>Netcat (debugger and exploration tool)

**Q19):Discuss URL with respect to:**

**1.Structure**

**2.Need**

**3.Defination**

**4.Example**

ans:

**1.Defination of url:**

What does URL stand for? URL is an abbreviation which stands for the term Uniform Resource Locator.

It contains a link to the server which is a storage of the searched resource.

In general, URL meaning is the track from the server to the final gadget

(which is a platform of the user's work) can be illustrated rather simply.

The upper element is the resource server, the lowest one – the user's device.

All the points in between the two are additional servers.

A URL is also a specific type of Uniform Resource Identifier (URI).

**2.Structure of url:**

URL address has a determined structure which includes:

method of access to the resource that is also named the network protocol;

access authorization;

**hosts** – DNS address that is inscribed as IP address;

**port** – one more obligatory detail included in combination with IP address;

**track** – determines the information about the method of gaining access;

**parameter** – the internal information of resource about the file.

**HTTP.** The first part is the name of the scheme. Then comes a colon and two slashes (/).

**WWW or webreference.** The second part is the name of the computer that hosts the document.

**:80.** The third part of the URL, which is optional, is the port number. Computers have a certain number of so-called ports. The meaning of the port is that through it there are interactions of a certain kind. One supports HTTP interactions, the other supports sending mail, and so on.

**something/something.html.** The fourth part is also optional. This is the path to the document we want to request. The path is a set of characters separated by slashes (/).

This is very similar to the paths to folders and files on your regular computer.

There is a root folder (directory), inside it, there are other folders, which, in turn, may contain other folders and files.

**?query.** The fifth part is the query string, which is also optional.

In fact, the query string is some kind of data intended for a certain program to process it and return the necessary information. The query string consists of a question mark (?)

Followed by the transmitted information (it completely depends on which program will process it).

### **3.Need for url:**

A URL (Uniform Resource Locator) is the reference point to your website and also the link that people refer to the most. ... If you want your customers to be attracted to your website and to keep returning regularly, you have to ensure your URL is simple, short and user-friendly.

### **4 EXAMPLE**

<https://portal.svkm.ac.in/usermgmt/loginSvkm>

**Q.20) Manoj wants to purchase an IoT device to secure his office. Give him some tips to avoid IoT disasters.**

Ans:

Common reasons of IoT failures:

1. They take too long to progress
2. The enterprise has limited expertise in IoT
3. The quality of data is subpar
4. There is a lack of integration across teams
5. Companies are plagued by budget overruns

Solutions:

- Leverage standards that ensure data coming from different sources is in sync
- Deploy secure devices and networks to prevent tampering with data
- Make sure devices in the field are operating as they should when gathering and sending data
  - Another step companies can take to succeed at IoT is to overcome any lack of integration across teams. IT and operations teams, like their respective leaders, must work closely together and coordinate their efforts as much as possible.
  - That includes collaborating on technology planning and purchasing. Operations are becoming more dependent on data and analytics, so it's increasingly important that IT and operations be on the same page as part of IoT initiatives. This is another area in which emerging IoT standards can help.
  - Finally, to enjoy success with IoT projects, manufacturers must do what is necessary to avoid budget overruns. That means getting maximum value and returns from all IoT investments and avoiding unnecessary expenses. Companies need to make smart buying decisions on both the IT and operations sides of IoT.

**Q. 21) Discuss major IoT security issues**

Ans:

1. Secure constrained devices
2. Authorize and authenticate devices
3. Manage device updates
4. Secure communication
5. Ensure data privacy and integrity

6. Secure web, mobile, and cloud applications
7. Ensure high availability
8. Detect vulnerabilities and incidents
9. Manage vulnerabilities
10. Predict and preempt security issues

Authorize and authenticate devices:

With so many devices offering potential points of failure within an IoT system, device authentication and authorization is critical for securing IoT systems.

Devices must establish their identity before they can access gateways and upstream services and apps. However, there are many IoT devices that fall down when it comes to device authentication, for example, by using weak basic password authentication, or using passwords unchanged from their default values.

Secure communication:

Once the devices themselves are secured, the next IoT security challenge is to ensure that communication across the network between devices and cloud services or apps is secure.

Many IoT devices don't encrypt messages before sending them over the network. However, best practice is to use transport encryption, and to adopt standards like TLS. Using separate networks to isolate devices also helps with establishing secure, private communication, so that data transmitted remains confidential.

Secure web, mobile, and cloud applications:

Web, mobile, and cloud apps and services are used to manage, access, and process IoT devices and data, so they must also be secured as part of a multi-layered approach to IoT security.

When developing IoT applications, be sure to apply secure engineering practices to avoid vulnerabilities such as the [OWASP top 10 vulnerabilities](#). Just like devices, apps should also support secure authentication, both for the apps themselves and the users of the applications, by providing options such as 2FA and secure password recovery options.

Predict and preempt security issues:

A longer-term IoT security challenge is to apply security intelligence not only for detecting and mitigating issues as they occur, but also to predict and proactively protect against potential security threats. [Threat modeling](#) is one approach used to predict security issues.

Other approaches include applying monitoring and analytics tools to correlate events and visualize unfolding threats in real-time, as well as applying AI to adaptively adjust security strategies applied based on the effectiveness of previous actions.

**Q. 22) Anisha is new in your area and she wants to some tips to make her smart home secure. What kind of tips will you provide to make her home more secure?**

Ans:

**Reroute:** Your router is the gateway to all your devices. If you haven't already reset your router after the FBI warning about malware a few months back, you should. Then rename it so that its brand and model are not immediately identifiable. Change the default password, if you haven't already done so. And set up a guest network for those who visit your home so that they don't have and retain access to your primary one.

**Turn on two-factor authentication:** Two-factor authentication (2FA) isn't a flawless way to protect yourself, but it's usually the first and best step you can take. Look for the feature in the manuals for any devices you have, including wearables. Turn off Wi-Fi Protected Setup so that potential hackers have a harder time trying to access your router.

**Don't give anyone a pass(word):** IoT devices generally come with a set user name and password. Change both on all your devices to avoid the easiest way for someone to gain control of them.

**Correct any defaults:** Go into the settings for all IoT devices and review the defaults. Turn off any that are unnecessary or that share more information than you'd like.

**Keep up to date:** Check for software updates on all your devices to make sure that any known vulnerabilities are patched. Even if you haven't received a notification about updates, visit the manufacturer's site and social media to see whether any have been issued.

**Q. 23) List different types of security measure that can protect your different types of IoT devices.**

Ans:

1. Install reputable internet security software on your computers, tablets, and smartphones. For instance, [Norton Security Deluxe](#) can provide real-time protection against existing and emerging malware, including ransomware and viruses.
2. Use strong and unique passwords for device accounts, Wi-Fi networks, and connected devices. Don't use common words or passwords that are easy to guess, such as "password" or "123456."
3. Be aware when it comes to apps. Always make sure you read the privacy policy of the apps you use to see how they plan on using your information and more.
4. Do your research before you buy. Devices become smart because they collect a lot of personal data. While collecting data isn't necessarily a bad thing, you should know about what types of data these devices collect, how it's stored and protected, if it is shared with third parties, and the policies or protections regarding data breaches.
5. Know what data the device or app wants to access on your phone. If it seems unnecessary for the app's functionality or too risky, deny permission.
6. Use a VPN, like [Norton Secure VPN](#), which helps to secure the data transmitted on your home or public Wi-Fi.
7. Check the device manufacturer's website regularly for firmware updates.
8. Use caution when using social sharing features with these apps. Social sharing features can expose information like your location and let people know when you're not at home. Cybercriminals can use this to track your movements. That could lead to a potential cyberstalking issue or other real-world dangers.
9. Never leave your smartphone unattended if you're using it in a public space. In crowded spaces, you should also consider turning off Wi-Fi or Bluetooth access if you don't need them. Some smartphone brands allow automatic sharing with other users in close proximity.

#### **24. Define / explain following in brief.**

- **Macro viruses :** A macro virus is a virus that is written in a macro language i.e. a programming language that is embedded inside a software application. When a software application is infected, it causes a sequence of actions to begin automatically when the application is opened.
- **File infectors :** These are file infecting viruses that usually copy their code onto executable files like .exe and .com. They replicate and spread, and might even damage



host programs.

- System or boot-record infectors : These infect executable code found on certain system areas on a disk. They attach to the master boot sector and the USB thumb drives or master boot records on hard disk.
- Polymorphic viruses : It's a kind of virus that uses polymorphic engine to mutate while keeping the original algorithm intact. It changes itself everytime it runs but its function does not change.
- Stealth viruses : It is a virus that uses various mechanisms to go undetected by any kind of anti-virus software.
- Trojans : It is a type of malware that is disguised as legitimate software. Users are tricked into loading and downloading this software by hackers trying to gain the users' personal information.
- Logic bombs : It is a piece of code that is deliberately inserted into a software so that it can set off some malicious function if certain requirements are met.
- Worms : A worm is a malicious, self replicating virus that can spread rapidly throughout a network without human assistance.
- Time Bombs :It is a part of a computer program which will start or stop functioning if a predetermined date or time has been reached.
- Ransomware : A kind of malicious software where the victim's computer data is locked and encrypted and a ransom is asked to restore access to that data.

**28. Rakhi wants to purchase an IoT device to secure his office. Give her some tips**

to avoid IoT disasters.

- Do not buy devices that are cheap and vulnerable.
- Do not trust blindly. Every device has its own defects.
- Only go for devices that offer security assurances.
- Ensure that the data you store is visible to only you. Enable private locks and security passwords.
- In case of open source data, do not leave the data unchecked.
- Never fail to use SSL layer and HTTPS protocol
- Take care of your personal belongings where you have stored your IoT data.

Example IMEI number

- There are several IoT wearable available. But choose only devices that are necessary to you, in your official or personal life. Never buy them just for the sake of experimenting. This could lead to trouble.

**Q26) Explain C0AP Protocol used in IOT?**

Ans) i) CoAP(Constrained Application Protocol) is an internet utility protocol for constrained gadgets.

ii) It is designed to enable simple, constrained devices to join IoT through constrained networks having low bandwidth availability.

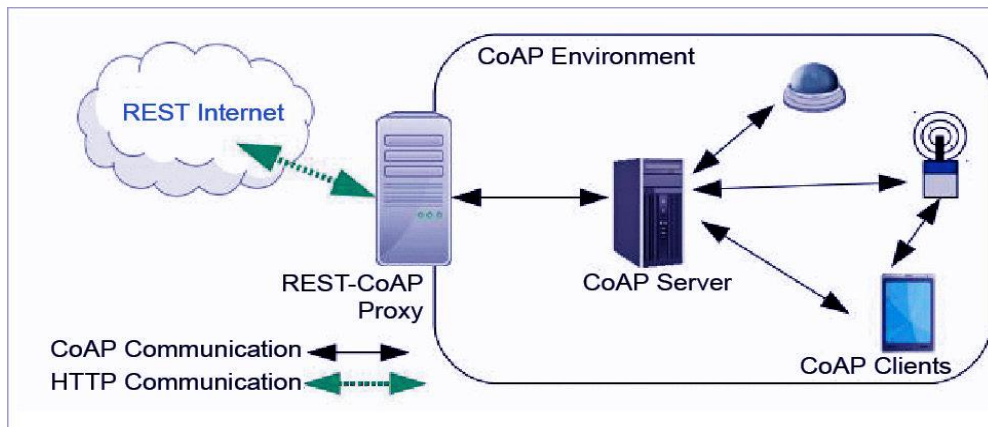
iii) This protocol is primarily used for machine-to-machine (M2M) communication and is particularly designed for IoT systems that are based on HTTP protocols.

iv) The main features of this protocol are:

- Web protocol used in M2M with constrained requirements

- Asynchronous message exchange
- Low overhead and very simple to parse
- URI and content-type support
- Proxy and caching capabilities

v) As you may notice, some features are very similar to HTTP. Although these similarities, CoAP must not be considered a compressed HTTP protocol because it is specifically designed for IoT. Therefore, it optimizes the M2M data exchange.



### Q27) State and Explain Various Type of Attacks

Ans)i) Many types of attacks have been around for a very long time.

ii) What's new is the scale and relative simplicity of attacks in the Internet of Things (IoT) – the millions of devices that are a potential victim to traditional style cyber attacks, but on a much larger scale and often with limited, if any protection.

iii) At its core, IoT is all about connecting and networking devices that up until now have not necessarily been connected. This means that all of those devices, whether it is your brand new connected refrigerator or your connected vehicle, are creating a new entry point to the network and therefore posing an increasing security and privacy risk.

A)Botnets: A botnet is a network of systems combined together with the purpose of remotely taking control and distributing malware. Controlled by botnet operators via Command-and-Control-Servers (C&C Server), they are used by criminals on a grand scale for many things: stealing private information, exploiting online-banking data, DDos-attacks or for spam and phishing emails. With the rise of the IoT, many objects and devices are in danger of, or are already being part of, so called thingbots – a botnet that incorporates independent connected objects.

B)Man-in-the-middle: The [man-in-the-middle](#) concept is where an attacker or hacker is looking to interrupt and breach communications between two separate systems. It can be a dangerous attack because it is one where the attacker secretly intercepts and transmits messages between two parties when they are under the belief that they are communicating directly with each other. As the attacker has the original communication, they can trick the recipient into thinking they are still getting a legitimate message. Many cases have already been reported within this threat area, cases of hacked vehicles and hacked "smart refrigerators".

C)Data and Identity Theft : While the news is full of scary and unpredictable hackers accessing data and money with all types of impressive hacks, we are often also our own biggest security enemy. Careless safekeeping of internet connected devices (e.g. mobile phone, iPad, Kindle, smartwatch, etc.) are playing into the hands of malicious thieves and opportunistic finders.

D)Denial of Service : A denial of service (DoS) attack happens when a service that would usually work is unavailable. There can be many reasons for unavailability, but it usually refers to infrastructure that cannot cope due to capacity overload. In a Distributed Denial of Service (DDoS) attack, a large number of systems maliciously attack one target. This is often done through a [botnet](#), where many devices are programmed (often unbeknownst to the owner) to request a service at the same time.

## **Q29)Discuss the Working of HTTP Protocol**

Ans)i) HTTP full form *HyperText Transfer Protocol* used mainly to access data on the World Wide Web.

ii) HTTP is a Server and Client communication Protocol, which is primarily set of rules for forming and transferring webpage data (text, images, video and Multimedia files) over the world wide web.

iii) This is the Protocol used to create communication between Web Servers and Web Users.

iv) The working of the HTTP protocol is explained step by step as follows:

**Step 1: Establishing a TCP/IP connection by the client.**  
The first step is initiating a TCP connection with the server by the client. Once the connection established, the browser and the server access TCP through their socket interfaces.

**Step 2: Initiating an HTTP GET request to the server by the client.**

The http request first line of the message is called the request line. The lines below the request line are known as header lines.

The request line has three fields- method, URL, and version. Method field can take several values like GET, POST, HEAD, PUT and DELETE, etc. The GET method used when the browser requests an object, with the requested object identified in the URL field.

**The meaning of all these values is:**

**GET:** it retrieves those document which identified in the URL.

**POST:** it will give information to the server.

**HEAD:** it retrieves the meta information about the document identified in the URL.

**PUT:** it stores those documents which are underspecified URL.

**DELETE:** it deletes the specified URL.

**TRACE:** it will loopback the request message.

**Step 3: HTTP Server Response to an HTTP GET Request.**

HTTP response message has three sections- status line, header lines, and an entity body. Further, the status line has 3 fields- version, status code, and phrase. After then, header lines are there. In the end, there is the entity body, which contains the requested message itself.

### **Q30. Smart health care**

#### **Ans.**

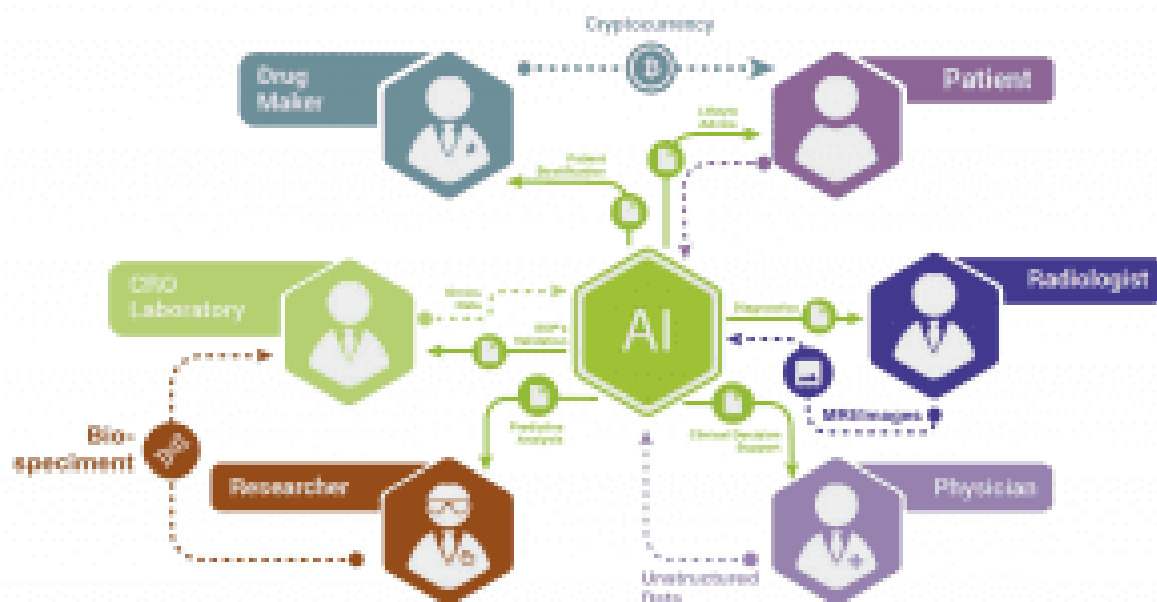
‘Smart Health Technology’ combines Smart Technology and latest mobile device with health. Nowadays, numerous initiatives have been designed to encourage a broader view of health and wellbeing thus smart wearables devices like fitness tracker or fitness bands and even health assessment apps in smartphones have gained grand attention amongst fitness enthusiasts. These devices are smart in the sense as they not only just monitor health but also provide solutions if needed at the right time. Smart devices act as the base of smart healthcare.

#### **Role of technologies in Smart Healthcare**

Connectivity provides the foundation of smart city services and it acts as an enabler of smart healthcare as well. With the help of this citizens are able to communicate with authorities easily and it helps authorities as well to gather more health data of its citizens which in return can be used to inform further city and service planning that makes public healthcare a priority. Here IoT plays an important role.

IoT in healthcare allows connecting data collected from smart devices and sensors to extract valuable insights. The technology can play a foremost role in healthcare observation and help in early detection of health issues. It would also help in assimilating the data collected from tests instantly, monitor the condition of the patient, and then convey that information to the doctors and staff in real-time, thus improving the effectiveness in the overall healthcare system. In the

near future personal IoT-based health checking devices will change the way, we track the health of individuals.



When the health data is collected it needs to be analyzed and managed for accurate treatment and here Artificial intelligence and automation are applied.

AI is also applied to perform tasks like analyzing laboratory tests, x-rays, CT scans, and data entry. AI-based apps can be used to access the current medical condition of patients that can provide assistance in medical consultation.

Technologies like Blockchain redefine the methods of maintaining and populating Electronic Health Records as well as they also help link them to other services like payments and insurance.

These advanced technologies are very critical in making healthcare a more determinate process, with concrete results, with a service that is more pertinent to the lifestyle of the modern citizen. Continuing innovation and improved data analysis will also help make it an area of constant enhancement that will continue to invent new ways of keeping people fitter and healthier.

### Q31. Smart Farming

Ans.

**Smart Farming** is an emerging concept that refers to managing farms using modern Information and Communication Technologies to increase the quantity and quality of products while optimizing the human labor required.

we'll talk about two major areas of agriculture that IoT can revolutionize:

1. Precision farming
2. Farming automation/robotization

### 1. Precision Farming

Precision farming, or precision agriculture, is an umbrella concept for IoT-based approaches that make farming more controlled and accurate. In simple words, plants and cattle get precisely the treatment they need, determined by machines with superhuman accuracy. The biggest difference from the classical approach is that precision farming allows decisions to be made per square meter or even per plant/animal rather than for a field.

By precisely measuring variations within a field, farmers can boost the effectiveness of pesticides and fertilizers, or use them selectively.

### 2. Precision Livestock Farming

As in the case of precision agriculture, smart farming techniques enable farmers better to monitor the needs of individual animals and to adjust their nutrition accordingly, thereby preventing disease and enhancing herd health.

Large farm owners can use wireless IoT applications to monitor the location, well-being, and health of their cattle. With this information, they can identify sick animals, so that they can be separated from the herd to prevent the spread of disease.

## Automation in Smart Greenhouses

Traditional greenhouses control the environmental parameters through manual intervention or a proportional control mechanism, which often results in production loss, energy loss, and increased labor cost.

IoT-driven smart greenhouses can intelligently monitor as well as control the climate, eliminating the need for manual intervention. Various sensors are deployed to measure the environmental parameters according to the specific requirements of the crop. That data is stored in a cloud-based platform for further processing and control with minimal manual intervention.

## Agricultural Drones

Agriculture is one of the major verticals to incorporate both ground-based and aerial drones for crop health assessment, irrigation, crop monitoring, crop spraying, planting, soil and field analysis and other spheres.

### Q32. Guidelines to tackle IOT threats

Ans.

The Internet of Things (IoT) is no longer a fad. It is here to stay. But as you'd expect with technology that thrives in inter-connectivity, it can be a target of malicious programs and attacks. Combine this with BYOD policies of multiple companies and you can have a security nightmare on your hands. Hackers can target the weakly secured devices your employees bring to the office and, assuming they're connected to your corporate networks, can use them as a [gateway into your systems](#).

It is not difficult to see that more attacks in the future will target IoT technology. The very thing that makes it so appealing – its ability to connect various devices and systems - also makes it susceptible to attacks. In addition to the devices themselves being affected, either used as a backdoor for hackers or enslaved as part of a [botnet](#), they can also put sensitive information in danger of being illegally accessed or intercepted while in transit.



Well, here are some tips:

### Limit IoT Devices at Work

Just because there is a BYOD policy, it shouldn't mean that employees can just bring any device they have and connect it to the office network.

IoT wearables, in particular, have [several security vulnerabilities](#) that can put an organization in danger of breaches. Many of these devices store and transmit data without encryption, often with no password or biometric authentication. It also connects to your smartphone through unsecure connections like Bluetooth or NFC, making it particularly vulnerable to brute-force attacks even more.

### Use a Separate Network

You know when you create a separate network that can only be accessed by guests so that they have limited to no access to your business' main network?

You can do the same for IoT devices. You can create a separate network that is dedicated to you and your staff's IoT devices. This way, you are allowing the use of such devices within your premises so your employees are happy, without putting your main network at risk.

### Use Strong and Unique Passwords

Like any security measure, it always starts with a strong password. The same goes for IoT security. Encourage your employees to use strong and unique passwords, especially if they are connecting their devices over a Wi-Fi network.

### Do Not Use Universal Plug and Play

Most IoT devices have universal plug and play (UPnP) features that make them easier to get connected to other devices. It makes it pretty easy for different devices like routers, printers, cameras, and others to discover and connect with each other without complex configurations.

### Always Update Firmware

Just because you have security features on your device doesn't mean you will automatically be safe.

Like your PC software, it is good practice to always update your IoT devices' firmware. These patches address bugs and other security-related issues, which are always evolving. Neglecting these updates makes it easier for your device's security to fail because it is unable to recognize new forms of attacks.

Q33. Difference between IOT and M2M.

Ans.

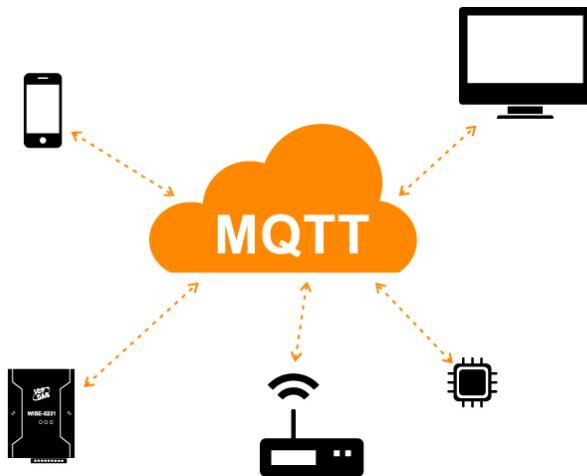
M2M versus the IoT	
M2M	IoT
M2M is about direct communication between machines.	The IoT is about sensors automation and Internet platform.
It supports point-to-point communication.	It supports cloud communication.
Devices do not necessarily rely on an Internet connection.	Devices rely on an Internet connection.
M2M is mostly hardware-based technology.	The IoT is both hardware- and software-based technology.
Machines normally communicate with a single machine at a time.	Many users can access at one time over the Internet.
A device can be connected through mobile or other network.	Data delivery depends on the Internet protocol (IP) network.

Q34) Explain the working of protocol which is responsible for publishing and subscribing.

Ans)

MQTT is one of the most commonly used protocols in IoT projects. It stands for Message Queuing Telemetry Transport.

In addition, it is designed as a lightweight messaging protocol that uses publish/subscribe operations to exchange data between clients and the server. Furthermore, its small size, low power usage, minimized data packets and ease of implementation make the protocol ideal of the "machine-to-machine" or "Internet of Things" world.



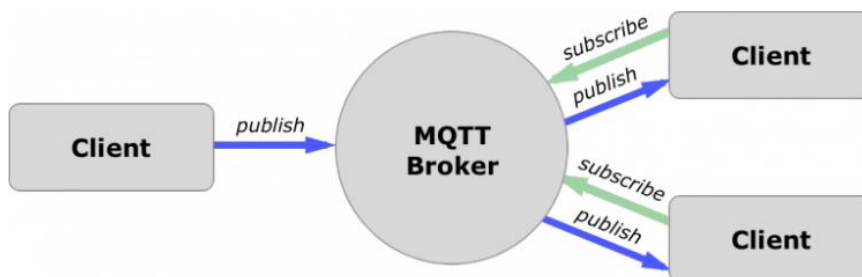
Like any other internet protocol, MQTT is based on clients and a server. Likewise, the server is the guy who is responsible for handling the client's requests of receiving or sending data between each other.

MQTT server is called a broker and the clients are simply the connected devices.

So:

When a device (a client) wants to send data to the broker, we call this operation a “publish”.

When a device (a client) wants to receive data from the broker, we call this operation a “subscribe”.



In addition, These clients are publishing and subscribing to topics. So, the broker here is the one that handles the publishing/subscribe actions to the target topics.

**Q35) Draw and explain the working of coap protocol used in IOT.**

Ans)

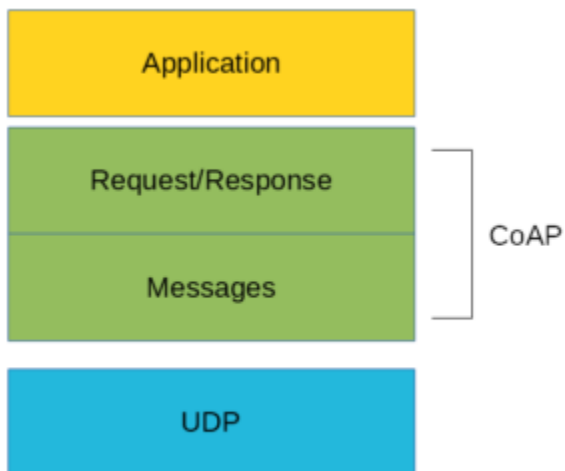
Constrained Application Protocol (CoAP) is a specialized web transfer protocol for use with constrained nodes and constrained networks in the Internet of Things. CoAP is designed to enable simple, constrained devices to join the IoT even through constrained networks with low bandwidth and low availability. It is generally used for machine-to-machine (M2M) applications such as smart energy and building automation.

Working:

CoAP functions as a sort of [HTTP](#) for restricted devices, enabling equipment such as sensors or actuators to communicate on the IoT. These sensors and actuators are controlled and contribute by passing along their data as part of a system. The protocol is designed for reliability in low bandwidth and high congestion through its low power consumption and low network overhead. In a network with a lot of congestion or limited connectivity, CoAP can continue to work where [TCP](#)-based protocols such as [MQTT](#) fail to exchange information and communicate effectively.

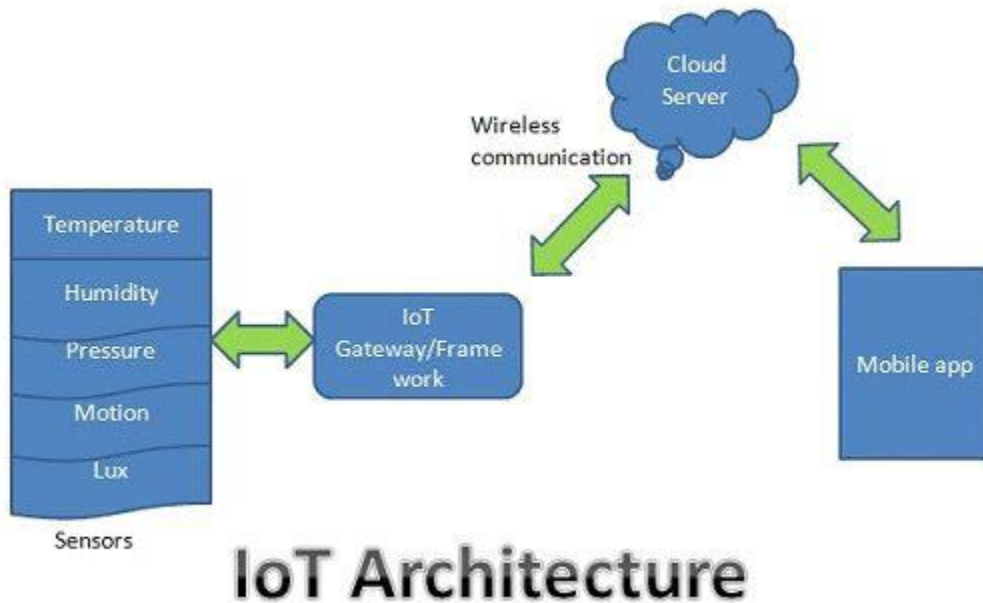
Additionally, the effective and conventional CoAP features enable devices operating in poor signal quality to send their data reliably or enable an orbiting satellite to maintain its distant communication successfully. CoAP's also supports networks with billions of nodes. For security, the DTLS parameters chosen for default are an equivalent to 3072 bit [RSA](#) keys.

COAP uses [UDP](#) as the underlying network protocol. COAP is basically a client-server IoT protocol where the client makes a request and the server sends back a response as it happens in HTTP. The methods used by COAP are the same used by HTTP.



**Q36) Explain the working of IOT with the help of a neat labelled diagram.**

Ans)



**Q37) Discuss major IOT security issues.**

Ans)

#### 1. Insufficient testing and updating

Currently, there are over 23 billion IoT connected devices worldwide. This number will further rise up to reach 30 billion by 2020 and [over 60 billion](#) by the end of 2025. This massive wave of new gadgets doesn't come without a cost.

In fact, one of the main problems with tech companies building these devices is that they are too careless when it comes to handling of device-related security risks.

Most of these devices and IoT products don't get enough updates while, some don't get updates at all.

This means that a device that was once thought of as secure when the customers first bought it becomes insecure and eventually prone to hackers and other security issues.

Early computer systems had this same problem, which was somewhat solved with automatic updates.

IoT manufacturers, however, are more eager to produce and deliver their devices as fast as they can, without giving security too much of a thought.

This leaves their trusted customers exposed to potential attacks as a result of outdated hardware and [software](#).

To protect their customers against such attacks, each device needs proper testing before being launched into the public and companies need to update them regularly.

Failing to do so is bad for both the companies and their consumers, as it only takes a single large-scale breach in consumer data to completely ruin the company.

## **2. Brute-forcing and the issue of default passwords**

The Mirai botnet, used in some of the largest and most disruptive DDoS attacks is perhaps one of the best examples of the issues that come with shipping devices with default passwords and not telling consumers to change them as soon as they receive them.

There are some government reports that advise manufacturers against selling IoT devices that come with default (read, hackable) credentials such as using “admin” as username and/or passwords.

That said, these are nothing more than guidelines now, and there aren’t any legal repercussions to incentivize manufacturers to abandon this dangerous practice.

Weak credentials and login details leave nearly all IoT devices vulnerable to password hacking and brute-forcing in particular.

The only reason why Mirai malware was so successful is that it identified vulnerable IoT devices and used default usernames and passwords to log in and infect them.

Therefore, any company that used factory default credentials on their devices is placing both their business and its assets and the customers and their valuable information at risk of being susceptible to a brute-force attack.

## **3. IoT malware and ransomware**

As the number of IoT connected devices continues to rise in the following years, so will the number of malware and ransomware used to exploit them.

While the traditional ransomware relies on encryption to completely lock out users out of different devices and platforms, there’s an ongoing hybridization of both malware and ransomware strains that aims to merge the different types of attack.

The ransomware attacks could potentially focus on limiting and/or disabling device functionality and stealing [user data](#) at the same time.

For example, a simple IP camera is ideal for capturing sensitive information using a wide range of locations, including your home, work office or even the local gas station.

The webcam can then be locked and footage funneled to an infected web address which could extract sensitive data using the malware access point and demand ransom to unlock the device and return the data.

The ever-increasing number of IoT devices will give birth to unpredictability in regards to future attack permutations.

#### **4. IoT botnets aiming at cryptocurrency**

The heated mining competition, coupled with the recent rise of cryptocurrency valuations is proving too enticing for hackers trying to cash in on the crypto-craze.

While most find blockchain resistant to hacking, the number of attacks in the blockchain sectors seems to be increasing.

IoT applications, structures, and platforms relying on [blockchain technology](#) need to become regulated and constantly monitored and updated if it were to prevent any future cryptocurrency exploits.

#### **5. Data security and privacy concerns (mobile, web, cloud)**

Data privacy and security continues to be the single largest issues in today's interconnected world.

Data is constantly being harnessed, transmitted, stored and processed by large companies using a wide array of IoT devices, such as smart TVs, speakers and lighting systems, connected printers, HVAC systems, and smart thermostats.

Commonly, all this user-data is shared between or even sold to various companies, violating our rights for privacy and [Data security](#) and further driving public distrust.

#### **6. AI and automation**

As IoT devices continue to invade our everyday lives, enterprises will eventually have to deal with hundreds of thousands, if not millions of IoT devices.

This amount of user-data can be quite difficult to manage from a data collection and networking perspective.

#### **7. Home Invasions**

Perhaps one of the scariest threats that IoT can possess is of the home invasion. Nowadays, IoT devices are used in a large number at homes and offices which has given rise to the home automation.

The security of these IoT devices is a huge matter of concern as it can expose your IP address that can pinpoint to your residential address.

This vital information can be sold by the hackers to the underground websites which are havens for criminal outfits.

Moreover, if you're using IoT devices in your security systems, then there is a possibility that they might compromise as well as leave your house at a huge potential threat.

#### 8. Remote vehicle access

Apart from home invasion, hijack of your car is also one of the threat possessed by the IoT.

Smart cars are on the verge of becoming reality with the help of connected IoT devices. However, due its IoT association, it also possesses a greater risk of a car hijack.

A skilled hacker might hijack by getting the access of your smart car through the remote access. This will be scary situation as anyone can have control over your car and it can leave you vulnerable to lethal crimes.

#### 9. Untrustworthy communication

There are many IoT devices which send messages to the network without any encryption. This is one of the biggest IoT security challenge which exists out there.

It's high time that all the companies ensure encryption of the highest level among their cloud services and devices.

To avoid this threat, the best way to do is to use transport encryption and standards like TLS. Another way is to use different networks that isolates different devices. You can also use private communication which ensures that the data transmitted is secure and confidential.

#### **Q38) List and Explain the various properties of IOT devices.**

- **Ans: Connectivity:** Internet connectivity is either available within the device itself or can be provided by a hub, smartphone or access point (base station). If connectivity is provided by an access point then it is most probably collecting data and operational information from a range of sensors for a specific device and then communicating with the cloud to relay this information.



- **Sensors:** Sensors are an important aspect of devices and systems within the internet of things. The sensors monitor, track and measure the activity and interactions of a device and then relay this information using the cloud. Some examples of such sensors include ones that monitor a person's health and fitness or sensors that can detect whether a door has been opened in your house or even ones that monitor usage statistics (eg: for utilities).
- **Processors:** Devices in the Internet of Things will need to have a certain degree of computing power even if it is just to be able to relay and transmit gathered data to the cloud. Like any computing device this will require a processor which will handle such tasks, so all devices connected to the internet of things will have a processor as well.
- **Security:** Devices connected to the Internet of Things may be transmitting information that maybe highly sensitive and regulated, an example is health related or financial data. Since a variety of sensitive data can be relayed by these devices it is very important that there is good data security as this is vital.
- **Energy-efficiency, Quality & Reliability:** Devices in the IoT may be operating in severe weather, unforgiving environments and hard to reach places for instance devices operating in outer space or deep inside mines. Since the devices may be operating in such environments it is important that they are made with the highest quality, are reliable and energy efficient (so batteries don't have to be charged or changed regularly).
- **Cost effectiveness:** Things or devices that require sensors to relay useful information may need to be circulated in a large quantity in order to be effective, hence they will need to be cost effective. For instance if it is a sensor attached to food products to monitor the expiry dates, it will need to be implemented into every single product – in addition to being disposable they will need to be relatively cheap to implement.

### Q39) State the reason why MQTT protocol used in IOT and explain its working

MQTT is one of the most commonly used protocols in IoT projects. It stands for Message Queuing Telemetry Transport.

In addition, it is designed as a lightweight messaging protocol that uses publish/subscribe operations to exchange data between clients and the server. Furthermore, its small size, low power usage, minimized data packets and ease of implementation make the protocol ideal of the “machine-to-machine” or “Internet of Things” world.

MQTT has unique features you can hardly find in other protocols, like:

- It's a lightweight protocol. So, it's easy to implement in software and fast in data transmission.
- It's based on a messaging technique. Of course, you know how fast your messenger/WhatsApp message delivery is. Likewise, the MQTT protocol.
- Minimized data packets. Hence, low network usage.
- Low power usage. As a result, it saves the connected device's battery.
- It's real time! That's is specifically what makes it perfect for IoT applications.

Like any other internet protocol, MQTT is based on clients and a server. Likewise, the server is the guy who is responsible for handling the client's requests of receiving or sending data between each other.

MQTT server is called a broker and the clients are simply the connected devices.

So:

- When a device (a client) wants to send data to the broker, we call this operation a “publish”.
- When a device (a client) wants to receive data from the broker, we call this operation a “subscribe”.

In addition, These clients are publishing and subscribing to topics. So, the broker here is the one that handles the publishing/subscribing actions to the target topics.

Let's say there is a device that has a temperature sensor. Certainly, it wants to send his readings to the broker. On the other side, a phone/desktop application wants to receive this temperature value. Therefore, 2 things will happen:

- The device defines the topic it wants to publish on, ex: “temp”. Then, it publishes the message “temperature value”.
- The phone/desktop application subscribes to the topic “temp”. Then, it receives the message that the device has published, which is the temperature value.

Again, the broker role here is to take the message “temperature value” and deliver it to phone/desktop application.

That takes us to the MQTT components, which are 5 as follows:

- Broker, which is the server that handles the data transmission between the clients.
- A topic, which is the place a device want to put or retrieve a message to/from.
- The message, which is the data that a device receives “when subscribing” from a topic or send “when publishing” to a topic.
- Publish, is the process a device does to send its message to the broker.
- Subscribe, where a device does to retrieve a message from the broker.

#### **Q40)What do you mean by IOT? Explain the components of the same.**

The **Internet of things (IoT)** is a system of interrelated computing devices, mechanical and digital machines are provided with unique identifiers (UIDs) and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction.<sup>[1][2][3][4]</sup>

- The definition of the Internet of things has evolved due to the convergence of multiple technologies, real-time analytics, machine learning, commodity sensors, and embedded systems.<sup>[1]</sup> Traditional fields of embedded systems, wireless sensor networks, control systems, automation (including home and building automation), and others all contribute to enabling the Internet of things. In the consumer market, IoT technology is most synonymous with products pertaining to the concept of the "smart home", covering devices and appliances (such as lighting fixtures, thermostats, home security systems and cameras, and other home appliances) that support one or more common ecosystems, and can be controlled via devices associated with that ecosystem, such as smartphones and smart speakers.

#### **1. Gateway**

Gateway enables easy management of data traffic flowing between protocols and networks. On the other hand, it also translates the network protocols and makes sure that the devices and sensors are connected properly.

It can also work to preprocess the data from sensors and send them off to next level if it is configured accordingly. It is essential to configure it as the presence of TCP/IP protocol allows easy flow.

Not only this, it gives proper encryption with the network flow and data transmission. The data flowed through it is in the higher order that is protected by using latest encryption techniques. You can assume it like an extra layer between the cloud and devices that filter away the attack and illegal network access.

## 2. Analytics

The analog data of devices and sensors are converted into a format that is easy to read and analyze. This is all possible due to the IoT ecosystem that manages and helps in improving the system. The main factor that is influenced is security.

The most important function of IoT technology is that it supports real-time analysis that easily observes the irregularities and prevents any loss or scam. Preventing the malicious things to attack the smart devices will not only give you a sense of security but also it will save all your private data from being used for illegal purposes.

The big companies collect the data in bulk and analyze it to see the future opportunity so that they can easily develop more business advancement and gain something out of it. This analysis easily helps in setting future trends that have a capability to rule the market. From this analysis, they can be one step ahead of the time and easily achieve success. Data may be a small word but it holds the power to make or break the business if used correctly.

## 3. Connectivity Of Devices

The main components that complete connectivity layer are sensors and devices. Sensors collect the information and send it off to the next layer where it is being processed. With the advancement of technology, semiconductor technology is used that allows the production of micro smart sensors that can be used for several applications.

The main components are:

- Proximity detection,
- Humidity or Moisture Level,
- Temperature sensors and thermostats,
- Pressure sensors,
- RFID tags.

The modern smart sensors and devices use various ways to be connected. The wireless networks like LORAWAN, Wi-Fi, and Bluetooth makes it easy for them to stay connected. They have their own advantages and drawbacks that are classified in various forms like efficiency rate, data transfer, and power.

## 4. Cloud

With the help of internet of things ecosystem, companies are able to collect bulk data from the devices and applications. There are various tools that are used for the purpose of data collection that can collect, process, handle and store the data efficiently in real time. It is also responsible for making a tough decision that can easily break the deal. This all is done by one system that is IoT Cloud.

It is an intimidating high-performance network that connects servers together to optimize the performance of data process that is being processed by many devices at once. It also helps in controlling traffic and delivering accurate data analytics results.

One of the most important components of IoT cloud is database management that is distributed in nature. The cloud basically combines many devices, gateways, protocols, devices and a data store that can be analyzed efficiently. These systems are used by many companies in order to have improved and efficient data analysis that can help in the development of the services and products. In addition to this, it also helps in forming an accurate strategy that can help in building an ideal business model.

## **5. User Interface**

This is another factor on which IoT ecosystem depends immensely. It provides a visible and physical part that can be easily accessed by the user. It is important for the developer to create a user-friendly interface that could be accessed without putting any extra efforts in it and that can help in easy interaction.

With the help of advancement, there are various interactive designs that could be used easily and that can easily solve any complex query. For examples, at home people have started to use the colorful touch panels instead of the hard controls that were used earlier. It is increasing day by day as now the touchpads are also launched that can switch on the air conditioners from a distance.

This has set out a trend for the digital generations and have managed to hype up today's competitive market. The user interface is the first thing that user pay attention to before buying a device. Even customers are oriented to buy the devices that are user-friendly and less complex that could be used with wireless connectivity.

## 6. Standards And Protocols

The webpages are now using the HTML format with the cascading style sheet. This has made the internet more stable and reliable service to use. They are the most used standard protocols making it not just friendly but easily acceptable. However, IoT doesn't have that standard.

It is important to choose a platform for IoT that can help in determining the way your platform will interact with the system. Thus, you will be able to have an interaction with devices and networks with the same standard as yours. It is important for having the same protocol to have a successful interaction.

## 7. Database

Internet of things are increasing dynamically and is all dependent on data that are used immensely in the data centers. It is essential to have a proper database system that can store and manage the data that is being gathered from various devices and end-users. There are also various management tools that offer many automated features that help in easy accumulation of data stored and managed in bulk at the same place.

## 8. Automation

As mentioned above, the database system is using the automated features that help in managing data and accumulating it. However, the data management is the only limited thing that is used by the internet of things. It is now used for a much more advanced version that allows the automatic adjustment of the wireless things. For example, you can easily control light with a click of the remote. The air conditioner is now connected to your smartphone and you can switch them on and off whenever you want. Even it is possible to play with the temperature.

## 9. Development

Internet of things is the latest advancement in technology. The need for the development is growing and increasing with time. Each and every one is not depending on the launch of various automated devices and smart sensors. There are various prototypes that are in the market that are being deployed and are running in the testing phase. Also, IoT is not working with only one device. Hence, it is important that the devices are completely tested according to the compatibility of the device and checked thoroughly that whether the devices can connect wirelessly or not.

The journey of the internet of things is growing for years. We have managed to experience many things and advancement in most of the technology. IoT ecosystem is used to make the protocols easily accessible, reasonably priced, efficient and secure. It will be excited to see the new demands and development in the several sectors that will be bought by the internet of things.

Especially the way it connects different companies and vendors together. The main is that we need to lookout the way everyone can incorporate this IoT ecosystem to increase their production.

#### **Q41)State and explain various types of attack?**

##### **. Physical cyber-attacks**

These attacks result from breaches to the IoT device's sensors. [Click to read more about vulnerabilities of IoT embedded devices.](#)

**It's estimated that approximately 70% of all cyber-attacks are initiated from the inside,** whether purposeful or the result of human error.

With an IoT physical cyber-attack, the hacker most often accesses the system through close proximity, like inserting a USB drive.

Tampering can enable the intruder to take over the controls, extract data, and/or infuse the system with malicious code (similar to malware) that opens a door to the system without being noticed.

Hackers can also strike with a distributed denial of service (DDoS) that basically shuts down the system. Another physical cyber-attack hits the batteries in the devices and the system. While you think you have them set to sleep mode, the power is actually draining from the batteries.

##### **2. Network cyber-attacks**

These don't require physical access to create a major disruption—like DDoS—in your network.

These attackers infiltrate your network devices to see what's flowing. They can insert themselves between you and your devices (known as "Man in the Middle" or "MitM"), creating fake identities, stealing information, and redirecting packets to their desired location, away from your network (also referred to as a "sinkhole" attack).

##### **3. Software attacks**

The third area that poses an IoT security risk is your software. Software attacks occur when malware is installed into your network's program. This malicious software sends a virus, corrupts or steals data, and can both interrupt and spy on the activities. A software attack can launch a DDoS, too.

##### **4. Encryption attacks**

Finally, **encryption attacks** strike at the heart of your algorithmic system. Hackers analyze and deduce your encryption keys, to figure out how you create those algorithms. Once the encryption keys are unlocked, cyber-assailants can install their own algorithms and take control of your system.

Consequently, it is essential that IoT users maintain an awareness of these cyber risks and put preventative measures in place

**Q42) State the security measure which can use to keep IOT device safe?**

1. Install reputable internet security software on your computers, tablets, and smartphones. For instance, [Norton Security Deluxe](#) can provide real-time protection against existing and emerging malware, including ransomware and viruses.
2. Use strong and unique passwords for device accounts, Wi-Fi networks, and connected devices. Don't use common words or passwords that are easy to guess, such as "password" or "123456."
3. Be aware when it comes to apps. Always make sure you read the privacy policy of the apps you use to see how they plan on using your information and more.
4. Do your research before you buy. Devices become smart because they collect a lot of personal data. While collecting data isn't necessarily a bad thing, you should know about what types of data these devices collect, how it's stored and protected, if it is shared with third parties, and the policies or protections regarding data breaches.
5. Know what data the device or app wants to access on your phone. If it seems unnecessary for the app's functionality or too risky, deny permission.
6. Use a VPN, like [Norton Secure VPN](#), which helps to secure the data transmitted on your home or public Wi-Fi.
7. Check the device manufacturer's website regularly for firmware updates.
8. Use caution when using social sharing features with these apps. Social sharing features can expose information like your location and let people know when you're not at home. Cybercriminals can use this to track your movements. That could lead to a potential cyberstalking issue or other real-world dangers.
9. Never leave your smartphone unattended if you're using it in a public space. In crowded spaces, you should also consider turning off Wi-Fi or Bluetooth access if you don't need them. Some smartphone brands allow automatic sharing with other users in close proximity

**Q42) State the security measures which can be used to keep IoT devices safe.**

**#1 Understand the Advantages of Connecting to the Internet**

Don't connect your smart device to the internet just because it has the capability. First you should check what features are available in your device without connecting it to the internet. You might discover that your smart device has good features which are available without internet connection. In that scenario, it is better to use the device offline. This is a good way of protecting your security without having to spend anything.

**#2 Use Secondary Network**

Most of the time, your WIFI router is capable of creating multiple networks which helps you to create restricted access for your family and your guests. You should also think of creating an



extra network just for your internet of things devices. This will help you to curb unauthorised access to your sensitive data when you are accessing your connection through your smart devices. Having a separate connection to act like a buffer will help to ensure that no outside entity is allowed to access your shared files and other kinds of encrypted data.

### **#3 Keep Changing your Passwords**

It is important that you keep changing your passwords on your PC's, individual accounts and mobile devices. You must be knowing this. What you should also remember is that it is equally important to change the passwords you use on your internet of things devices. You should be diligent with these passwords and ensure that each device has a unique password. You can use a password manager to remember your passwords or even use the traditional method of pen and paper. Remember that every password must be changed twice every year.

### **#4 Don't Enable Universal Plug & Play Features**

Nearly all smart devices have a feature, called as UPnP. With this feature, different devices can find one another and connect to each other. This features makes the devices more convenient because you don't need to configure each of these devices separately. What you need to be aware, however is that UPnP protocols make use of local networks for connecting and are thus vulnerable to outside access. In case there is an attack, outside entities might be able to gain access to multiple devices simultaneously. Hence, it is a good practice to turn off the UPnP feature on every device.

### **#5 Update your every Device**

You should enable automatic updates, if it is available and in its absence, check regularly for firmware updates. This is necessary as this is the way new security patches are installed on your devices. As hackers and other intruders are consistently coming up with new ways to hack IoT devices, technology manufacturers are involved in the constant effort to counteract such threats with security measures. Thus, making sure that all your devices are updated will help to strengthen the security in your house as a whole.

### **#6 Reduce your use of Cloud Technology**

As you might know, most IoT providers provide free cloud storage with their devices. The reason for this is that it is considered as emerging technology. However, there are some potential drawbacks with this technology. The first one is that you need an active connection if you want to access the data that you store in the cloud; that information cannot be accessed without a connection. Secondly, outsiders might be able to hack into your connection as you are accessing your account. So, before you start using the cloud, make sure that you are familiar with the ways to secure your data and thoroughly understand the privacy measures.

### **#7 Be Careful of Where you Take your Wearables**

Wearable devices use WIFI connectivity to collect and store personal data, so that it can later provide you with an accurate analytics. Hence, when you take your wearable device to a public place, as it connects with the public WIFI, your data becomes instantly accessible to whoever is connecting to the same network. So, it is a safe bet to avoid taking you wearables which has a

shared network. In case you do take it, ensure that you disable the device when you are not using it.

### **# 8 Avoid using Devices that are Risky**

There are some IoT devices that need to be constantly connected to the internet to operate effectively. There are also some that do not bring as much efficiency to your life as you had thought of. It is, therefore, a good practice to evaluate the usefulness of your devices from time to time. If you feel that they don't add anything to your life or if they seem risky, better discard them. Additionally, if there is a single device that can replace several others, invest in that device. This is because the smaller number of devices that you use, the less is the risk factor.