IoT

10th Jan 2022

```
#include <stdio.h>
#include <GL/glut.h>
#include <math.h>
```

28/Jan/22

✱ Internet of Things :-

- Lets demonstrate in term by term :
  - A thing is any object or device besides a traditional computer, that has some features and performs some function, for eg, a refrigerator or air conditioner.
  - To this thing, we add computational intelligence to improve its functionality.
    - The computational intelligence could be of any type, typically a microcontroller that runs some code.
  - We then add network connectivity - internet connection to enhance its function further.
  - This device is called an IoT device, a thing that doesn't look like a computer, but has a computer hidden within it.

✱ An Intelligent Device vs IoT device :-

- An intelligent air conditioner might be able to perform tasks like :
  - Using sensors to detect human presence in the room, & switching on or off on that basis.
  - Beeping or hooting when it requires servicing or repair.
  - This an intelligent device uses local sensors to collect physical data or requires locally stored data to enhance the device functionality without network connectivity.
- An IoT air conditioner has access to the internet and hence can :-

- Acquire information about when you come home via GPS and switch on a few minutes before you arrive so that the temperature is set to an ambient value.
- Monitor external weather conditions from weather data sites to improve its own efficiency and to determine the inside temperature.
- Book an appointment for servicing and repair online, instead of just indicating need for repair.
- Give your information to sites that market to you.

* IoT ?
- A network connecting (either wired or wireless) devices, or 'things', that is characterized by autonomous provisioning, management, and monitoring. The IoT is innately analytical and integrated (IDC)
- IoT is the next evolution of the Internet, connecting the unconnected people, processes, data and things in your business today (cisco)
- IoT devices as those capable of two-way data ~~transmission~~ transmission (excluding passive sensors and RFID tags). It includes connections using multiple communications methods such as cellular, short range and others. (GSMA)
- Sensors & actuators connected by networks to computing systems. These systems can monitor or manage the health and actions of connected objects and machines. Connected sensors can also monitor the natural world, people & animal " (McKinsey).

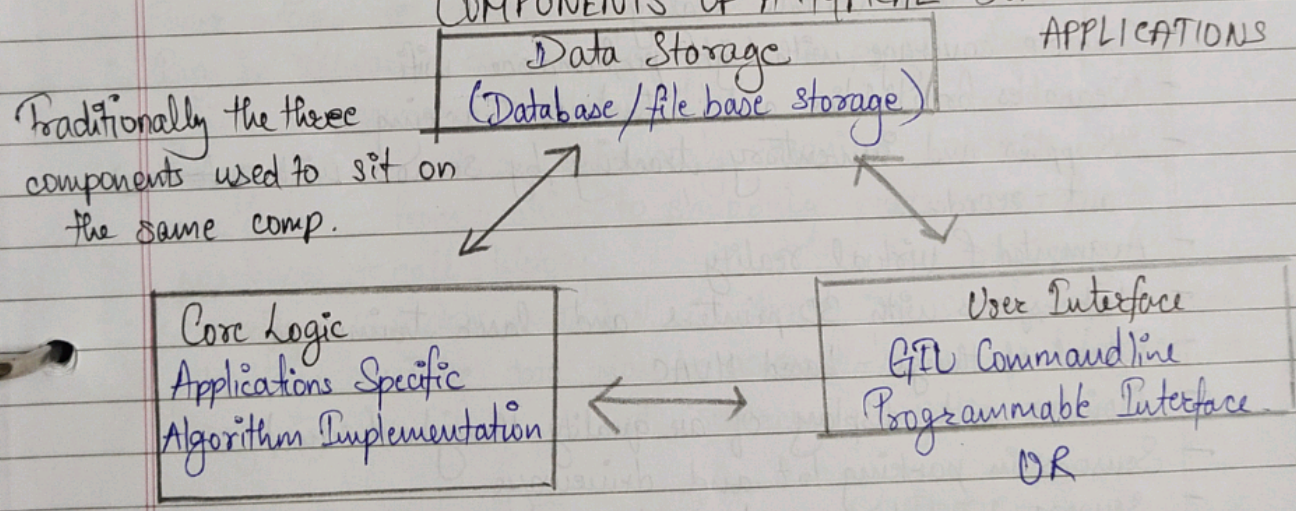
★ IoT (IEEE)?

- An IoT system is a network of networks where, typically, a massive number of objects, things, sensors or devices are connected through communications and information infrastructure to provide value-added services via intelligent data processing and management for different applications (e.g. smart cities, smart health, smart grid, smarthome, smart transportation, and smart shopping.)

what is mainly IoT?

- IoT is a concept encompassing various platforms, applications, and technologies implemented under a number of radio communications services.

COMPONENTS OF A TYPICAL SOFTWARE APPLICATIONS



Traditionally the three components used to sit on the same comp.

★ Various names, One Concept

- M2M (Machine to Machine)
- "Internet of Everything" (Cisco Systems)
- "World Size Web" (Bruce Schneier)
- "Skynet" (Terminator movie)

★ Examples of IoT :- Wearable Tech, Smart Appliances, Healthcare.

IoT Example (1)

☆ The Smart Internet of Things School :-

- Personalized learning with adaptive eTextbooks
- Digital classroom interactive boards and display.
- Video recorders for lecture capture
- International colaboration and social exchange
- Online Testing
- Student devices & eTextbooks : Notebooks, Tablets and Smartphones
- Sensors on trash receptacles
- Robot Cleaning
- Robotics for STEM and remote presence
- Surveillance security cameras
- Network applications analytics to monitor devices and network behaviour
- Augmented and virtual reality
- Complete coverage with high performance wifi
- Wearables for athletics and attendance tracking
- Supplies and inventory tracking by sensors with auto-reorder.
- Augmented & virtual reality
- Makerspaces with 3D printers and laser trimmers
- Internet of things - based HVAC
- Monitor and display of air quality throughout school.
- Sensors in parking lot and driveways
- Sensors track buses and verify student passengers.
- Wifi sensors & locks - Entrances and exists
  - Classroom doors
- File and program storage, local or cloud - based
  - Demographics, academics, behaviour, interests
  - LMS, CMS, SIS
  - Educational programs and applications
  - Video files : lectures and recorded lab experiments.

IoT Example (2)

★ Scanning and Logistics

- Logistics and tagging : From 1970's, barcode scanning came into picture to keep track of boxes and products.
- RFID tag is an electronic version of barcode.
- It can store not only a unique identity tag for a box, but also data related to when and where the box moved over time.
- A barcode can't change over time while an RFID can have flash memory that can't change.
- Can be easily network connected.

★ The other concern & Risks : Privacy and Security

- Dependency on devices, not people can lead to social isolation.
- Power outage, device failure or network failure cause problems.
- Bugs in software - can be dangerous as people depend on devices like a pacemaker for their lives.
- Privacy - IoT devices continuously monitor, observe and record our data - from location to shopping preferences, from purchases to call history.
- Security - The data acquired is stored in cloud servers and potentially sensitive data may be accessed by unwanted people or organizations if it is not securely encrypted.

★ Top 10 Trends Research areas in IOT :-

1. Trend No. 1 : Artificial Intelligence (AI) :
   - Gartner forecasts that 14.2 billions connected things will be in use [(28 Jan 2022) 40:39]
2. Trend No 2 : Social, Legal, and Ethical IoT.
3. Trend No 3 : Infonomics and Data Broking.
4. Trend No 4 : The Shift from Intelligent Edge to Intelligent Mesh

5. Trend No: 5 : IoT Governance
6. Trend No 6 : Sensor Innovation
7. Trend No. 7 : Trusted Hardware and operating System
8. Trend No. 8 : Novel IoT user Experiences
9. Trend No 9 : Silicon Chip Innovation
10. Trend No 10 : New Wireless Networking Technologies for IoT

28 Jan 2022

★ How is IoT device different from a computer?

Computer	IoT device
• A computer executes program to perform computation	• Has a main function apart from computation, and uses computation only to improve that main functionality.
• A computer is general-purpose. Not particularly efficient for a particular operation but generally good at everything.	• For example, a car is meant to drive people around, and this function is enhanced by computational intelligence.
• You can use a phone to capture a photo as well as make phone calls, but a dedicated DSLR can do the job of capturing a photo better.	• IoT devices are special-purpose things - software and hardware are efficient for one task but inefficient for other.

☆ Technological trends leading to IoT.

- Reduction in cost - computational technology is lot cheaper.
- Reduction in size - smaller, lighter technology that can be easily incorporated in several 'things'.
- Increased computational ability - ~~computal~~ computers can do more and more things they weren't able to few years back, and at higher speed.
- Improved internet access (also wireless internet access).

☆ Trends in IoT :-

- Computational ability :
  - A standard laptop processor can execute billions of ~~inst~~ instructions per second
  - Higher clock rates
  - Network connectivity
  - Internet Access ; Wired and wireless ; depends on geography : This is making IoT more available and affordable.
  - Transmit large data
    - With Audio and Video processing :
    - Can stream multiple movies in real time.
- So the bandwidth actually has enabled a lot of different features in IoT devices.

☆ What is IoT ? (Advantages maybe?) (Online Source)

- Access to low-cost, low-power sensor technology: Affordable and reliable sensors are making IoT technology possible for more manufacturers.
- Connectivity : A host of network protocols for the internet has made it easy to connect sensors to the cloud and to other

"things" for efficient data transfer.

- Cloud computing platforms :- The increase in availability of cloud platforms enables both businesses and consumers to access the infrastructure they need to scale up without actually having to manage it all.
- Machine learning and analytics :- With advances in machine learning and analytics, along with access to varied and vasts amounts of data stored in the cloud, businesses can gather insights faster and more easily. The emergence of these allied technologies continue to push ~~to~~ the boundaries of IoT and the data produced by IoT also feeds these technologies.
- Conversational artificial intelligence (AI) :- Advances in neutral networks have brought natural - language processing (NLP) to IoT devices and made them appealing, affordable, and viable for home use.

(Next para but ~~no~~ plagarism remover used)

- Industrial IoT (IIOT) refers to the use of IoT technology in industrial settings, particularly in relation to metal fabrications and control of sensors and devices involving cloud technology.
- Recently, industries have been using machine - to - machine communication (M2M) to achieve automation and wireless control. But with the advent of cloud and related technologies, industries can achieve a new level of automation and thus generate new revenue and business models. IIOT is sometimes referred to as the fourth wave of industrial revolution, or Industrial 4.0. The following are some common uses of IIOT.

- Smart manufacturing
- Connected assets and preventive and predictive maintenance
- Smart power grids
- Smart cities
- Connected logistics
- Smart digital supply chains.

[Indian Railways Assignment] 31st Jan 2022

27 Jan 2022

★ Cloud Computing :-
- Innovation not in the technology
- Innovation in the application of the technology
- Cloud computing consists of
  - Development of self contained components
  - Delivering these components as services.
- Similar to utilities like electricity, mobile network
  - Pay-per-use, without large infrastructural cost.

★ Cloud as a Service :-
- Software as a Service (SaaS)
- Infrastructure as a Service (IaaS)
- Platform as a Service (PaaS).

★ Internet of Things :- (ppt) (without sourse)

Lets deconstruct it term by term :-

- A thing is any object or device besides a traditional computer, that has some features and performs some functions, for example, a refrigerator or air conditioner.
- To this thing, we add computational intelligence to improve its functionality.
  - The computational intelligence could be of any type, typically a microcontroller that runs some code.
- We then add network connectivity - internet connection to enhance its function further.
- This device is called an IoT device, a thing that doesn't look like a computer, but has a computer hidden within it.

Example ★ Efficiency in Paramount :-

- Embedded systems are not designed to only perform a task, but to do it at high speed, with low power consumption and greater efficiency.
- This is because most embedded products are :
  - Cost-critical : manufacturing cost, design cost, time to market.
  - Life-critical : medical or safety devices on which life of user depends, such as pacemaker, car-reverse system etc. Performance and power key constraints.

★ Discussion focus Area

- Embedded & Cyber-Physical Systems
- Internet of Things
- Digital Platforms.

☆ Embedded CPS?

- Cyber Physical Systems not (well) understood by the public and politicians
  - NOT related to Cyberspace
  - NOT related to Cybercrime
  - ~~Related~~ to Cybernetics,
  - For the public and politicians we better use the expression Embedded Intelligence.

☆ Cyber-Physical Systems vs Embedded Systems (Already without plgiarism)

- As indicated by Edward Lee at all * Cyber-Physical Systems and Embedded Systems are in most cases identical but different in their (model) description including the various continuous dynamics of the involved physical processes, often described using differential equations in combination with models of software.

☆ Embedded Systems :-

- Embedded Systems are everywhere
  - Ubiquitous, invisible
  - Hidden (computer inside)
  - Dedicated purpose
- ~~Microcontro~~ Microprocessor
  - Intel : 4004, ... 8080, ... x86
  - freescale : 6800 ..... ~~9~~ 9S12, ... PowerPC
  - ARM, DEC, SPARC MIPS, PowerPC, Natl. Semi, ...
- Microcontroller
  - Processor + Memory + I/O Ports (Interfaces)

☆ Components of a microprocessor / controller.
- CPU : Central Processing Unit
- I/O : Input / Output
- Bus : Address bus & Data Bus
- Memory : RAM & ROM
- Timer
- Interrupt
- Serial Port
- Parallel Port

#2 - IOT Target Board Programming

☆ Objective :-
- Know your lab equipment.
  - Hands on Experiment / Demo with the use of laboratory
  - Recognize discrete components and ICs.
  - Read and understand the data sheets.
- Setting up a target Board environment.
  - Writing the first sketch.

☆ Voltage (Volts)
- Voltage is the measure of work required to move a unit charge from one location to another, against the force which tries to keep electric charges balanced.
- In the context of electrical power sources, voltage is the amount of potential energy available (work to be done) per unit charge, to move charges through a conductor.
- Because voltage is an expression of potential energy, representing the possibility or potential for energy release as the charge moves from one "level" to another, it is always referenced between two points.

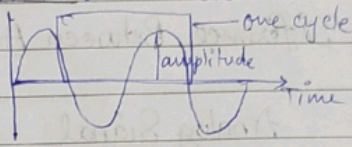
★ Current (Amperes) :-

- The flow rate of charge with respect of time is known as electric current.

★ Alternating Current :-

- If the high & low voltage terminals switch locations periodically, the current will flow "back and forth" in the circuit. This is called alternating current (AC)

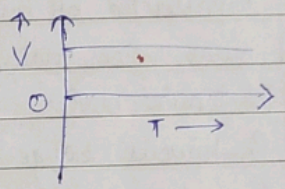
T = Period in sec ; F = 1/T Hertz



★ Direct Current :-

- If the voltage is maintained between two points in a circuit, charge will flow in one direction - from high to low potential. This is called direct current (DC)

Frequency = 0 Hz



★ AC and DC :-

- Most of the home appliances run on DC.
- Each home appliance is equipped with its own power supply.

★ Some notations, Symbols units.

Electrical Parameter	Measuring Unit	Symbol	Descriptions
Voltage	Volt	V or E	Unit of Electrical Potential $V = I \times R$
Power	Watts	W	Unit of Power $P = V \times I$ or $I^2 \times R$
Impedance	Ohm	Z	Unit of AC Resistance $Z^2 = R^2 + X^2$
Frequency	Hertz	Hz	Unit of frequency $f = 1 \div T$
Resistance	Ohm	R or Z	Unit of DC Resistance $R = V \div I$
Current	Ampere	I or i	Unit of Electric Current $I = V \div R$

★ Digital vs Analog? :-

- Information in the electrical form ready to transmit - Signal
- Digital has two value : on and off
- Analog has many (infinite) values
- Computers don't really do analog, they quantize

★ Difference Between Analog and Digital Signal.

Analog Signal	Digital Signal
- Continous Signals	Discrete Signals
- Represented by sine waves	Represented by square waves
- Human voice, natural sound, analog electronic devices are few examples	Computers, optical drives, and other electronic devices
- Continous range of values	Discontinous values
- Records sound waves as they are	Converts into a binary waveform
- Only be used in analog devices	Suited for digital electronics like computers, mobiles and more.

★ Breadboard :-

- All holes in a side column are connected
- All hole in end row are connected
- Each row of 5 holes is connected
- Used for fast, non-permanent wiring
- Holes fit 24-gauge wire and, 0.1 inch spacing } uses.
  - Common prototype sizes

☆ Safety Instructions :-

- Electricity, when improperly used, is very dangerous to people and to equipment.
- Basic knowledge of Injuries
  - Electrical shock
  - Electrical burns
  - falls caused by electrical shock.
- Temperature
  - Energy dissipation most of the times is in the form of temperature.
  - Appropriate Heat sinks, fans are essential to main the temp.

☆ Resistor :-

- Each color has a number, as follows :-
  - Black 0
  - Brown 1
  - Red 2
  - Orange 3
  - Yellow 4
  - Green 5
  - Blue 6
  - Violet 7
  - Grey 8
  - White 9

- The first two striped are the first two digits of the value, so red, red means 2,. The next stripe is the number of zeros that need to come after the first two digits, so if the third stripe is brown, as it is in the photograph above, then there will be one zero and so the resistor is 220 Ω.

- A resistor with stripes brown, black, orange is 10 and three zeros so 10,000 Ω in other words 10 kΩ.
- Unlike LEDs, resistors do not have a positive & negative lead. They can be connected either way around.

11th Feb 2022

★ Test your knowledge :-

- Complete of One full pattern - one cycle
- The information in digital data - discrete - low & high - true/false
- The information in analog data - Continuous
- mili Ampere - 1/1000th - 10^-3

★ What is Microcontroller ?

- A small computer on a single chip containing a processor, memory, and input/output.
- Typically "embedded" inside some device that they control
- A microcontroller is often small and low cost.

★ What is a development Board ?

- A printed circuit board (PCB) designed to faciliate work with a particular microcontroller.
- Components are :-
  - power ~~cir~~ circuit
  - programming interface
  - basic input ; usually buttons and LEDs
  - I/O pins.

★ What is the Arduino ?

- Arduino is an open-source platform for prototyping hardware using the Atmel Atmega series of microcontrollers.
- It uses its own coding environment.
- The Arduino IDE and the programs written for Arduino are called sketches.

- The Arduino basically means :-
  1. A physical piece of hardware
  2. A programming environment
  3. A community and philosophy.

- ~~Physical Com~~

✱ Why was Arduino developed?

- Physical Computing - using components that can interact with people and with the world around us.
- The Arduino was originally developed for artists and designers to prototype interactive displays
  - Developed for non-scientists
  - Minimalist programming
  - "forgiving" circuitry that can handle a wide variety of wiring errors.

✱ Arduino Hardware :-

- This microcontroller comes from a company called Atmel and the chip is known as an AVR.
- It is slow in modern terms, running at only 16MHz with an 8-bit core, and has a very limited amount of available memory, with 32 kilobytes of storage and 2 kilobytes of random access memory.
- Its design is not entirely new or revolutionary, beginning with a curious merger of two, off-the-shelf reference designs, one for an inexpensive microcontroller and the other for a USB-to-serial converter.

☆ Arduino UNO Specifications :-

- The ATmega328 has many more capabilities than the Arduino provides. Arduino has :-
  - Atmel ATmega 328 MCU
  - 32K bytes flash program memory
  - 1K byte ~~EEPROM~~ EEPROM
    [EEPROM - Electronically erasable programmable read-only memory]

☆ Different Arduino Specifications :-

1. LilyPad

	different Types	
Processor :-	:- ATmega168V	ATmega328P
Operating/Input Voltage :-	:- 2.7-5.5 V	/ 2.7-5.5 V
CPU Speed :-	:- 8MHz	
Analog Input/output :-	:- 6/0	
Digital Input/Plus Width Modulation :-	14/6	
EEPROM	:- 0.512	
Static Random Access Memory (SRAM) :-	1	
Flash (kB)	:- 16	
USB (Universal Serial Bus)	:- -	
[UART] Universal Asynchronous receiver transmitter	:- -	

2. UNO

Processor	-	ATmega 328P
O/I Voltage	-	5V/7-12V
CPU Speed	-	16 MHz
Analog I/O	-	6/0
Digital IO/PWM	-	14/6
EERPOM	-	1
SRAM	-	2
Flash (kB)	-	32
USB	-	Regular
UART	-	1

3. Nano :-

Processor	- ATmega168 · ATmega328P	
D/I Voltage	- 5V/7-9V	
CPU Speed	- 16 MHz	
Analog I/O	- 8/0	
Digital IO/PWM	- 14/6	
EERPOM	- 0.512	1
SRAM	- 1	2
Flash (kB)	- 16	32
USB	- Mini	1
UART	-	

Microprocessor vs Microcontroller →

Microprocessor	Microcontroller
• CPU is stand-alone, RAM, ROM, I/O, timer are separate.	• CPU, RAM, ROM, I/O and timer are all on a single chip
• Designer can decide on the amount of ROM, RAM and I/O ports	• fixed amount of on-chip ROM, RAM, I/O ports
• Expensive	• for applications in which cost, power and space are critical
• Versality	• single purpose (control oriented)
• General purpose	• Low processing power
• High processing power	• Low power consumption
• High power consumption	• Bit level operations
• Instruction sets focus on processing-intensive operations	• Instruction sets focus on control and bit-level operations
• Typically 32/64-bit	• Typically 8/16 bit
• Typically deep pipeline (5-20 stages)	• Typically single-cycle / two-stage pipeline

14th Feb 2022

★ The Arduino Platform :- (The circuit board)

- Arduino UNO is a microcontroller board based on the ATmega328P
- It has 14 digital input/output pins (of which 6 can be used as PWM outputs)
- 6 analog inputs.
- a 16 MHz quartz crystal
- a USB connection
- a Power Jack
- an ICSP header and a reset button

★ Getting Started :-

- It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC to DC adapter or battery to get started.

- "Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0. The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer released.

★ Add an External LED to pin 13

- LED's have polarity
  - Negative indicated by flat side of the housing and a short leg.

★ Sketch :-

- Two default methods that are part of any Arduino sketch are :-
  - void setup() // runs first time as code is executed, and only once.

- void loop() // runs iteratively until program is terminated.
- Pins on the Arduino can be used as both input and output, specified by the function pinMode() in setup.

★ Getting Started about Arduino Platform

- The Arduino ecosystem begins with the Arduino platform, itself several layers of hardware and software working together to create a cohensive whole.

- We can start with the hardware ~~today~~ interface board - that little, blue circuit board that you build into your projects

- It has a fairly standard onboard microcontroller that can interact with the world around it by using its programmable inputs and outputs, as well as a port and controller for easily communicating with your computer

★ Using Arduino IDE :-

- Write your sketch
- Press Compile button (to check for errors)
- Press upload button to program Arduino board with your sketch

★ Terminology :-

- "sketch" - a program you write to run on an Arduino Board.
- "pin" - an input or output connected to something eg: output to an LED, input from a knob.
- "digital" - value is either HIGH or LOW

- "analog" - values ranges, usually from 0-255
  eg LED brightness, motor speed, etc.

☆ Program #1

```
#include <avr/io.h>
#include <util/delay.h>
int main(void){
while (1){
PORTB = 0x20;
_delay_ms(1000);
PORTB = 0x00;
_delay_ms(1000);
}
return 1;
}
```

Listing 1-2. Blink LED with Arduino

```
void setup(){
pinMode(13, OUTPUT)
}
Void loop(){
digitalWrite(13, HIGH);
delay(1000);
digitalWrite(13, LOW);
delay(1000);
}
```

★ Digital I/O

pinMode (pin, mode)
Sets pin to either INPUT or OUTPUT

digitalRead (pin)
Reads HIGH or LOW from a pin

digitalWrite (pin, value)
Writes HIGH or LOW to a pin

Electronic stuff

Output pins can provide 40 mA of current
Writing HIGH to an input pin installs a 20 kΩ pullup

★ LAB 1.1 LED Switching On or Off.

```
void setup() {
  // put your setup code here, to run once:

    pinMode (13, OUTPUT);
    digitalWrite (13, HIGH);
}
void loop () {
    // put your main code here, to run repeatedly;
}
```

☆ Arduino Timing

- delay (ms)
  - Pauses for a few milliseconds
- delayMicroseconds (us)

☆ LAB 1.2 LED blinking on and off / LED blinking on & off (any other digital pin)

```
void setup() {
    // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);   // turn the LED on
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);              // wait for a second.
}
```

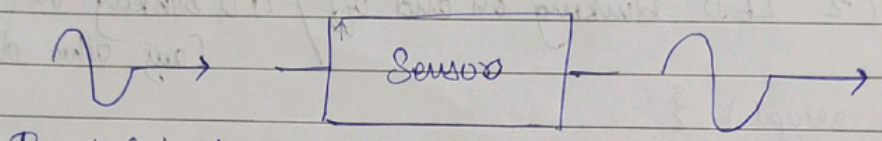
24th Feb 2022

☆ Sensors :-

- A sensor detects (senses) changes in the ambient conditions or in the state of another device or a system, and forwards or processes this information in a certain manner.
- "A device which detects or measures a physical property and records, indicates, or otherwise responds to it"

☆ What are Sensors?

- American National Standards Institute (ANSI) definition
  - A device which provides a usable output in response to a specified measurand
- A sensor acquires a physical parameter and converts it into a signal suitable for processing (eg. optical, electrical, mechanical)



Input Signal — Output Signal

☆ Detectable Phenomenon :-

Stimulus	Quantity
Acoustic	Wave (amplitude, phase, polarization), Spectrum, Wave Velocity
Biological and Chemical	Fluid Concentrations (Gas or Liquid)
Electric	Charge, Voltage, Current, Electric field, Conductivity, Permittivity
Magnetic	Magnetic field (amplitude, phase, polarization), Flux, Permeability
Optical	Refractive Index, Reflectivity, Absorption
Thermal	Temperature, Flux, Specific Heat, Thermal Conductivity
Mechanical	Position, Velocity, Acceleration, force, Strain, Stress, Pressure, Torque.

← continuation

← ☆ What are Sensors? ←

- The main difference between sensors are transducer is that a transducer is a device that can convert energy from one form to another, whereas a sensor is a device that can detect a physical quantity and convert the data into an electrical signal. Sensors are also a type of transducers (eg touch, sight, etc)

- A transducer
  Microphone, Loud Speaker, Biological Senses.

☆ Sensors vs Transducer :-

- The term transduce is a compound word used for sensors that can be used to sense many types of energy such as motion, electric signals, frequency, thermal or magnetic forces and actuators that can be used to change voltages or currents.

☆ Physical Principles :-

- **Amperes' Law :-**
  - A current carrying conductor in a magnetic field experiences a force (eg galvanometer)

- **Curie - Weiss Law :-**
  - There is a transition temperature at which ferromagnetic materials exhibit paramagnetic behaviour.

- **Faraday's Law of Induction :-**
  - A coil resist a change in magnetic field by generating an opposing voltage/current (eg. transformer).

- Photoconductive Effect :-
  - When light strikes certain semiconductor materials, the resistance of the material decreases (eg. photoresistor)

☆ Need for Sensors :-

- Sensors are omnipresent. They are embedded in our bodies, automobiles, airplanes, cellular telephones, radios, chemical plants, industrial plants and countless other applications.
- Without the use of sensors, there would be no automation

☆ Choosing a Sensors :-

Environmental Factors	Economic Factors	Sensors Characteristics
Temperature range	Cost	Sensitivity
Humidity effects	Availability	Range
Corrosion	Lifetime	Stability
Size		Repeatability
Overrange protection		Linearity
Susceptibility to EM interferences		Error
Ruggedness		Response time
Power consumption		Frequency response
Self-test capability		

★ 25 Feb 2022

★ Types of Sensors

1. DIRECT :-
   - A sensor that can convert a non-electrical stimulus into an electrical signal with intermediate stages.
     - Thermocouple (temperature to voltage)

2. INDIRECT :-
   - A sensor that multiple conversion steps to transform the measured signal into an electrical signal.
     - A fibre-optic displacement sensor:
     - Current → photons → current

★ Digital Sensors :-

- Digital Sensors produce discrete digital output signals or voltages that are a digital representation of the quantity being measured.
- Digital Sensors produce a binary output signal in the form of a logic "1" or a logical "0", ("ON" or "OFF").
- Digital signal only produces discrete (non-continuous) values, which may be output as a single "bit" (serial transmission) or by combining the bits to produce a single "byte" output (parallel transmission).

* Analog Sensor

- Which is generally proportional to the quantity being measured.
  - Physical quantities such as temperature, Speed, Pressure, Displacement, Strain, etc. are all analog quantities as they tend to be continuous in nature.

for example, the temperature of a liquid can be measured using a thermometer or thermocouple (e.g in geysers) which continuously responds to temperature changes as the liquid is heated up or cooled down.

* Scaler Sensors :-

- Scaler Sensors produce output signal or voltage which is generally proportional to the magnitude
- Physical values such as temperature, color, pressure, type, etc, they are all scalar as their size is enough to convey information
- for Eg, the room temperature can be ~~measured~~ measured using a thermometer or thermocouple, which responds to it temperature change regardless of the shape of its sensor or direction.

* Vector Sensors :-

- Vector Sensors produce an output signal or voltage which is normal in proportional to the size, direction, and position of estimated quantity.

- Physical values such as sound, image, speed, acceleration, shape etc. they are all vector values, as their magnitude is absent enough to convey complete information.

- For example, body speed can be measured using an accelerometer, which provides parts for body acceleration, which provides parts for body accerate acceleration with respect to axes connecting x, y, z.

★ Sensors features :-

- It is only sensitive to the measured property.
- It is insensitive to any other property likely to be encountered in its application.
- Does not influence the measured property.

- Sensor Resolution :
  - Resolution of sensor is the smallest change it can detect in the quantity that it is measuring.
  - More the resolution of a sensor the accurate is its precision.
  - Sensors accuracy ~~dep~~ does not depend upon its resolution

★ Sensor Types :-

- Light - Light Dependent resistor
  - Photo Diode
- Temperature - Thermocouple
  - Thermistor
- Force - Strain gauge
  - Pressure Switch
- Position - Potentiometer, Encoders
  - Opto - coupler
- Speed - Reflective / Opto - coupler
  - Doppler effect sensor
- Sound - Carbon Microphone
  - Piezoelectric Crystal.

- Chemical - Liquid Chemical sensor
  - Gaseous chemical sensor

★ Digital I/O

pinMode (pin, mode)
Sets pin to either INPUT or OUTPUT

digitalRead (pin)
Reads HIGH or LOW from a pin.

digitalWrite (pin, value)
Writes HIGH or LOW to a pin.

Electronic stuff
Output pins can provide 40 mA of current
Writing HIGH to an input pin installs a 20k-Ω pullup

25th feb 2022 (2)

★ Comparison Operators :-

==	equal to
!=	not equal to
>	greater than
<	less than
>=	greater than or equal to
<=	less than or equal to

☆ Sensors vs Transducers :-

- The word transduce is a collective term used sensors that can be used to sense scope the range of forms of energy such as movement, electric, wide, thermal or magnetic signals emery and actuators can be used change the voltages or currents.

☆ Arrays and Loops :-

- for
[Control Structure]

- Definition -
  - The for statement is used to repeat a block of statements that are embedded in curly instruments
  - An increament counter is usually used to enlarge and ~~loop~~ terminate the loop. The for statement is useful for any repetitive operation, and is often used in combination with arrays to operate on collections of data / pins.
- Syntax :-

for (initialization ; condition ; increament)

{ //statement(s) ; }

The initialization happens first and exactly once. Each time through the loop, the condition is tested ; if it's true, the statement block, and the increment is executed, then the condition is tested again. When the condition becomes false, the loop ends.

- Array
  [Data Types]

- Description
  - An array is a collection of variables that are accessed with an index number. Arrays in the C programming language, on which Arduino is based, can be complicated, but using simple arrays is relatively straightforward.

- Creating (Declaring) an Array

```
int myInts[6];
int myPins[] = {2, 4, 8, 3, 6};
int mySensVals[6] = {2, 4, -8, 3, 2};
char message[6] = "hello";
```

★ buzzer :-

```
// put your setup code here, to run once;
/* Arduino tutorial - Buzzer / Piezo Speaker

const int buzzer = 9; //buzzer to arduino pin 9
void setup(){
  pinMode(buzzer, OUTPUT); //Set buzzer - pin 9 as an output
}
void loop(){
  tone(buzzer, 1000); //Send 1KHz sound signal
  delay(1000);        // for 1 sec
  noTone(buzzer);     //Stop sound...
  delay(1000);        // for 1 sec
}
```

28th Feb 2022

★ LDR (Light Dependent Resistor)

- Hardware Required
- UNO
- LED / Built in
- LDR - Photoresistor
- Resistors
- Wires
- Breadboard

Program :-

```
const int ledPin = 13;
const int ldrPin = A0;
void setup () {
Serial.begin (9600);
pinMode (ledPin, OUTPUT);
pinMode (ldrPin, INPUT);
}
void loop () {
int ldrStatus = analogRead (ldrPin);
if (ldrStatus <= 300) {
digitalWrite (ledPin, HIGH);
Serial.println ("LDR is DARK, LED is ON");
}
  else {
digitalWrite (ledPin, LOW);
Serial.println ("--------");
  }
```

★ Actuators :-

- It is a device that causes something to happen in the physical world.
- Outputs of the IoT device
  - Visual : LED, LCD, Monitor
  - Audio : Speaker
  - Motion : Motors, Valve, Pump
  - Tactile : Heating and cooling.
- On - Off Actuation
  - The only control is power (current)
- Current Limits
  - both devices and MCU
  - Ex : LED 20mA
- Arduino can only supply 40mA
  - Can not drive a motor that require Amperes.

★ Analog Voltage Control :-

- Many actuators need an analog voltage control
- DC motor speed controlled by voltage.
- LED brightness can be controlled by voltage.
- Arduino cannot generate analog outputs. A digital to Analog converter can be externally connected to it.
- Other option is with PWM.

4th March 2022

Program :-

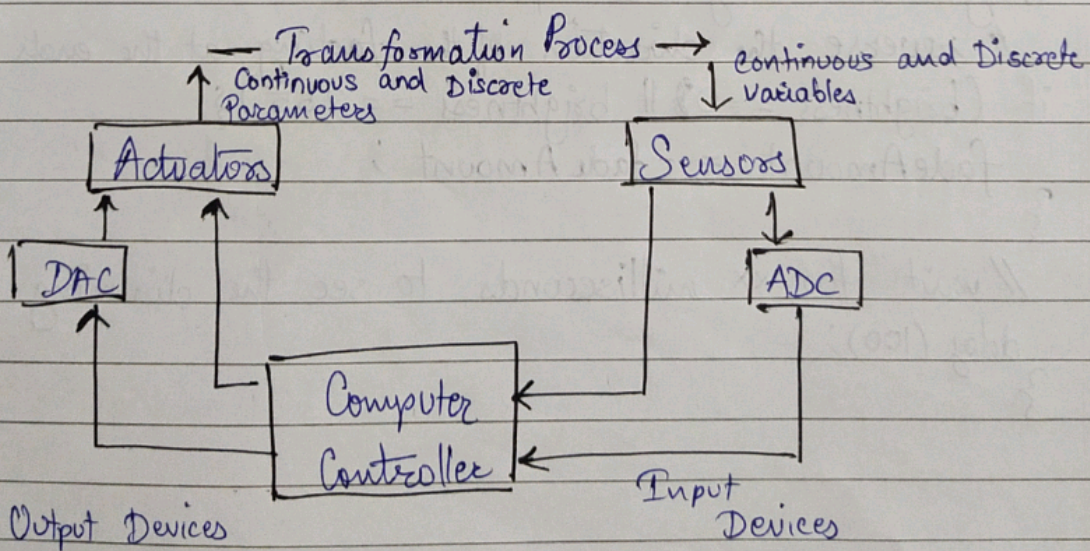
```
int led = 9;  // the pin that the LED is attached to
int brightness = 0;  // how bright the LED is
int fadeAmount = 3;  // how many points to fade the LED by

// the setup routine runs once when you press reset:
void setup() {
  // declare pin 9 to be an output:
  pinMode(led, OUTPUT);
}
// the loop routine runs over and over again forever:
void loop() {
  // set the brightness of pin 9:
  analogWrite(led, brightness);
  // change the brightness for next time through the loop:
  brightness = brightness + fadeAmount;
  // reverse the direction of the fading at the ends of the fade:
  if (brightness == 0 || brightness == 255) {
    fadeAmount = -fadeAmount;
  }
  // wait for xx milliseconds to see the dimming effect
  delay(100);
}
```

★ Sensor - Actuators :-

Sensor	Control Center	Actuator
Temperature sensor detects heat.	→ Sends this detect signal to the control center	→ Control center sends command to sprinkler	→ Sprinkler turns on and puts out flame.

Sensor to Actuator Flow →

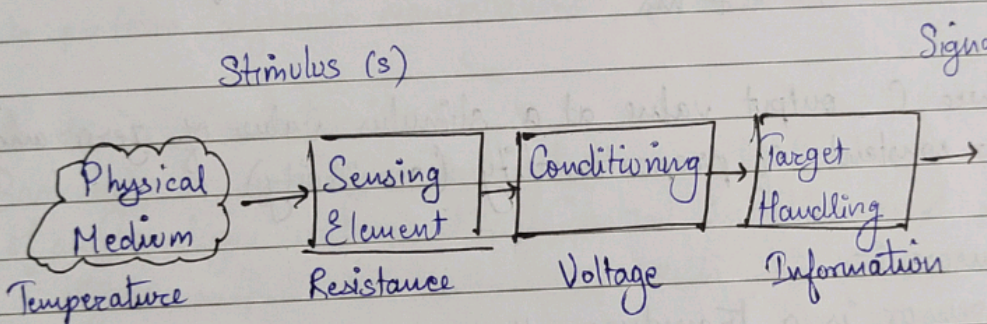
★ Computer Process Control System



✯ Computer - Process Interface

- To implement process control, the computer must collect data from and transmit signals to the production process.
- Components required to implement the interface:
  - Sensors to measure continuous and discrete process variables.
  - Actuators to drive continuous and discrete process parameters.
  - Devices for ADC and DAC
  - I/O devices for discrete data.

✯ Sensors



✯ Example word Problem :-

- The output voltage of a particular thermocouple sensor is registered to be 42.3 mV at temperature 105°C. It had previously been set to emit a zero voltage at 0°C. Since an output/input relationship exists between the two temperatures, determine (1) the transfer function of the thermocouple, and (2) the temperature corresponding to a voltage output of 15.8 mV.

$S = C + ms$

$42.3mV = 0 + m(105°C) = m(105°C)$ or $m = 0.402857/429$

$S = 0.4(s)$

$15.8mV = 0.4(s)$

$15.8/0.4 = s$

$S = 39.22°C$

★ Transfer Function :-

$$S = f(s)$$

Where S = output signal ; s = stimulus ; and ~~f(s)~~
f(s) = functional relationship
for binary sensors =
S = 1 if s > 0 and S = 0 if ≤ 0.

The ideal functional form for an analogue measuring device is a simple proportional relationship, such as :

$$S = C + ms$$

where C = output value at a stimulus value of zero and
m = constant of proportionality (sensitivity).

★ Sensors :-

- A sensor is a transducer that converts a physical stimulus from one form into a more useful form to measure the stimulus.

- Two basic categories :
  1. Analog
  2. Discrete
     - Binary
     - Digital (eg pulse counter)

- Examples :- Ultrasonic (distance)
  - Sound (db pressure)
  - Touch
  - Light (light intensity)

☆ Other Sensors

- Temperature
- RFID
- Barcode
- Proximity
- Vision
- Gyroscope
- Compass
- Tilt / Acceleration Etc

★ Data Converter Circuit :-

- Because digital integrated circuits are economical and accurate, it is convenient to process signals in digital form, for example, to perform algebraic manipulations, to transmit or store signals.

★ A to D Converter :-

- Basically an analogue to digital converter takes a snapshot of an analogue voltage at one instant in time and produces a digital output code which represents this analogue voltage.
- The number of binary digits, or bits used to represent this analogue voltage value depends on the resolution of an A/D converter
- For example, a 4-bit ADC will ~~the~~ have a resolution of one part in 15, $(2^4 - 1)$ whereas an 8-bit ADC will have a resolution of one part in 255, $(2^8 - 1)$. Thus an analogue to digital converter takes an unknown continuous analogue signal and converts it into an "n" - bit binary number of 2^n bits.
- Parallel or "Flash" A/D converters use a serial of interconnected but equally spaced comparators and voltage references generated by a series network of precision resistors for generating an equivalent

output code for a particular n-bit resolution.

(V_{IN} < V_{REF}) the output is "OFF"

(V_{IN} > V_{REF}) the output will be ON.

★ Analog read and write.

★ Arduino 5 (Analog read and write)
Program :-

```
#define LED_PIN 11
#define POTENTIOMETER_PIN A1
  void setup()
  {
    pinMode (LED_PIN, OUTPUT);
  }
  void loop()
  {
    int potentiometerValue = analogRead (POTENTIOMETER_PIN);
    int brightness = potentiometerValue
  analogWrite (LED_PIN, brightness);
}
```

04 March 2022 (2)

★ Analog Voltage Control

- Many actuators needs an analog voltage control.
- DC motors speed controlled by voltage.
- LED brightness can be controlled by voltage.
- Arduino cannot generate analog outputs. A digital to Analog converter can be externally connected to it.
- Other option is with PWM.

★ Pulse Width Modulation:-

- PWM, is a program for obtaining analog results in digital ways.
- Digital control is used to create a square wave, a modified signal between opening and closing.
- This closing pattern can mimic voltages between full openings (5 Volts) and close (Volts 0) by changing the time interval. The signal expires compared to the time ~~the~~ the signal expires.
- The duration of "on time" is called the pulse width. To get varying analog values, you change, or modulate, that pulse width. If you repeat this on-off pattern fast enough with an LED for example, the result is as if the signal is a steady voltage between 0 and 5v controlling the brightness of the LED.

★ analogWrite()

- Write an analog value (PWM wave) to a pin.
- Can be used to light a LED at varying brightness or drive a motor at various speeds.
- After a call to analogWrite(), the pin will generate a steady rectangular wave of the specified duty cycle until the next

call to analogWrite() (or a call to digitalRead() or digitalWrite() on the same pin.

★ Serial Communication Protocols :-

- Communication :- exchange of data between two microcontrollers in the form of bits.
- This exchange of data bits in microcontroller is done by some set of defined rules known as communication protocols.
- Data is sent bit after or in series i.e. one after the other then the communication protocol is known as Serial Communication Protocol.

★ Serial Buses - ES

- UART serial bus for sending debug messages to your development host.
- I2C serial bus for communicating with sensors (e.g. the accelerometer).
- SPI serial bus for communicating with the Bluetooth Low Energy radio.

★ Protocol :-

- Each character is sent as
  - a logic low start bit
  - a configureable number of data bits (usually 7 or 8, sometimes 5)
  - an optional parity bit
  - one or more logic high stop bits
  - with a particular bit timing ("baud")

☆ Serial Bus Interface Motivations

- Motivation
  - Without using a lot of I/O lines
    - I/O lines require I/O pads which cost $$$ and size.
    - I/O lines require PCB area which costs $$$ and size.
  - Connect different systems together
    - Two embedded systems
    - A desktop and an embedded system
  - Connect different chips together in the same embedded system
    - MCU to peripheral
    - MCU to MCU
  - Often at relatively low data rates
  - But sometimes at higher data rates.
- So, what are our options?
  - Universal Synchronous/Asynchronous Receiver Transmitter
  - Also known as USART

☆ Serial Bus ~~Design Space~~ Examples :-

	S/A	Type	Duplex	#Devices	Speed (kbps)	Distance (ft)	Wires
RS232	A	Peer	full	2	20	30	2+
RS422	A	Multidrop	Half	10	10000	4000	1+
RS485	A	Multipoint	Half	32	10000	4000	2
I2C	S	Multimaster	Half	?	3400	<10	2
SPI	S	Multimaster	Full	?	>1000	<10	3+
Microwave	S	Master/slave	full	?	>625	<10	3+
1-Wire	A	Master/slave	Half	?	6	1000	1+

★ UART Uses
- Used to be commonly ~~use~~ used for internet access.

★ UART
- Universal Asynchronous Receiver/Transmitter.
- Hardware that translates between parallel and serial forms.
- Commonly used in conjunction with communications standards such as EIA, RS-232, RS-422, or RS-485.

★ Arduino Serial :-
- We can use the built-in Arduino serial library to communicate with the hardware serial ports.
- Serial libraries simplify the use of the serial ports by insulting you from hardware complexities.
- Serial Message Protocol the hardware or software serial ~~library~~ libraries handle sending and receiving information.
- This information often consists of groups of variables that need to be sent together.
- For the information to be ~~interper~~ interpreted correctly, the receiving side needs to recognize where each message begins and ends.
- Meaningful serial communication, or any kind of machine-to-machine communication, can only be achieved if the sending and receiving sides fully agree how information is organized in the message.
- The formal organization of information in a message and the range of appropriate responses to requests is called a communications protocol.
- To print text and numbers from your sketch, put the Serial.begin(9600) statement in setup(), and then use Serial.print()

statements to print the text and values you want to see.

- The Arduino Serial Monitor function can display serial data sent from Arduino.

Code for Arduino Serial :-

```
void setup()
{
   Serial.begin(9600) ;  //send and receive at 9600 baud
}
int number = 0;
void loop()
{
Serial.print ("The number is =");
Serial.println(number);  // print the number
delay (500);  //delay half second between numbers
number ++ ;  // to the next number
}
```

★ 14th March 2022 :-

★ How to use interrupts on UNO :-

- Interrupts are useful for making things happen automatically in microcontroller programs, and can help solve timing problems.
- Most processors have interrupts
- Interrupts let you respond to "external" events while doing something else
- Most Arduino designs have two hardware interfaces (referred to such as "interruption 0" and "interruption 1") heavily connected to digital I/O pins 2 and 3 respectively.

- We can describe the process using a special function called "Service Cycle Disruption" (commonly known as ISR).
- You can define a schedule and specify conditions on the ~~disrupt~~ ~~edge~~, rising edge, falling edge or both. At these specific conditions, the interrupt would be serviced.
- It is possible to have that function executed automatically, each time an event happens on an input pin.

★ About Interrupt Service Routines :-

- Generally an ISR should be as short and fast as possible.
- If your sketch uses multiple ISRs, only one can run at a time, other interrupts will be executed after the current one finishes in an order that depends on the priority they have.
- millis() relies on interrupts to count, so it will never increment inside an ISR.
- Since delay() requires interrupts to work, it will not work if called inside an ISR.
- To make sure variables shared between an ISR and the main program are updated correctly, declare them as volatile

★ How to use Interrupts on UNO :- (continue)

- Syntax

  attachInterrupt(digitalPinToInterrupt(pin), ISR, mode); (recommended for UNO) Parameters.

- interrupt : the number of the interrupt (int)

  pin : the pin number

- ISR: the ISR to call when the interrupt occurs; this function must take no parameters and return nothing. This function is sometimes reffered to as an interrupt service routine.
- mode: defines when the interrupt should be triggered. Four constants are predefined as valid values:-
- LOW, CHANGE, RISING, FALLING.

Program:-

```
const byte LED = 13;
const byte BUTTON = 2;
// Interrupt Service Routine (ISR)
void switchPressed()
{
    if (digitalRead (BUTTON) == HIGH)
     digitalWrite (LED, HIGH);
   else
     digitalWrite (LED, LOW);
}    // end of switchPressed
void setup()
{
  pinMode (LED, OUTPUT); // so we can update the LED
  digitalWrite (BUTTON, HIGH); // internal pull-up resistor
  attachInterrupt (digitalPinToInterrupt (BUTTON), switchPressed,
                   CHANGE); // attach interrupt handler
} // end of setup
void loop()
{
    // loop doing nothing
}
```

24th March 2022

IOT Networking and Communication Technologies :- (PPT Name)

Properties of IOT

- Self-adaptation
- Self-organization
- Self-optimization
- Self-configuration
- Self-protection
- Self-healing
- Self-description
- Self-discovery
- Self-matchmaking
- Self-energy supplying

Key Technologies for IOT :-

- Security Privacy
- Knowledge Aggregation
- Future Internet
- Standards
- Sensor Networks
- Cloud Computing
- Communications
- Discovery Services
- Embedded Systems
- Nanoelectronics
- Software
- System Integration

Functional Components of IOT :-

- Component for interaction and communication with other IOT devices
- Component for processing and analysis of operations
- Component for internet interaction
- Components for handling Web services of applications
- Component to integrate application services
- User interface to access IOT.

IOT Components :-

- Device (the thing)
- Local Network
- Internet
- Backend Services
- Applications

★ An Example IoT Implementation

Strategic Research and Innovation Agenda (SRIA) intends time frame the number of connected devices, their features, their distribution and implied communication requirements it includes :
- Networking Technology
- Communication Technology.

★ Edge Computing :- (100% Plagarism)

- In edge computing, critical data processing occurs at the data source rather than in a centralized cloud-based location. Other items sometimes used to describe edge computing include 'fog' computing and grid computing.

★ Edge Computing : Drivers Latency :-

- Data processing close to ~~write~~ where it originates avoids round-trip time to the cloud Bandwidth.
- Optimization of communication to and from the cloud Privacy / Security (Plagarism Removed)
- Processing data near the source avoids the time to and from the bandwidth of the cloud.
- Communication improvements to and from the cloud Privacy/security.
- Sensitive data is always connected locally.
- Continuous processing (in some cases) despite a lack of ~~cut~~ cloud connectivity dependency.

- Data processing is close to end user interaction points and other components of the system.

★ Edge Computing (0% Plagarism) :-

- At the computer end, important data processing takes place in the data source rather than in the middle of the cloud. Other words sometimes used to describe edge computing includes 'fog' computing and grid computing.
- With typical network cloud systems, data is ~~paused~~ pushed up to the server and then downloaded by customers. This works well for videos, photos, music, documents and non-essential applications, but it is not good design for real time IIot data
- If the data does not remain in the source, the quality of the data may be unreliable. Issues with newness, data accuracy, and delivery speed may be affected when using cloud architecture.
- When supporting a sophisticated IIOT solution, centralized cloud computing becomes costly in terms of bandwidth and computer resources. Problems of latency & network availability also have an impact of IIOT clouds.
- Edge computing and cloud computing are interconnected structures that combine to create powerful IIOT platforms. One does not change the other.

★ Use cases :-

- Telco :- Opportunity for providing edge computing devices in existing infrastructure.
- Eg: micro data centers at the base of cellular towers
-

- Examples
  - Azure IOT Edge - deploy business logic to edge devices and monitor from the cloud.
  - Amazon - AWS CloudFront - CDN Service includes Lambda@Edge - AWS Green grass - connected IOT devices can run AWS Lambda functions and other code on locally - connected data.

★ Edge and Cloud :-

Edge Computing	Cloud Computing
• Basic data visualization	• Complex analytics
• Basic data analytics and short term data historian features	• Big data mining
• Data caching, buffering and streaming	• Sources of business logic
• Data pre-processing, cleansing, filtering and optimization	• Machine learning rules
• Some data aggregation	• Advanced visualizations
• Device to Device communications/M2M	• Long term data storage/ ~~work~~ warehousing

25th March 2022 :-

★ Machine to Machine (M2M) :-

- Machine - to - Machine (M2M) refers to the connection of machines (devices) for the purpose of monitoring and remote control and data exchange.
- The M2M local network includes equipment (or M2M nodes) embedded hardware, audio, and communication hardware module.

- Different communication systems can be used for local M2M networks such as Zigbee, Bluetooth, Modbus, M-Bus, Wireless M-Bus, Power Line Communication (PLC), 6LoWPAN, IEEE 802.15.4, etc.
- The communication network provides connectivity to remote M2M local networks.
- The network may use wireless or wireless (IP-based) networks.
- While local M2M networks use proprietary or non-IP-based communication ~~agreements~~, the ~~network~~ network uses IP-based networks.

★ M2M Gateway :-

- Since non-IP protocols are used within local M2M networks, M2M nodes within a single network cannot communicate with nodes on an external network.
- To enable connectivity between remote M2M networks, M2M gates are used.

★ Difference between IOT and M2M :-

- **Communication Protocols :-**
  - M2M and IOT may differ in the way that communication between devices or devices occurs.
  - M2M uses proprietary or non-IP communication protocols for communications between local M2M networks.

- **Machines in M2M vs Things in IOT**
  - "Objects" in IOT refers to tangible objects that have distinct identities and are able to sense & and communicate their external environment (and applications) or their internal physical conditions.

- M2M systems, unlike IOT, usually have the same type of equipment within the local M2M network.

- Hardware vs Software Emphasis :-
  - While M2M emphasis is more than hardware with embedded modules, IoT emphasis is more than software.

- Data Collection and Analysis :-
  - M2M data is collected in point solutions and usually in local storage infrastructure
  - Unlike M2M, data on IoT is collected in the cloud (either public, private or mixed)

- Apps :-
  - M2M data is collected on point solutions and can be accessed through in-app applications such as diagnostic applications, service management applications, and in-house business applications.
  - IOT data is collected in the cloud and can be accessed through cloud applications such as analytics applications, business applications, remote diagnostics and management applications, etc.

★ M2M Architectures - Three domains :-

- M2M Device Domain
  - Gateway Connectivity Interface (Communication and Processing Units) and Edge Computing (data element analysis and transformation)

- M2M network :-
  - M2M server, device ownership, device management, and device network, Data Analysis, Summary, Collection, Management, uni-cast deployment and multicast messaging and critical monitoring functionality.
  - M2M devices Domain Communication Gateway Connectivity Interface (Edge Computing Units) and Edge Computing (data element analysis and conversion)
  - Portable Devices and Controls (IoT objects) [Sensors, Devices Intelligent Edge nodes for Different Types of Communication (Communication and Processing Units).

- M2M application :-
  - Integration, Collaboration and M2M Application Services Application (Reporting, Analysis, control).

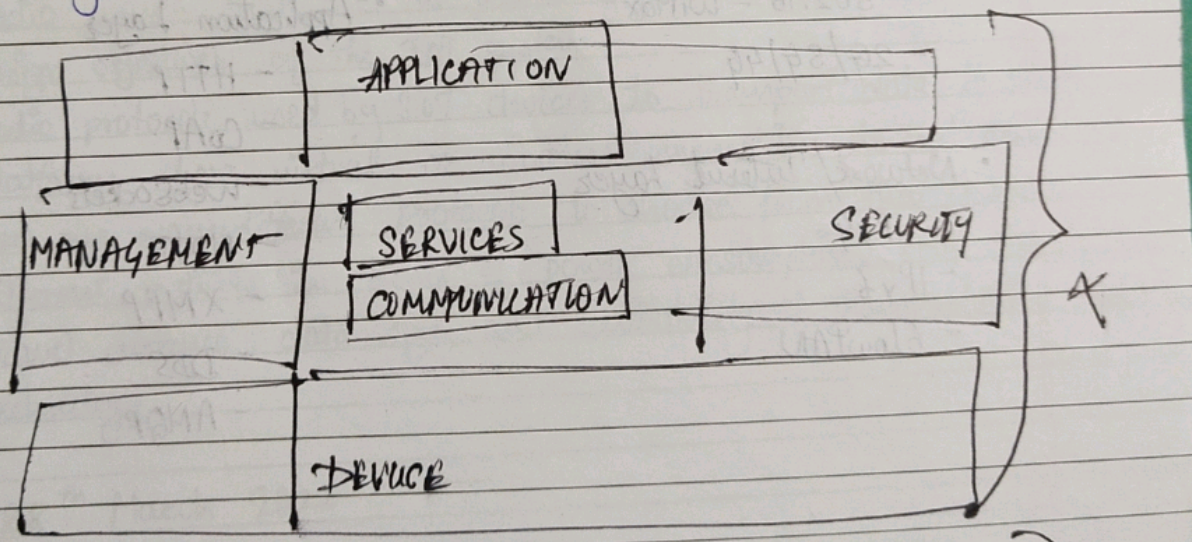
★ M2M Applications :-
- Connected Cars for Safety and Infotainment
- Remote Monitoring
- ATMs/Point of Sales Terminal Connected for centralized Security
- Remote Monitoring, Truck Fleet Management
- Food packaging
- Shop floor operations
- Sequential operations

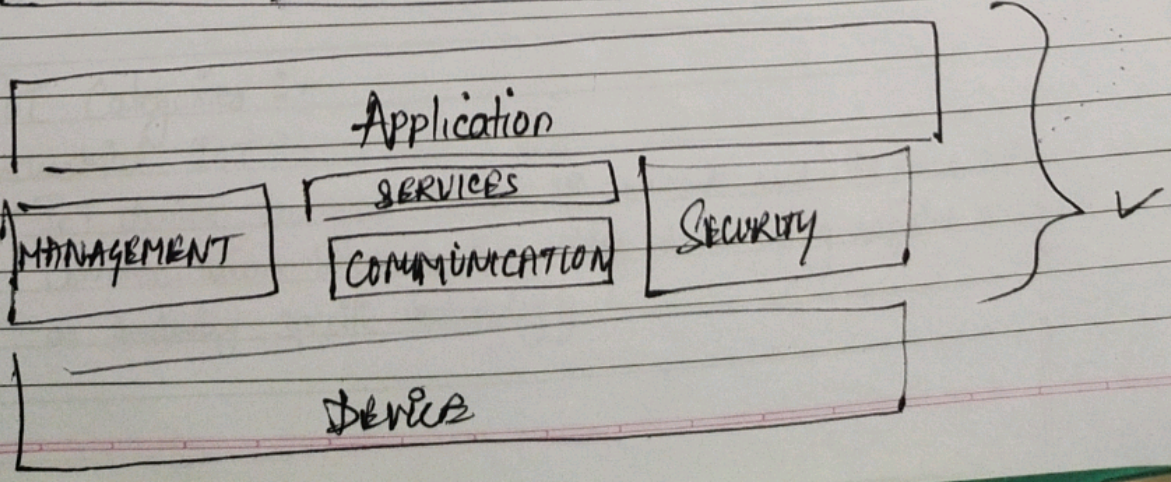
★ Physical Design of IOT :-
- IoT object refers to IoT devices with different identifices as well capable of remote sensing, operation and monitoring capabilities
-

- IoT device can :-
  - Exchange data with other connected devices and applications (directly or indirectly)
  - Collect data from other devices and process data locally
  - Send data to central servers or cloud applications for data processing.
  - Perform other functions locally and others within IoT infrastructure based on time and location.

★ Logical Design of IoT

- Abstract representation
- No low level specs
- Functional blocks for ~~eg~~ capabilities
- Sensing, identification, actuation, communication and management.





PAGE No. 58
DATE / /

★ Generic Block Diagram of IOT

- I/O interface for sensors
- Internet connectivity
- Memory and storage
- Audio / Video interface

★ IOT Protocols / Middleware

IOT Protocols

- Link Layer
  - 802.3 - Ethernet
  - 802.11 - Wifi
  - 802.15.4 - LR-WPAN
  - 802.16 - WiMax
  - 2G/3G/4G
- Network / Internet Layer
  - IPv4
  - IPv6
  - 6LowPAN
- Transport Layer
  - TCP
  - UDP
- Application Layer
  - HTTP
  - CoAP
  - WebSockets
  - MQTT
  - XMPP
  - DDS
  - AMQP

25 March 2022 (2)

★ Radio Communication :-

- It is the transmission of signals through the fluctuations of electrical currents with lower frequencies than visible light.
- Electromagnetic radiation travels through circulating electromagnetic fields passing through air and space vacuum.
- Information is transmitted by systematically changing (modelling) other material of broadcast waves such as amplitude, frequency, or phase.
- IoT Wireless Technology in its simplest form requires a radio transmitter capable of receiving and transmitting radio waves as a means of transmitting signals or data.
- A radio wave is an electric frequency used for long-distance communication. IoT solutions rely on IoT wireless technology, radio waves that use the power of compliance based on the design objectives of the IoT system.
- Radio protocols used by IoT devices to transport data to cloud platforms where virtual or wireless connections do not exist. There are many different protocols to choose from that have different features in terms of power consumption, body size, travel distance, data size, and availability of transportation technology.

28th March 2022

★ IoT Categories :-

- Industrial IoT :-
  - IoT device connects to an IP network and the globe internet.
  - Communication between the nodes done using regular as well as industry specific technology.

- Consumer IoT :-
  - IoT device communication within the locally networked devices.
  - Local communication is done mainly via Bluetooth, Zigbee or Wifi.
  - Generally limited to local communication by a Gateway.

★ Technical Deviation from Regular Waves :-

IoT Stack
- Application Layer
  - Application, Management, Binary, JSON, CBOR, MQTT, GAP, XMPP, AMQP
- Transport Layer
  - UDP, DTLS
- Internet Layer
  - IPV6, 6LOWPAN
- Network Interface
  - IEEE 802.15.4 MAC, IEEE 802.15.4 PHY/Radio

Web Stack
- Application Layer
  - HTML, Web Applications, XML, JSON, HTTP, DHCP, DNS, TLS/SSL
- Transport Layer
  - TCP, UDP
- Internet Layer
  - IPv6, IPv4, IPSec
- Network Interface
  - Ethernet, DSL, ISDN, Wireless LAN, Wi-fi.

☆ IoT challenges :-

- Security
- Scalability
- Energy ~~Effen~~ Efficiency
- Bandwidth management
- Modeling and Analysis
- Interfacing
- ~~Interp~~ Interoperability
- Data Storage
- Data Analytics
- Complexity management (eg SDN)

☆ Complexity of Network :-

- Growth of networks
- Interference among devices
- Network management
- Heterogeneity in networks
- Protocol standardization within networks.

☆ Wireless Networks

- Traffic and load management
- Variations in wireless networks - Wireless Body Area Networks and other Personal Area Networks.
- Interoperability
- Network management
- Overlay networks

★ Scalability

- Flexibility within Internet
- IoT integration
- Large scale deployment
- Real-time connectivity of billions of devices.

☆ Wireless and Mobile Network (Topic)

☆ Elements of a wireless network :-

1. Wireless hosts :-
   - laptop, smartphone, IoT
   - run applications
   - may be stationary (non-mobile) or mobile
     - wireless does not always mean mobility!)

2. Base Station :-
   - typically connected to wired network
   - relay - responsible for sending packets between wired network and wireless host(s) in its "area"
   - e.g., cell towers, 802.11 access points

3. Wireless Link :-
   - typically used to connect mobile(s) to base stations, also used as backbone link
   - multiple access protocol coordinates link access
   - various transmission rates and distances, frequency bands.

4. Infrastructure mode :-
   - base station connects mobiles into wired network
   - handoff: mobile changes base station providing connection into wired network.

5. ad hoc mode :
   - no base stations
   - nodes can be only transmit to other nodes within link coverage
   - nodes organize themselves into a network: route among themselves

☆ Wireless network taxonomy :-

	single hop	multiple hop
infrastructure (e.g. APs)	host connects to base station (wifi, cellular) which connects to larger Internet	host may have to relay through several wireless nodes to connect to larger Internet : mesh net
no infrastructure	no base station, no connection to larger Internet (Bluetooth, ad hoc nets)	no base station, no connection to larger Internet, May have to relay to reach other a given wireless node MANET, VANET

☆ IoT Technical Solutions

- fixed & Short Range
  - RFID
  - Bluetooth
  - Zigbee
  - Wifi
  - ....
- Long Range Technologies
  - Non 3GPP Standards (LPWAN)
  - 3GPP standards

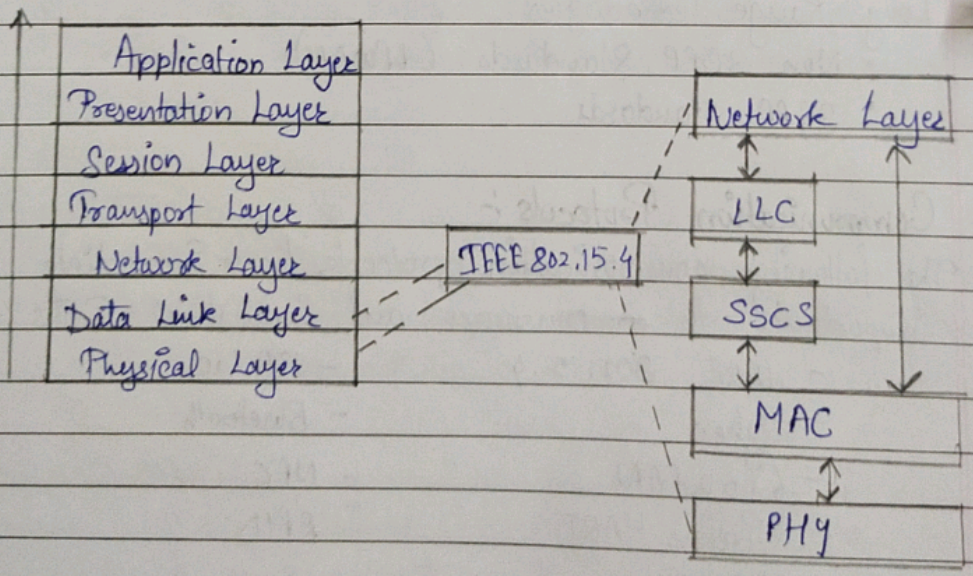
☆ Communication Protocols :-

- The following communication protocols have immediate importance to consumer and industrial IoTs :-
  - IEEE 802.15.4
  - Zigbee
  - 6LoWPAN
  - Wireless HART
  - Z-Wave
  - ISA 100
  - Bluetooth
  - NFC
  - RFID

- ✓ Connectivity (6LoWPAN, RPL)
- ✓ Identification (EPC, uCode, IPv6, URIs)
- ✓ Communication/Transport (Wifi, Bluetooth, LPWAN)
- ✓ Discovery (Physical Web, mDNS, DNS-SD)
- ✓ Data Protocols (MQTT, CoAP, AMQP, Websocket, Node)
- ✓ Device Management (TR-069, OMA-DM)
- ✓ Semantic (JSON-LD, Web Thing Model)
- ✓ Multi-layer-Frameworks (Alljoyn, IoTivity, Weave, Homekit)

★ Features of IEEE 802.15.4

- Well-known standard for low data-rate WPAN.
- It is designed to ~~use~~ monitor and control applications with low data quality and extended use of low power life.
- This standard uses only the first two layers (PHY, MAC) and logical link control (LLC) and add-ons service specific convergence sub-layer (SSCS) to communicate with all the top layers.
- Works on ISM band.



- ✓ Uses direct sequence spread spectrum (DSSS) flexibility.
- ✓ It is very tolerant of noise and distortion and provides ways to improve link reliability.
- ✓ Low-speed versionds use Binary Phase Shift Keying (BPSK)
- ✓ High-level versions of data use offset-quadrature phase-shift keying (O-QPSK)
- ✓ It useo multiple carrier access to prevent conflict (CSMA-CA) to access the channel.
- ✓ Multiplexing allows multiple users or nodes to access without interruption on the same channel at different times.

- ✓ Energy consumption is reduced due to the rare occurrence of very short packets with a low activity cycle (< 1%)
- ✓ The minimum rated power is 3dBm or 0.5 mW.
- ✓ Transmission, in most cases, is a Line of Visibility (LOS)
- ✓ The standard transmission distance varies between 10 m and 75 m
- ✓ The best distance of acquired external appeal can be up to 1000m.
- ✓ Defined network topologies --- Star, and Mesh.

☆ Full Function Device (FFD)
- Can talk to all types of devices
- Supports full protocols

☆ Reduced Function Device (RFD)
- Can only talk to an FFD
- ~~Supports full protocol~~ Lower Power Consumption
- Minimal CPU/RAM required

30th March 2022 :-

Zigbee :-

- International standard IEEE 802.15.4
- Developed by Zigbee Alliance [1]
- Industry, Science, Medicine, and IoT
- The physical distance is 10 to 100 meters
- Zigbee uses IEEE's 802.15.4 local area network to communicate with other Zigbee devices.
- These can speak upto 300+ meters wide with a clear line of sight, operating between 75-100 meters inside the house.
- Zigbee is targeted at radio-frequency (RF) applications which require a low data rate, long battery life, and secure networking.
- Zigbee protocol features include :-
  - Support of for multiple network topics such as point-to-point, point-to-multipoint and spaced networks
  - Low performance cycle - gives longer battery life
  - Low delay
  - Upto 65,000 nodes per network
  - 128 bit AES encryption for secure data communications
  - Avoiding ~~collcat~~ collisions, trying and thanking.

★ Features of Zigbee :-

- Extensive use of IEEE 802.15.4.
- Zigbee protocol is defined as layer 3 and above. Works 802.15.4 layers 1 and 2.
- The standard uses layers 3 and 4 and above. Works 802.15.4 layers 1 a
- The standard uses layers 3 and 4 to describe additional communication enhancements.
- These enhancements include authentication by valid nodes, encryption for security, and data channel strength and transmission capability that allows mesh networking.
- The most popular use of Zigbee wireless nerve networks is using mesh topology.

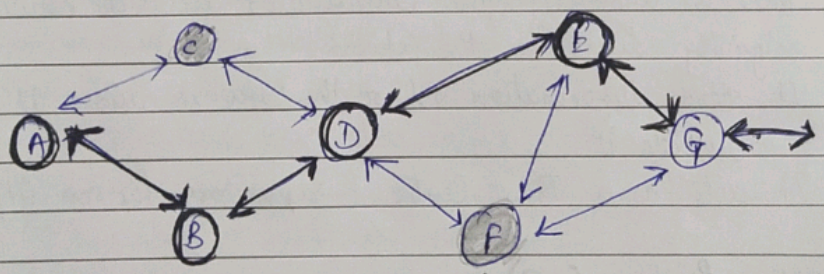
★ ZigBee Types :-

1. ZigBee Coordinator (ZC) :-
   - The Coordinator forms the root of the Zigbee network tree and might act as a bridge between networks
   - There is a single ZigBee Coordinator in each network, which originally initiates the network.
   - It stores information about the network under it and outside it.
   - It acts as a Trust Center & repository for security keys

2. Zigbee Router (ZR) :-
   - Capable of running applications, as well as relaying information between nodes connected to it.

3. Zigbee End Device (ZED) :-
- It contains just enough functionality to talk to the parent node, and it cannot relay data from other devices.
- This allows the node to be asleep a significant amount of the time thereby enhancing battery life.
- Memory requirements and cost of ZEDs are quite low, as compared to ZR or ZC.

★ Zigbee Mesh :-
- In a mesh, any node can communicate with any other node within its range
- If nodes are not in range, messages are relayed through intermediate nodes.
- This allows the network deployment over large areas.
- Meshes have increased network reliability.
- For ex, if nodes C and F are down, the message packets from A can still be relayed to G via nodes and E.
- Zigbee mesh networks are self-configuring and self-healing



Transcievers
Coordinator
Faulty

Mesh networks enable high levels of reliability and longer coverage range by providing more than one path through the network for any wireless link.

✱ Zigbee Network Layer :-

- The network layer uses the Ad Hoc On-Demand Distance Vector (AODV) route.
- To secure a place to stay, AODV distributes the route request to all its immediate neighbors.
- Neighbors passed the same information to their neighbours, eventually spreading the request across the network.
- When location is detected, its calculates the lowest cost and is notified to the requesting device by unicast message.

01st April 2021:-

★ RFID :- Radio Frequency Identification

- Appeared in 1945.
- Features :-
  - Identify objects, record metadata, or control individual target.
  - More complex devices (e.g., reader, interrogators, beacons) usually connected to a host computer or network.
  - Radio frequencies from 100 kHz to 10 GHz

- Operations :-
  - Reading Device called Reader (connected to backend network and communicates with tags using RF)
  - One or more ~~tags~~ tags (embedded antenna connected to chip based and attached to object).

(What is RFID?) ↓ / main : How does RFID works ↓

- RFID is an acronym for "radio-frequency identification".
- Data digitally encoded in RFID tags, which can be read by a reader.
- Somewhat similar to barcodes.
- Data read from tags are stored in a database by the reader.
- As compared to traditional barcodes and QR codes, RFID tag data can be read outside the line-of-sight

★ RFID Features :-

- RFID tag consists of an integrated circuit and an antenna.
- The tag is covered by a protective material which also acts as a shield against various environmental effects.
- Tags may be passive or active.
- Passive RFID tags are the most widely used.
- Passive tags have to be powered by a reader inductively

before they can transmit information, whereas active tags have their own power supply.

☆ Working Principle :-

- Derived from Automatic Identification and Data Capture (AIDC) technology.
- AIDC performs object identification, object data collection and mapping of the collected data to computer systems with little or no human identification.
- AIDC uses wireless communication
- RFID uses radio waves to perform AIDC functions.
- The main component of an RFID system include an RFID tag or smart label, an RFID reader, and an antenna.

☆ Applications :-

- Inventory management
- Asset tracking
- Personnel tracking
- Controlling access to restricted areas
- ID badging
- Supply chain management
- Counterfeit prevention (e.g. in the pharmaceutical industry)

☆ (Project CHIP (ppt Network and communication IoT) and its application) (slide 110)

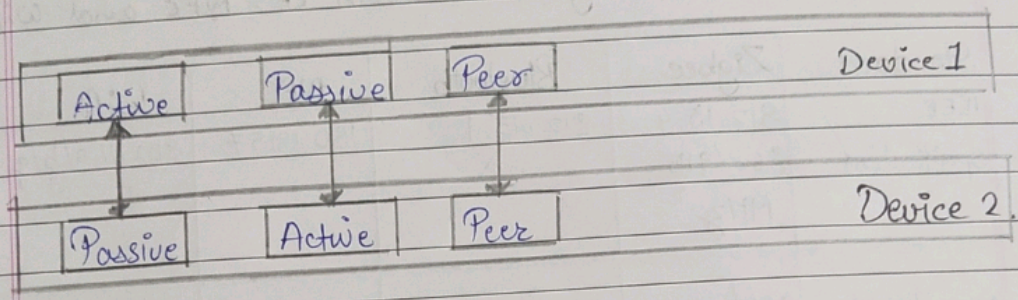
☆ Near Field Communication (NFC):-

- Near-field Communication (NFC) is a set of communication protocols for communication between two electronic devices over a distance of 4 cm (11/2 in) or less.
- It is a contact-less communication technology based on a radio frequency (RF) field using a base frequency of 13.56 MHz.
- The NFC platform was established in 2004, to improve the use of NFC technology. Provides NFC level.

- The main idea is to deploy electronic ticketing based on secure microcontrols (today called secure elements) similar to those used on SIM cards, and to supply them remotly with inductive integeration.
- Rf field generated by NFC forum device to communicate with NFC forum
- The marker has three function:-
  1. Transfer power from NFC forum Device to NFC field marker.
  2. The NFC device sends information to the NFC forum Tag by modelling the Rf platform signal (signal conversion).
  3. NFC device receives information from NFC Forum Tag by hearing the load change generated by the NFC forum (volume change).

- NFC tags are idle data stores that can be read, and under certain written terms, an NFC device. They usually contain data (from between 96 and 8,192 bytes) and are readable only for normal use, but may be rewritten.
- The Rf interface support data communications of 106, 212 and 424 kbps data as of today.

- NFC uses different flexibility schemes such as ASK (Amplitude Shift Keying) with varying depths of 100 or 10% fluctuations or load fluctuations
- Coding techniques such as NRZ-L (Non-return to zero), Manchester and Modified Miller code for data transfer.



Passive :-
- The startup ~~door~~ device provides the network company field and the target device, which acts as a transponder, communicates by modeling ~~the~~ the incident field. In this mode, the target device may pull its operating force to the magnetic field provided by the launcher.

Active :-
- Both the launcher and the target ~~door~~ device are connected by producing their fields seperately.
- The device stops transferring to receive data from another. This mode requires both the devices to install power supplies.

- Card emulation mode (passive mode) : the NFC device behaves like an existing contactless card conforming to one of the legacy standards.
- Peer-to-Peer mode : two NFC devices exchange information. The initiator device (polling ~~door~~ device) requires less power

compared to the reader/writer mode because the target (listner) uses its own power supply.

- Reader/writer mode (active mode): the NFC device is active and reads or writes to a passive legacy RFID tag.

★ Comparison of the Zigbee, Bluetooth LE, NFC and Wifi.

Standard	Zigbee	Bluetooth	NFC	Wifi
IEEE Specification	812.15.4 868/915 MHz	812.15.1	ISO 13157	802.11 a/b/g
Freq. Band	868/915 MHz 2.4GHz	2.4-2.5GHz	13.56 MHz	2.4 to 5 GHz
Max Signal Rate	250 kbps	305 kbps	424 kbps	54 Mbps
Nominal Range	10 m	~50m	~5cm	100 m
Network Type	WPAN	WPAN	P2P	WPAN/P2P
Power Consumption	40mA	12.5mA	50mA	116mA

★ 6LoWPAN

- Most simple embedded devices still use 8-bit and 16-bit controls with limited memory, as they have lower power, smaller and cheaper.
- At the same time, the portable commercialization of wireless technology has resulted in short, low-power wireless broadcasters with limited data rates, frame sizes and activity cycles similar to IEEE 802.15.4 standard.
- You need to connect a number of devices with online resources.

★ Introduction :-

- Low Power Personal Area Networks Wireless Over IPv6.
- Allows very small devices with limited processing capacity to transmit data wirelessly using an Internet Protocol.
- Allows low power devices to connect to the Internet.
- Created by Internet Engineering Task force (IETF) - RFC 5933 and RFC 4919.
- Compiled to RFC6282
- The 6LoWPAN concept originated from the idea that "the Internet Protocol could and should be applied even to the smallest devices, and ~~should be applied even to the~~ that low-power devices with limited processing capabilities should be able to participate in the Internet of Things.

- It works great with open IP standard including TCP, UDP, HTTP, CoAP, MQTT & web-sockets.
- It offers end to end IP addressable nodes. There's no need for a gateway, only a router which can connect the 6LOWPAN network to IP.

- It supports self-healing, robust and scalable mesh routing. Offers one-to-many & many-to-one routing.
- The 6LoWPAN mesh routers can route data to others nodes in the network. In a 6LoWPAN network, leaf nodes can sleep for a long duration of time.
- It also offers through support for the PHY layer which gives freedom of frequency band & physical layer, which can be used across multiple communication platforms like Ethernet, WiFi 802.15.4 or Sub-1GHz ISM with interoperability at the IP level.

★ IpV4 and IpV6

- Internet Protocol version 4, generally referred to as ~~IPv4~~ IPv4, was developed in the early 1980s.
- An IPv4 address comprises four numbers, each ranging from 0 to 255, which are separated by ~~pe~~ periods.
- The Domain name Service (DNS) came along, which translates numbers into names
- IPv4 has a theoretical limit of 4.3 billion addresses
- Developed to solve these capacity issues for good, IPv6 was needed when IPv4 could no longer support the load.
- IPv6 uses 128-bit addresses, allowing for a theoretical 340, 282, 366, 920, 938, 463, 463, 374, 607, 431, 768, 211, 456.
- The advent of IPv6 brought more functionality, in addition to more IP addresses, security.
- IPv4 networks are mature and thus highly optimized, more so than IPv6 networks. So with time and tuning, IPv6 networks will get faster.
- IPv6 is designed for end-to-end encryption, so in theory, widespread adoption of IPv6 will make man-in-the-middle attacks significantly more difficult.

- The Internet Society reported last year that there are 24 countries in the world where IPv6 totals more than 1% of overall IP traffic, and 49 that have topped the 5% threshold. So migration from IPv4 to IPv6 is progressing very slowly.

IPv4	IPv6
• deployed 1981	deployed 1998
• 32-bit IP address	128-bit IP addresses
• 4.3 billion addresses	7.9×10^{28} addresses
Addresses must be reused and masked	Every device can have a unique address
• Numeric dot-decimal notation	Alphanumeric hexadecimal notation
	50b2:6400:0000:0000:6c3a:b17d:0000:10a9.
	~~(Simplified : 50b2:6400:630a:b17d:0:10~~
	(Simplified : 50b2:6400:6c3a:b17d:0:10a9)
• DHCP or manual configuration	Supports autoconfiguration

★ Features of 6LoWPANs :-

- Allows IEEE 802.15.4 radios to carry 128-bit addresses of Internet Protocol version 6 (IPv6)
- Header compression and address translation techniques allow the IEEE 802.15.4 radios to access the internet.
- IPv6 packets compressed and reformatted to fit the IEEE 802.15.4 packet format.
- 6LoWPAN stack comprises of IPv6 and UDP modules at Network and Transport layers respectively and uses IEEE 802.15.4 MAC for carrying IPv6 packets.
- Uses include IoT, Smart Grid, and M2M applications

- 6LowPAN uses an adaptation layer between the network (IPv6) and data link layer (IEEE 802.15.4 MAC) to fragment and reassemble IPv6 packets.

★ Addressing in 6LOWPAN :-

- IPv6 nodes are assigned 128 bit IP addresses in a hierarchical manner, through an arbitary length network prefix.
- IEEE 802.15.4 devices may use either of IEEE 64 bit extended addresses or, after an association event, 16 bit addresses that are unique within a PAN.
- 64 bit address : globally unique
- 16 bit addresses : PAN specific; assigned by PAN coordinator.

★ Header Type : Dispatch
- Dispatch : Initiates communication
- 0,1 : Identifier for Dispatch Type
- Dispatch
  - 6 bits
  - Identifies the next header type
- Type Specific Header :-
  - Determined by Dispatch header

★ Header Type : Mesh Addressing
- 1,0 : ID for Mesh Addressing Header
- V : '0' if originator is 64-bit extended address, '1' if 16-bit address
- F : '0' if destination is 64-bit addr ; '1' if 16 bit addr.
- Hops Left : decreamented by each node before sending to next hop

☆ 6LoWPAN Architecture :-

- A LoWPAN consists of nodes, which may play the role of host or router, along with one or more edge routers.
- Neighbor Discovery defines how hosts and routers interact with each other on the same link.
- LoWPAN nodes may participate more than one LoWPAN at the same time (called multi-homing), and fault tolerance can be achieved between edge routers.
- LoWPAN Nodes are free to move throughout the LoWPAN, between edge routers, and even between LoWPANs.
- A multihop mesh topology within the LoWPAN is achieved either through link-layer forwarding (called Mesh-under) or using IP routing (called Route-Over). Both techniques are supported by 6LoWPAN.

- The network interfaces of the nodes in a LoWPAN share the same IPv6 prefix which is distributed by the edge router and routers throughout the LoWPAN.
- In order to facilitate efficient network operations, nodes register with an edge router. These operations are part of Neighbor Discovery (ND), which is an important basic mechanism of IPv6.

☆ 6LoWPAN Considerations :-

- ✓ Mesh routing within the PAN space
- ✓ Routing between IPv6 and the PAN domain
- ✓ Routing protocols in use : ~~Routing~~, LOADng, RPL

✱ Node MCU

- The development baard equips the ESP-12E module which contains the ESP8266 chip with a Tensilica, Xtensa 32 bit LX106 RISC microprocessor operating at 80 to 160 MHz adjustable clock frequency and supports RTOS.

✱ ESP 8266 :-

- The ESP8266 module is an IoT device that ~~xxxxx~~ incorporates a 32-bit ARM microprocessor supported by Wifi network and built-in flash memory.
- This structure allows it to be configured independently, without the need for other microcontrollers like Arduino.
- The problem is that, in order to build a development base focused on ESP8266, we need to build a region with other integrated components to ~~facilitate~~ facilitate the work. That is why a number of bad projects including EXP8266 have emerged, In which we highlight the following :-
  - NodeLHc
  - ESP201
  - WemosD1
  - Adafruit HUZZAH ESP8266
  - Node MCU

✱ Node MCU (cont.) :-

- Node MCU is a complete hardware and IoT prototyping software that includes the following features :-
  - Control board including ESP8266 module
  - Micoo USB port for power (5 volts) ouce editing
  - 10 digital inputs GPIOs operating at 3.3V and an analog input GPIO to 1.8V.

- Development kit based on the Lua Language.

★ Wifi Transceiver :-

ESP-12E Chip specification
- Tensilica Xtensa 32 bit LX106
- 80 to 160 MHz Clock Freq.
- 128kb internal RAM
- 4MB external flash
- 802.11 b/g/n Wifi-transceiver

- The ESP8266 integrates 802.11 b/g/n HT40 wifi transceiver, so it can not only connect to a wifi network and interact with the Internet, but it can also set up a network of its own, allowing other devices to connect directly to it.

★ Power Requirement :-

Power Requirement :-
- Operating Voltage : 2.5V to 3.6V
- On board 3.3V 600mA regulator
- 80mA Operating Current
- 20μA during Sleep Mode.

★ Peripherals and I/O
- ADC channel - A 10-bit ADC channel.
- UART interface - UART interface is used to load code serially.
- PWM output - PWM pins for dimming LEDS or controlling motors.

- SPI, I2C & I2S interface - SPI and I2C interface to hook up all sorts of sensors and peripherals.
- I2S interface - I2S interface to add sound to our project.

Multiplexed I/Os

- 1 ADC channels
- 2 UART interfaces
- 4 PWM outputs
- SPI, I2C & ~~I2S~~ I2S interface

I2S ↑ I not 1

* Onboard LED and Switch :-

- The ESP8266 NodeMCU features two buttons. One marked as RST located on to top left corner is the Rest button, used of course to reset the ESP8266 chip. The other FLASH button on the bottom left corner is the download button used while upgrading firmware.
- Switches & Indicators
  - RST - Reset the ESP8266 chip
  - FLASH - Download new programs
  - Blue LED - User Programmable

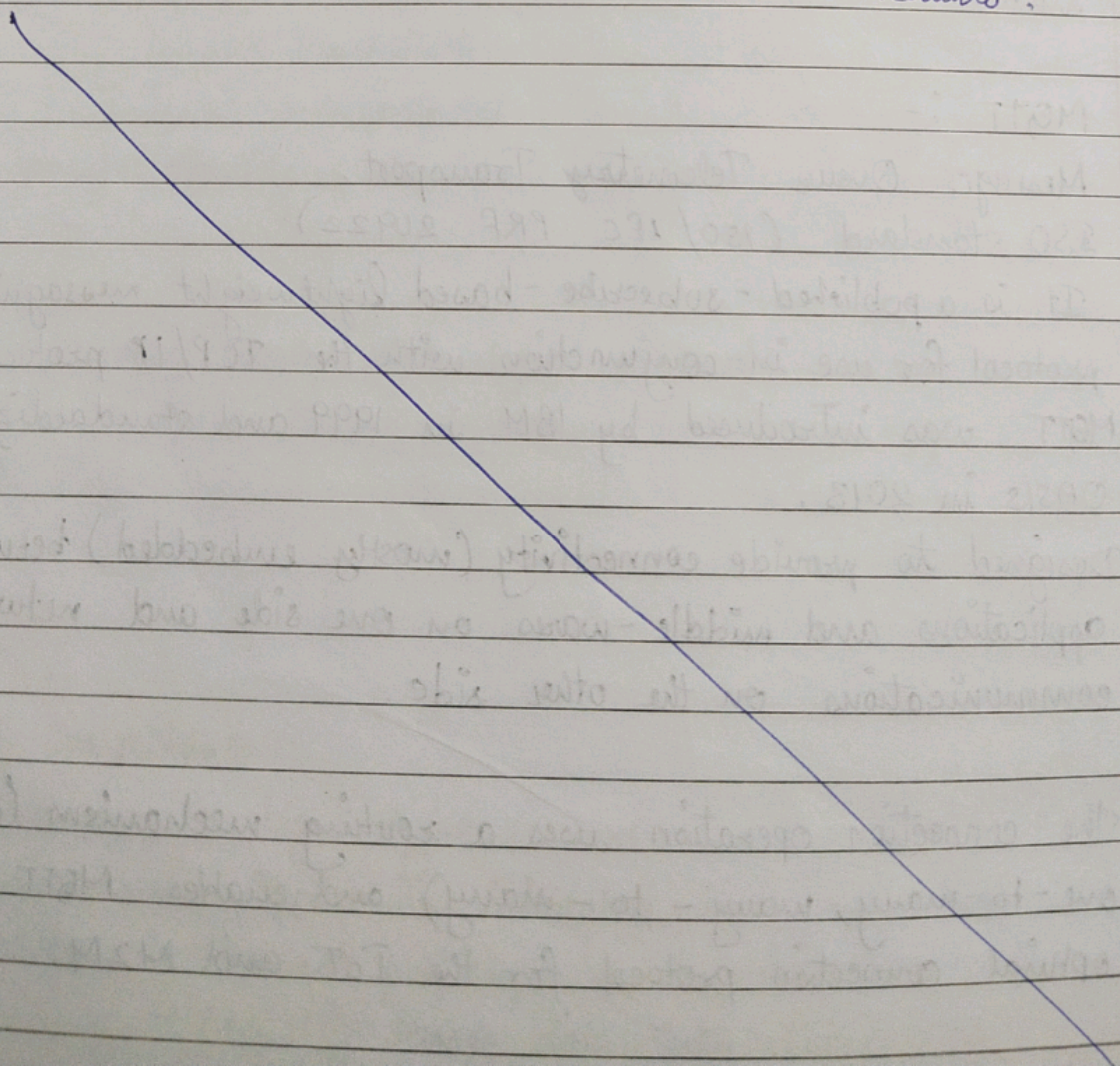
* Serial Communication :-

- The board includes CP2102 USB-to-UART Bridge Controller from Silicon Labs, which converts USB signal to serial and allows your computer to program and communicate with the ESP8266 chip.

★ Node MCU

- Advantages of Node MCU platform relative to the Arduino.
  - Low Cost
  - Integrated support for Wifi network
  - Reduced size of the board
  - Low energy consumption

- Disadvantages :-
  - Need to learn a new language and IDE
  - Reduced pinout
  - Scarce documentation
  - The ~~to~~ Node MCU programming can be as easy as in Arduino.



★ Summary 6LOWPAN :-

- Uses open IP Standards
- Offers End to End IP Addressable Nodes.
- Small packet size
- 16-bit short or IEEE 64-bit extended media access control addresses
- Low bandwidth (250/40/20 kbps)
- Topologies include star and mesh
- Low power, typically battery operated
- Relatively low cost
- Networks are ad hoc and devices have limited accessibility and user interfaces.
- Inherently unreliable due to nature of devices in the wireless medium

★ MQTT :-

- Message Queue Telemetry Transport.
- ISO standard (ISO/IEC PRF 20922)
- It is a published-subscribe-based lightweight messaging protocol for use in conjunction with the TCP/IP protocol.
- MQTT was introduced by IBM in 1999 and standardized by OASIS in 2013.
- Designed to provide connectivity (mostly embedded) between applications and middle-wares on one side and networks and communications on the other side.
- The connection operation uses a routing mechanism (one-to-one, one-to-many, many-to-many) and enables MQTT as an optimal connection protocol for the IoT and M2M.

- MQTT utilizes the publish/subscribe pattern to provide transition flexibility and simplicity of implementations
- Also, MQTT is suitable for resource constrained devices that use unreliable or low bandwidth links.
- MQTT is built on top of the TCP protocol.

★ Message broker :-
- A message broker controls the publish - subscribe messaging pattern.
- A topic to which a client is subscribed is updated in the form of messages and distributed by the message broker.
- Topics are UTF - 8 strings
- Designed for :
  - Remote connections
  - Limited Bandwidth
  - Small - code footprint
- 3 levels of Quality of Services

★ MQTT :-
- Binary header and lightweight protocol.
- Agnostic to the content of payload.
- Retain Flag
- Last Will
- Keep Alive

★ MQTT Components :-
1. Publishers :- Lightweight sensors
2. Subscribers :- Applications interested in sensor data.
3. Brokers :- Connect publishers and subscribers
   - Classify sensor data into topics.

★ MQTT Methods :-
- Connect
- Disconnect
- Subscribe
- Unsubscribe
- Publish

★ Communication :-
- The protocol uses a publish/subscribe architecture (HTTP uses a request/response paradigm).
- Publish/subscribe is event-driven and enables messages ~~to~~ to be pushed to clients.
- The central communication point is the MQTT broker, which is in charge of dispatching all messages between the senders and the rightful receivers.
- Each client that publishes a message to the broker, includes a topic into the message. The topic is the routing information for the broker.
- Each client that wants to receive messages subscribes to a ~~cert~~ certain topic and the broker delivers all messages with the matching topic to the client.
- Therefore the clients don't have to known each other. They only communicate over the topic.
- This architecture enables highly scalable solutions without dependencies between the data producers and the data consumers.

★ MQTT Topics :-

- A topic is a simple string that can have more hierarchy levels, which are separated by a slash.
- A sample topic for sending temperature data of the living room could be house/living-room/temperature.
- On one hand the client (e.g. mobile device) can subscribe to the exact topic or on the other hand, it can use a wildcard.

- Subscribing to house/+/temperature will result in all messages sent to a house with a predefined theme/living room/temperature, and any subject with a negative value in the living room area, such as house/kitchen/temp.
- The plus sign is a single level indicator and only allows for unreasonable values one category.
- If more than one level needs to be registered, such as, every small tree, there is also multi-level wildcard (#).
- Allows subscription to all levels of hierarchy below.

★ QOS (Quality of Service)

- Quality of Service requirements are technical specifications that specify the system quality of features such as performance, availability, scalability, and serviceability.

★ SMQTT :-

- Protected MQTT is an extension of the MQTT encrypted users based on lightweight encrypted attributes
- The great advantage of using such encryption is the streaming encryption feature, where one message is encrypted and

delivered to multiple nodes, which is common in IoT systems.

- Generally, the algorithm consists of four main categories: setup, encryption, publishing.
- In the setup phase, registrars and publishers register themselves to the seller and obtain the key secret, according to their choice of keyword generator algorithm.
- Once the data is published, encrypted & published by the merchant and sent to the subscribers, it is finally decrypted, at the end of the subscriber with the same secret key.
- Generation key & encryption algorithms are not set.
- SMQTT is only ~~[illegible]~~ proposed to improve MQTT security features.

☆ IoT Middle Ware :-

- Intermediate Internet is software that acts as a visual connector between IoT components that makes connections between objects that they would otherwise not known.
- Middleware is part of the architecture that allows for the interconnection of large numbers of objects by providing a layer of sensory connectivity and application layers that provide services that ensure effective communication between software.
- These products provide API management as well as basic messaging, routing and message modification. Comprehensive IoT platforms include middleware as well as sensors and communications components.
- Mulesoft, Oracle & Redhat.

SDS Page No 89
Date

22nd April 2022

REST - Representation State Transfer

- REST - An architectural Style of Networked System
  - Underlying architectural model of the world
  - Guideline framework for web protocol standards.

- REST based web services
  - Online shopping
  - Search services
  - Delivery Services

Features of REST :-

- Abstraction - Resource and not the service
- Client - Server Architecture
- Acts upon resources
- Resource could be sensors, actuator or a control system
- A resource is uniquely identified by its URL.
- A web service can be called RESTful - Client Server, Stateless, cachcable, layered system and uniform interface
- Stateless - servers will not store anything about the latest HTTP request made by the client. It will treat everyone's request as new. No session, no history.
- Uniform Interface - A developer becomes familiar with one of your APIs he should be able to follow a similar approach.
- Cacheable - Cookies where we ~~took~~ are told to accept cookies, these means those are some kind of small data stored.

- URL - Uniform Resource Identifier

#25th April 2022

★ REST methods :-
- POST - Creating a resource
- GET - Reading a resource
- PUT - Updating a resource
- DELETE - Deleting a resource.

★ HTTP and CoAP Bridging :-
CoAP : Constrained Application Protocol
- Web transfer protocol for use with constrained nodes and network.
- Designed for Machine to Machine applications such as smart TV & HAS.
- Based on request response - Response model between end points.
- Client Server interaction is asynchronous over a datagram oriented transport protocol such as UDP.
- CoAP architecture is divided into two main sub layers :-
  - Messaging
  - Request/Response
- CoAP messaging nodes
- Confirmable messages
- non-confirmable messages.
- Piggy bank message
- Separate message

★ Features :-
- Reduced overheads and parsing complexity.
- URL had content-type support.
- Support for the discovery of resources provided by known CoAP services.
- Simple Subscription for a resource, and resulting push notifications.
- Simple caching based on maximum message age.