Mid-Term Note (Summer 2025)

Course code: CSE233

Course Title: Embedded Systems and IoT

Provided by:

Name: Bishal Biswas

Section: 61_G

Attention: Since this is a handwritten note, there may be minor errors like spelling mistakes, calculation mistakes, or something else. For which I apologize. Those who read this note should first follow the rules and instructions of their Course teacher and then, if they wish, follow this note as well.

Planned effort never goes in vain.

(Best of Luck)

Lecture 1 : Embedded systems & IOT Introduction

Introduction to Embedded System :

(i) What is an Embedded system?

⇒ An embedded system is a combination of software & hardware (Like microcontroller or microprocessor), made to do specific tasks quickly & also work on its own without needing help. It must also work in real time without delays.

(ii) Characteristics of Embedded systems :

- Works for specific task only.
- Often runs without human helps.
- uses Less Power & Less memory.
- Small in size & Low cost.
- Real time operation (gives quick result)
- Minimal user Interface use.
- Work within a specific time.

(iii) Applications of Embedded system :

- Washing Machine : control washing, rinshing & spinning Auto

- Traffic Light : Manage red, yellow & green lights in order

- Remote control : sends signals to control TVs or other devi

- Car Engine control system : controls engines functions for better performance

- Smartwatch : shows time & tracks health using small computer chips.

- Digital camera : captures & stores photos and videos

- Gamming consol : Runs games & handles controls and display

- Medical Devices : check & support body functions like heartbeat or sugar level

Example :

🗆 **Microprocessor**: A microprocessor is a computer processor contained on a single integrated circuit (IC) or a small number of ICs.

🗆 **Micro Controller**: A microcontroller is a compact integrated circuit designed to govern a specific operation in an embedded system.

(iv) **Difference between Micro Processor & Micro controller**:

Microprocessor	Micro controller
• used in general purpose of computer	• used in specific control tasks
• Parts inside only CPU, needs RAM, ROM, I/O separately	• CPU, RAM, I/O all in one chip
• More expensive	• comparably cheaper more.
• use more Power	• Less power consume
• High Processing speed	• comparably slower
• Needs complex Programming	• Easy & task-based Programming
• Ex: Laptop, Desktop (Intel - core i3 / i5 / i7 / i9 / AMD Ryzen 3/5/7 ...)	• Ex: Digital cammera, smartwatch, Remote control (Arduino, ARM cortex-M series)

⊞ What is Arduino?

⇒ Arduino is an open-source Platform designed for building & Prototyping electronics Project.

- It consist of
  - → Hardware : Microcontroller boards
  - → Software : Arduino IDE → for Programming

→ beginner friendly

→ Simple Programming Interface

→ Wide range of boards

→ Suitable for task like : Automation, IOT, Robotics & more

→ Version : Mega, UNO, Nano

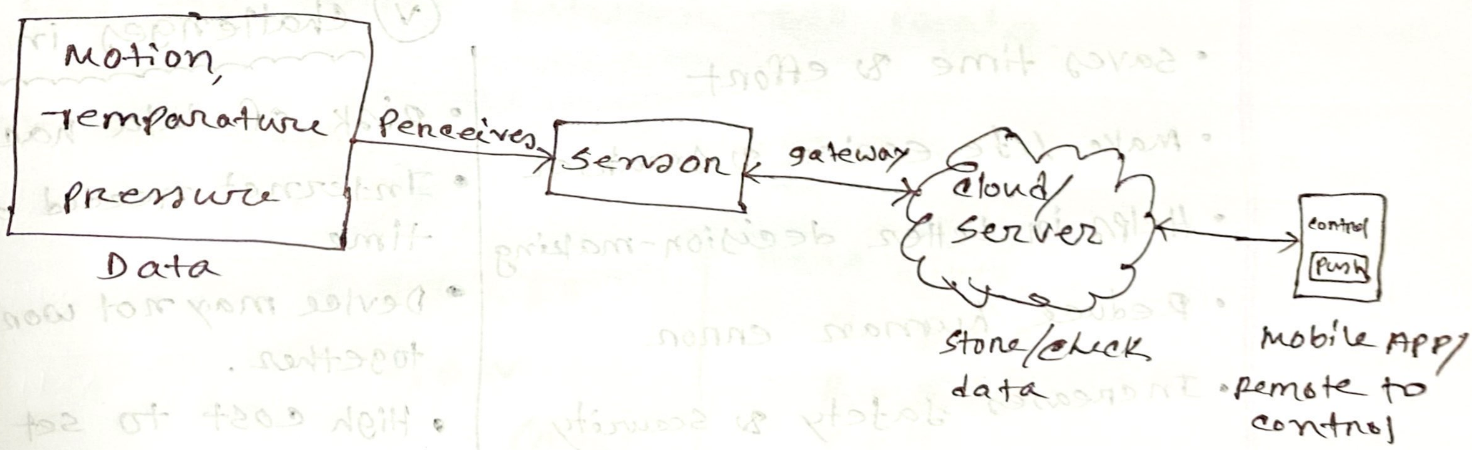
4) Introduction to IOT

(i) What is IOT?

⇒ IOT (Internet of things) means connecting everyday objects to the internet to share data.

(ii) How IOT works?

⇒
- Device have sensor
- sensors collect data
- Data is sent to the Internet
- The cloud stores & checks data
- you can control things using apps.



(iii) Application of IOT :

- Smart homes : control lights, Ac, & devices using phone or voice
- Smart Farming : check crops health using sensors & data.
- Health tracking : Fitness bands track steps, heart rate, & sleep.
- Smart city traffic control : Manage traffic using sensor and smart signals
- Industry machine monitoring : Watch machines to detect Problems early.

(iv) Benefits of IOT :

- saves time & effort
- Make life easier & smarter
- Helps in better decision-making
- Reduce human error
- Increases safety & security

(v) Challenges in IOT :

- Risk of data hacking
- Internet needed all the time
- Device may not work together.
- High cost to set up
- Needs strong data Protection

Ohm's Law : Ohm's law state that the electric current through a conductor between two points is directly propotional to the voltage across the two points.

$$I \propto V$$

$$\Rightarrow I = GV$$

$$\Rightarrow I = \frac{1}{R} V$$

$$\Rightarrow V = IR$$

$$\begin{bmatrix} 1\,k\Omega = 1000\,\Omega \\ 1\,mA = 0.001\,A \\ 1\,kW = 1000\,W \end{bmatrix}$$

Here,
V = voltage
I = current
R = Registance

Ex : If the resistance of an electric iron is $50\,\Omega$ & a current of $3.2\,A$ flows through the registance. Find the voltage between two points.

$\Rightarrow$ We know,

$$V = IR$$
$$= 3.2 \times 50$$
$$= 160\,V$$

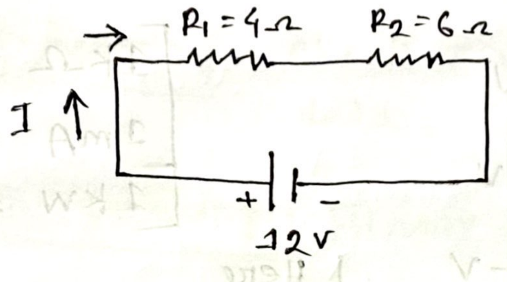
(Ans.)

Here Given,
$I = 3.2\,A$
$R = 50\,\Omega$
$V = ?$

Formula :

- $V = IR$
- $P = VI$ [Power = P]
- Series : $R_S = R_1 + R_2 + R_3 + \dots R_n$
- Parallel : $R_P = \left(\frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3} + \dots \frac{1}{R_n}\right)^{-1}$

Ex: In a series circuit with a 12-volt battery, if there are two resistor of $4\,\Omega$ & $6\,\Omega$, what is the current flowing through the circuit?



⇒ In series:

$$R_S = R_1 + R_2$$
$$= 4 + 6$$
$$= 10\,\Omega$$

Given,

$V = 12\,V$

$R_1 = 4\,\Omega$

$R_2 = 6\,\Omega$

$I = ?$

We know,

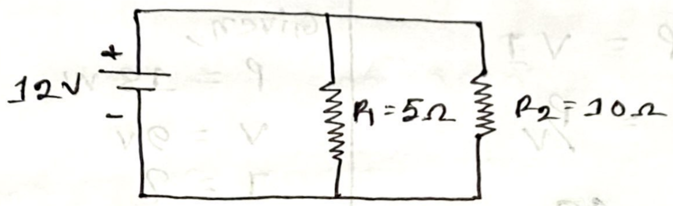
$$V = IR$$

$$\therefore I = \frac{V}{R_S}$$

$$= \frac{12}{10}\,A$$

$$= 1.2\,A \quad \text{(Ans.)}$$

Ex: calculate the total resistance in a parallel circuit with two resistor of 5Ω & 10Ω. If a voltage of 12 V applied across the circuit, what is the total current flowing through the circuit.



⇒ Given,

$V = 12V$

$R_1 = 5\Omega$

$R_2 = 10\Omega$

In parallel circuit:

$$\frac{1}{R_P} = \frac{1}{R_1} + \frac{1}{R_2}$$

$$= \frac{1}{5} + \frac{1}{10}$$

∴ Total registance $R_P = \left(\frac{1}{5} + \frac{1}{10}\right)^{-1}$

$= 3.33\Omega$ (Ans.)

Now,

we know,

$V = IR$

$$I = \frac{V}{R_P} = \frac{12}{3.33}$$

$= 3.6A$ (Ans.)

Scanned with CamScanner

Ex: A device using 18 watts of power is connected to a 9 V battery. calculate the current drawn by the device & determine the resistance of the device.

$\Rightarrow$ We know,

Given,
$P = 18 W$
$V = 9 V$
$I = ?$
$R = ?$

Power, $P = VI$

$\Rightarrow I = P/V$

$= \frac{18}{9}$

$= 2 A$ (Ans.)

We know,

Voltage, $V = IR$

$\Rightarrow R = V/I$

$= \frac{9}{2}$

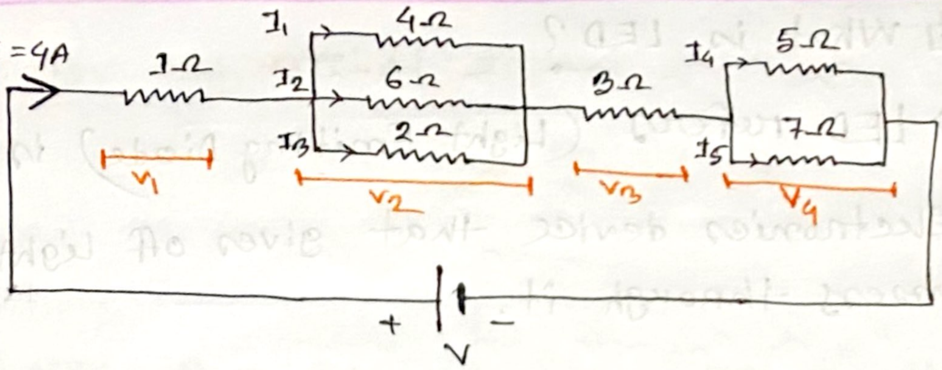
$= 4.5 \Omega$ (Ans.)

Attention:

→ In series circuit: same current but voltage divide across resistors.

→ In parallel circuit: same voltage but current divide across resistors.

Ex:



From the above circuit find out I_1, I_2, I_3, I_4, I_5. Where total current is 4 A.

Given,
$R_1 = 1\Omega$
$R_2 = 4\Omega$
$R_3 = 6\Omega$
$R_4 = 2\Omega$
$R_5 = 3\Omega$
$R_6 = 5\Omega$
$R_7 = 7\Omega$
$I = 4A$

$\Rightarrow$ • $V_1 = I \times R_1$
$= 4 \times 1$
$= 4V$

• **Now for V_2:**

$$R_P = \left(\frac{1}{R_2} + \frac{1}{R_3} + \frac{1}{R_4}\right)^{-1}$$
$$= \left(\frac{1}{4} + \frac{1}{6} + \frac{1}{2}\right)^{-1}$$
$$= 1.09\Omega$$

$V_2 = I \times R_P = 4 \times 1.09\Omega$
$= 4.36 V$

• $V_3 = I \times R_5 = 4 \times 3$
$= 12V$

• **For V_4:**

$$R_P' = \left(\frac{1}{R_6} + \frac{1}{R_7}\right)^{-1}$$
$$= \left(\frac{1}{5} + \frac{1}{7}\right)^{-1}$$
$$= 2.92\Omega$$

$V_4 = I \times R_P'$
$= 4 \times 2.92$
$= 11.67 V$

Current:

$I_1 = \frac{V_2}{R_2} = \frac{4.36}{4} = 1.09 A$

$I_2 = \frac{V_2}{R_3} = \frac{4.36}{6} = 0.72A$

$I_3 = \frac{V_2}{R_4} = \frac{4.36}{2} = 2.18 A$

$I_4 = \frac{V_4}{R_6} = \frac{11.67}{5} = 2.33A$

$I_5 = \frac{V_4}{R_7} = \frac{11.67}{7} = 1.67 A$

☐ What is LED?

⇒ LED refers (Light Emitting Diode) is a small Electronics device that gives off light whe electricity Passers through it.

☐ **capacitor** : A capacitor is a small electronics element that store charges & releases charges quickly.

☐ **capacitance** : capacitance is a measure of a capacitor's ability to store charge on its plates.

☐ **1 farad** : It means a capacitor can store $1C = 6.242 \times 10^{18}$ electrons charge when 1 volt is applied. (Unit of capacitance)

• **Formula** :

- $Q = CV$ [Q = charge (C), C = capacitance (F), V = Votage]
- series : $C_{total} = \left(\frac{1}{C_1} + \frac{1}{C_2} + \dots \frac{1}{C_n}\right)^{-1}$
- Parallel : $C_{total} = C_1 + C_2 + \dots C_n$
- $1 \mu F = 1 \times 10^{-6} F$

Note : capacitance decrease in series, capacitance Increase in Parallel

Ex: If 40V are applied across a 470 μF capacitor. find the charge on the plates.

$\Rightarrow$ We know

charge $Q = CV$

$= (470 \times 10^{-6} \times 40)$

$= 0.0188\ C$ (Ans.)

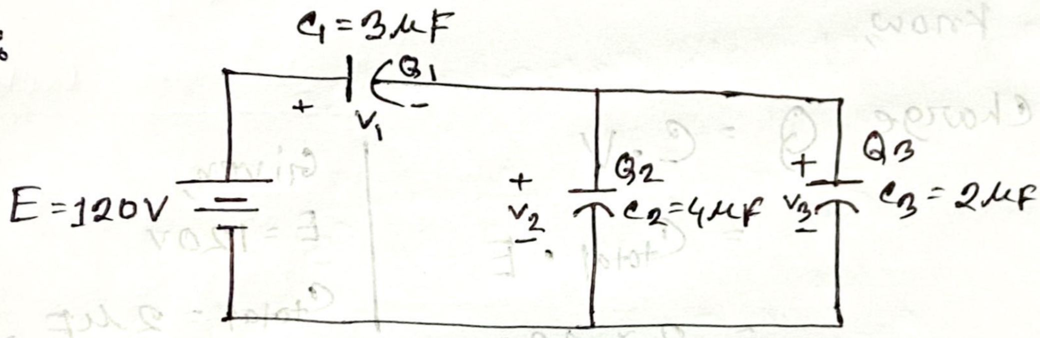
Given,

capacitance, $C = 470\ \mu F$

$= 470 \times 10^{-6} F$

voltage, $V = 40\ V$

Ex:



(a) Find the total capacitance.

(b) Determine the charge on each plate.

(c) Find the voltage across each capacitor.

solution:

(a) C_2 & C_3 are in parallel:

$$C_p = C_2 + C_3$$
$$= 4 + 2 = 6\,\mu F$$

C_1 & C_p are in series:

$$C_{total} = \left(\frac{1}{C_1} + \frac{1}{C_p}\right)^{-1}$$
$$= \left(\frac{1}{3} + \frac{1}{6}\right)^{-1}$$
$$= 2\,\mu F$$

(b) we know,

charge $Q = C \cdot V$	Given,
$= C_{total} \cdot E$	$E = 120V$
$= 2 \times 120V$	$C_{total} = 2\mu F$ ---- (a)

$= 240\,\mu C$; this charge is on C_1 ; so $Q_1 = 240\mu C$.

(series combination → same charge through all components).

(c) Voltage across C_1:

$$V_1 = \frac{Q}{C_1} = \frac{240\,\mu C}{3\,\mu F}$$
$$= 80\,V \quad \text{(Ans.)}$$

Voltage across parallel C_2 & C_3:

$$V_2 = V_3 \Rightarrow (E - V_1)$$ [Parallel G- voltage same]
$$\Rightarrow 120 - 80$$
$$\Rightarrow 40\,V \quad \text{(Ans.)}$$

Now charge on each plates:

$$Q_1 = 240\,\mu C$$

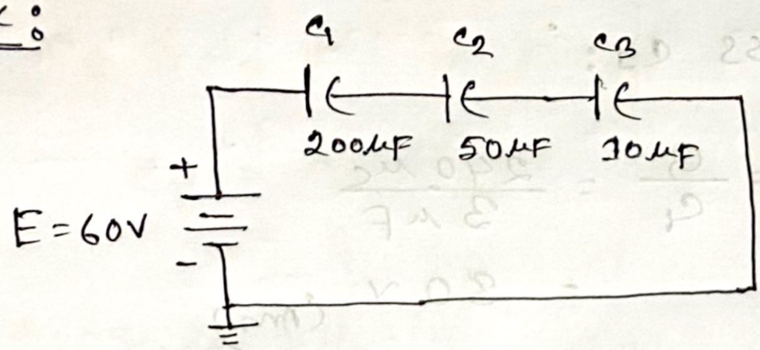
$$Q_2 = C_2 \cdot V_2 = (4\,\mu F \cdot 40\,V)$$
$$= 160\,\mu C$$

$$Q_3 = C_3 \cdot V_3 = (2\,\mu F \cdot 40\,V)$$
$$= 80\,\mu C$$

Attention:

- In parallel capaciton: same → voltage; Divide → charge
- In series capaciton: same → charge; Divide → voltage

Ex:



(a) Find the total capacitance

(b) Determine the charge on each plate

(c) Find the voltage across each capacitor

→ (a) For series total capacitance:

$$\frac{1}{C_T} = \frac{1}{C_1} + \frac{1}{C_2} + \frac{1}{C_3}$$

$$C_T = \left(\frac{1}{C_1} + \frac{1}{C_2} + \frac{1}{C_3}\right)^{-1}$$

$$= \left(\frac{1}{200} + \frac{1}{50} + \frac{1}{10}\right)^{-1}$$

$$= 8\,\mu F$$

(b) Though all are series connection so charge Q on each all capacitor are same.

$$Q = C_T \cdot V$$

$$= 8\,\mu F \cdot 60V$$

$$= 480\,\mu C$$

$$\therefore Q_1 = Q_2 = Q_3 = 480\,\mu C.$$

(c) Voltage on each capaciton :

$$V_1 = \frac{Q}{C_1} = \frac{480}{200} = 2.4 \text{ V}$$

$$V_2 = \frac{Q}{C_2} = \frac{480}{50} = 9.6 \text{ V}$$

$$V_3 = \frac{Q}{C_3} = \frac{480}{10} = 48 \text{ V}$$

Some Theory :

- **Short circuit :** A short circuit happens when electricity flows through a shortcut path with very low resistance, usually by accident. This can cause high current, heat or even damage to the circuit.

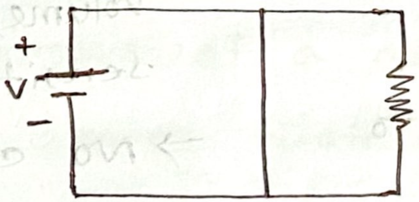
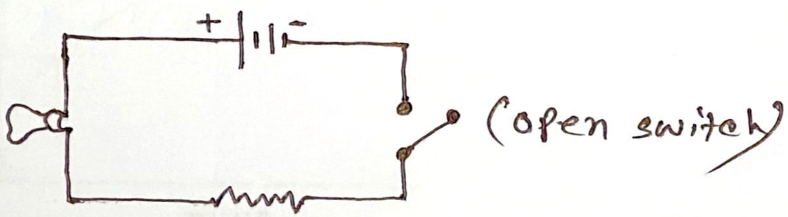


fig: Short circuit

- **Open circuit :** An open circuit means there is a break or gap in the path, so current can't flow. The circuit is in-complete. Ex: A switch is turned off - the wire is cut.



• **Resistance :** Resistance is the opposition to the flow of electric current in a circuit. It is measured in Ohms (Ω).

High resistance = Less current

Low resistance = more current

$$R = \frac{V}{I}$$; R = Resistance.
V = voltage.
I = current.

• **Types of Resistors :**

(i) Fixed Resistor

→ constant resistance value

→ can't be changed

→ use color code

→ used in circuits to control current or voltage.

(ii) Variable Resistor (potentiometer)

→ Adjustable resistance

→ used to control brightness, volume, speed or sensor sensitivity

→ No color code use

→ Ex: volume knob in a speaker.

⊞ Types of capacitors : There are many types, but two main categories : (i) fixed capacitor (ii) variable capacitor

(i) fixed capacitor

→ fixed capacitance value

→ you can't change their capacity

→ used in most electronic circuits

→ Types include ceramic, electrolytic, tantalum etc.

(ii) variable capacitor

→ capacitance value can be adjusted manually

→ used in radios, tuners & special circuit

→ Generally used in High-frequen devices

→ Types include Trimmer, tuning, vaccum variable etc

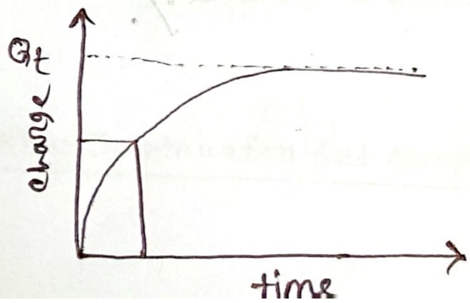
⊞ Charging & Discharging phase of capacitor :

• Charging phase :

→ When you connect a capacitor to a power source, it starts storing electric charge.

→ The voltage across the capacitor gradually increase.

→ The current is high at first, then slowly decreases as it becomes full.

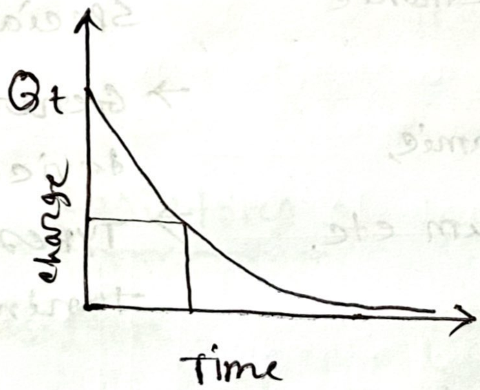


• Discharging Phase :

→ When the power source is removed, the capacitor releases its stored energy.

→ The voltage & current go down slowly

→ The capacitor gives power to the circuit for a short time.



• Why capacitor used in Embedded System?

⇒ capacitors stores and release electrical energy. They are used to :

→ smooth power supply
→ filter noise
→ help sensor & micro controller work steadily.

Lecture 2 : Introduction to Microcontroller

• **Microcontroller** : A microcontroller is a small computer on a single chip that has a processor, memory & I/O ports to performe a specific task.

• **Types of microcontroller** :

(i) 4 -bit : → simple & slow
→ used in basic device like toys, calculator etc.

(ii) 8 -bit : → simple & low cost
→ used in small device like Arduino & remote control

(iii) 16 - bit : → Medium performance
→ used in industrial machine & motor control system

(iv) 32- bit : → High performance & fast
→ used in smart device like smartphone, tablets etc.

(v) Embedded microcontroller : → Built inside a device for a specific task
→ Ex: microwave oven controller.

(vi) External memory microcontroller : → use external RAM/ROM to store data or program
→ used when large memory needed.

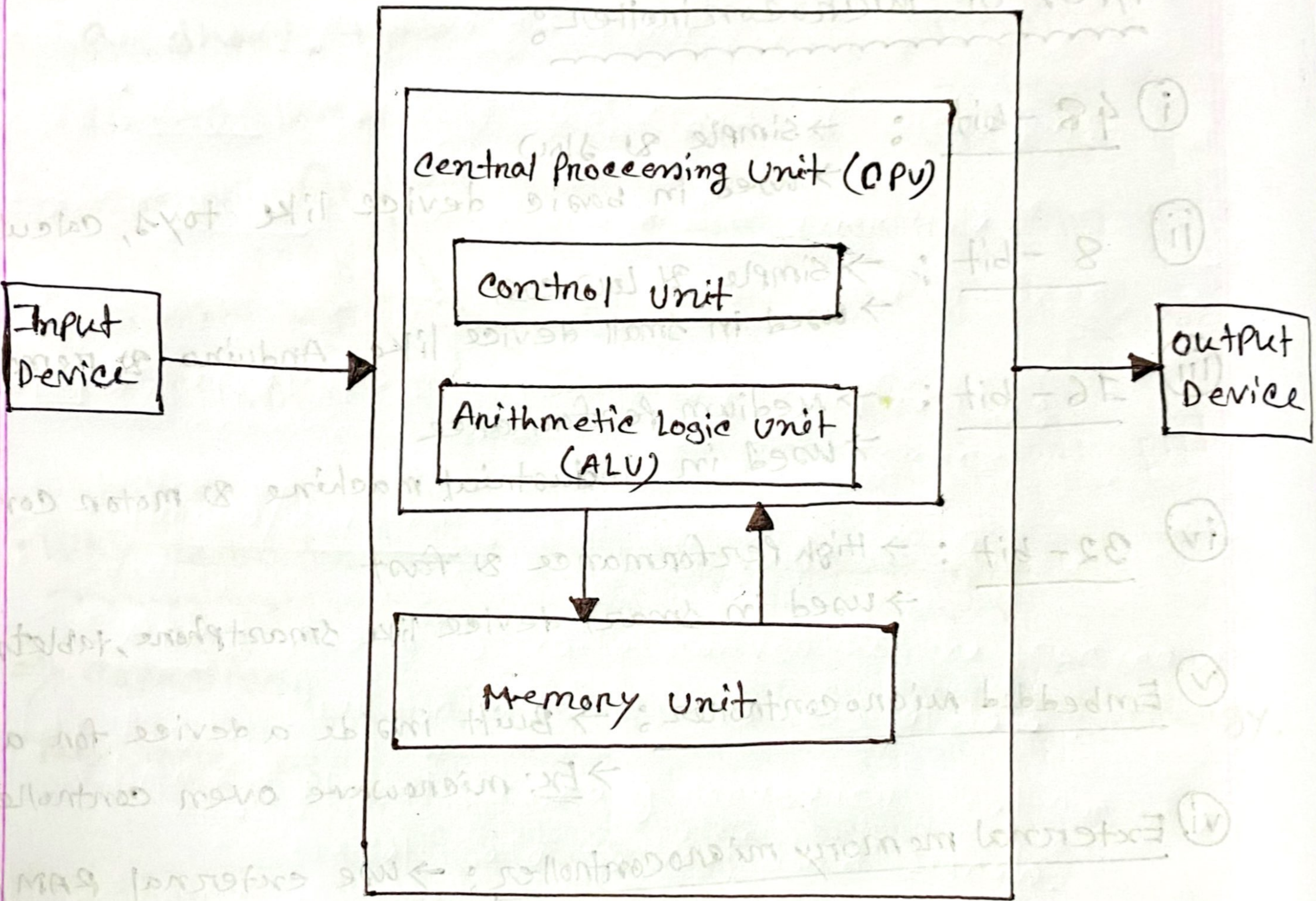
(vii) RISC (Reduced Instruction set computer) : → simple & fast
→ Less power consume
→ Ex: ARM micro controller.

(viii) CISC (complex Instruction set computer) : → Handle more complex Instruction
→ Slower & many Instruction
→ Ex: Intel 8051

• Microcontroller Architecture : Two main architecture

(i) Harvard (ii) Von Neumann

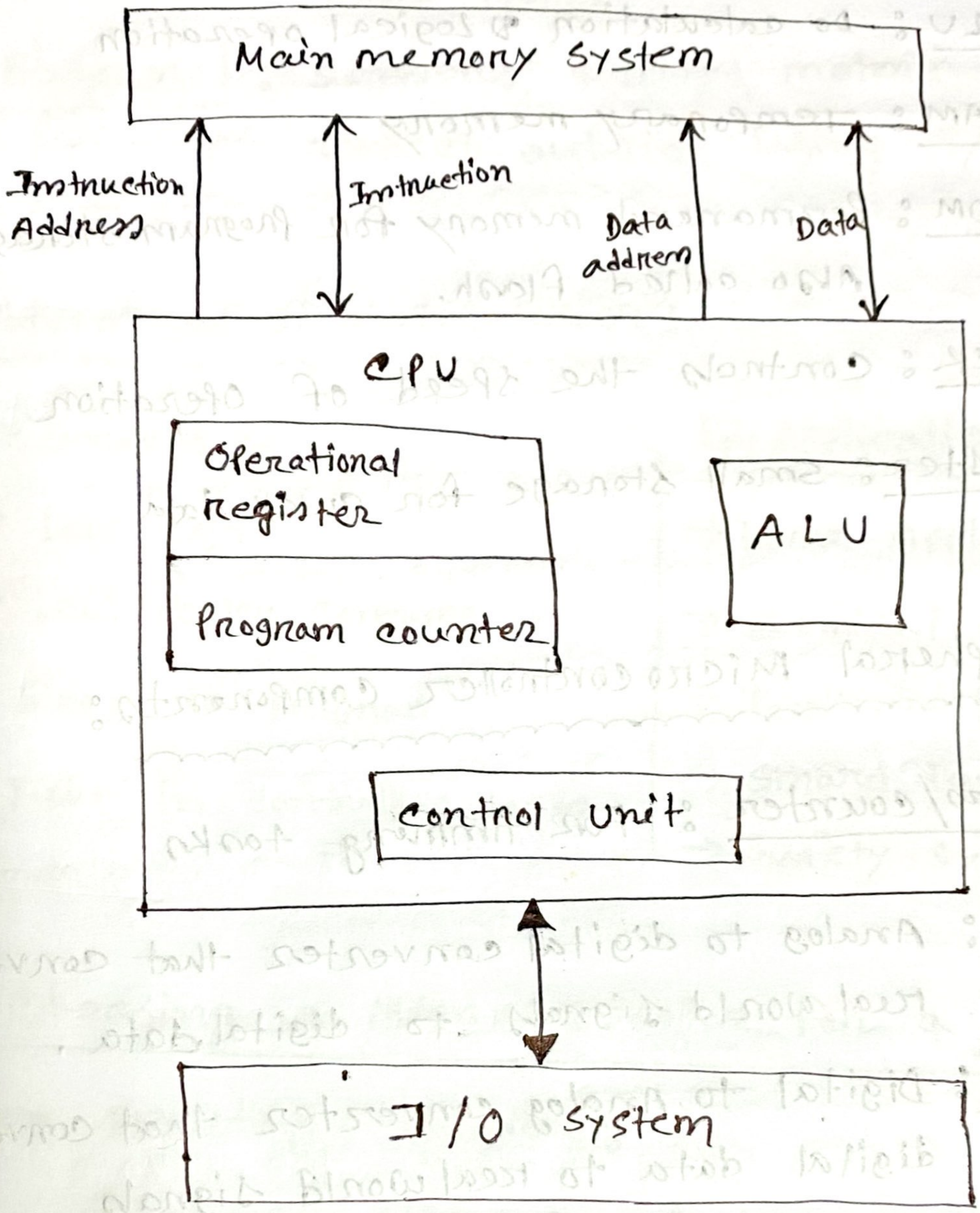
• Von-neumann architecture : same memory for data & Programs.



Weakness : Two separate fetches can slow up the controllers operation.

Example : Motorola 68HC11

• Harvard Architecture : separate memory for data & programs.



→ Execution ocur in parallel
→ much faster
→ complex design
→ Ex: Intel MCS-51 family & PIC microcontroller

Core microcontroller component:

- **CPU** : The brain that runs instruction
- **ALU** : Do calculation & logical operation
- **RAM** : Temporary memory
- **ROM** : Permanent memory for program storage. Also called flash.
- **Clock** : Controls the speed of operation
- **Register** : Small storage for quick data

Peripheral Microcontroller components:

- **timers/counter** : For timming tasks
- **ADC** : Analog to digital converter that converts real world signals to digital data.
- **DAC** : Digital to Analog converter that converts digital data to real world signals.
- **USART/SPI/I2C** : Communication Interfaces.
- **GPIO** : General Purpose Input output that connects with external devices.

Internal & External operation of Microcontroller:

- **Internal:** CPU, RAM, ROM & built-in peripherals work inside.
- **External:** sensors, displays, motors & other devices connect outside using I/O ports.

Advantages of Microcontroller:

→ Small size

→ Low cost

→ Low Power consume

→ Easy to Program

→ Fast in controlling tasks

Application of Microcontroller

→ Home Appliances (TV, microwave)

→ Medical devices (heart monitor)

→ Industrial Automation

→ Smart devices (watch, thermostat)

→ safety system (ABS break airbag control)

Choosing a Microcontroller for a specific Application:

When selecting, consider:

(i) Memory size: Enough to store your code

(ii) speed: Fast enough for your task

(iii) Num. of I/O Pins: to connect devices

(iv) Power consumption: Low for battery device

(v) cost: fit to your budget

(vi) Available Peripherals: like ADC, timers etc.

Lecture 3 : Getting started with Arduino

Arduino Basic function :

Function	What it does
• pinMode (pin, mode)	→ sets a pin as input or output
• digitalWrite (pin, val)	→ sets a digital pin to High or Low
• digitalRead (pin)	→ Reads digital pin if it is High or Low
• analogWrite (pin, val)	→ sets a PWM pin to value between 0-255
• analogRead (pin)	→ Reads analog pin value (0 - 1023)
• delays (ms)	→ Pause Program for given milisec.
• serial.begin (baud)	→ start serial monitor with given speed.
• serial.Print (value)	→ send a value to serial monitor for debbugging.

Digital Pins (0 - 13) :

- used as input or output with pinMode ()
- WORK with digitalRead () & digitalWrite ()
- Each pin gives or takes up to 40mA current.
- Pin 0 (RX) & Pin 1 (TX) used for serial communication
- Avoid using Pins 0 & 1 when uploading code via USB

PWM (Pulse Width Modulation) Pins (3, 5, 6, 9, 10, 11):

- used with analogWrite() to simulate analog output
- used to dim LEDs or control motor speed

Analog Pins ($A_0 - A_5$):

- Reads voltages from 0 to 5V
- use analogRead() to get values from 0 to 1023
- commonly used with sensors.

Power Pins:

- Vin : Input power (6-20V) to Arduino from external source
- 5V : Regulated power output for components
- 3.3V : 3.3V power (max 50mA)
- GND : Ground connection
- RESET : Restarts the Arduino board

Other Pins:

- AREF : External analog voltage reference
- IOREF : shows the voltage level of the board (usually 5V)

Master → controll communication (Arduino)
Slave → respond to master command (display, sd card)

SPI communication pins (10-13) : → serial peripheral interface

- Pin 10 (SS - slave select) : used to select the SPI slave device.
- Pin 11 (MOSI - Master out slave In) : sends data from the Arduino to the connected SPI device.
- Pin 12 (MISO - Master in slave out) : Receives data from the connected SPI device.
- Pin 13 (SCK - serial clock) : provides the clock signal used to synchronized data transmission.

→ SPI allows fast & full-duplex (two-way) communication.
→ one master (Arduino) and one or more slaves.
→ SS pin is used to select which slave to talk to.

I2C communication :

- A4 – SDA (data line)
- A5 – SCL (clock line)
- used for simple communication with sensors using only 2 wires

LED Indicator : Pin 13 has a built-in LED – great for testing.

How SPI Works :

→ Master controls the communication, slaves just respond

→ Data goes both ways at the same time (full duplex)

→ Master send clock signal for sync data (via SCK pin)

→ Each slave is selected by lowering (SS pin).

→ Data sent from master to slave (via MOSI pin)

→ Data sent from slave to Master (via MISO pin)

Advantages of SPI :

- very fast data transfer
- simple to use (less code)
- full duplex ; send & Receive data same time.
- Work well with many device ; just add more SS pin.
- Reliable & accurate data communication.

Lecture 4: PWM signal (Analog IO)

- **PWM**: Pulse Width Modulation is a technique used in iot that refers a digital signal which switches between ON (1) and OFF (0) rapidly, where the duration of the ON time (called Duty cycle) controls the power deliver to the device.

- **Why PWM is used?**

  → To simulate analog output using digital pins.

  → To control speed, brightness or position in various application.

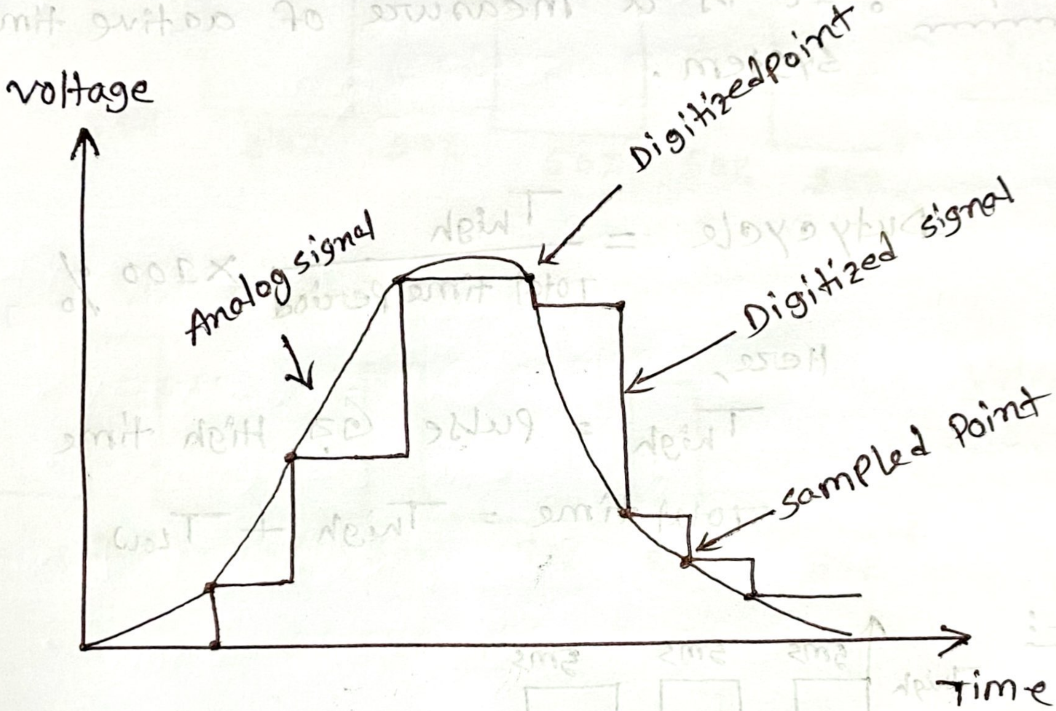
- **common used in IOT:**

  - Motor Control : speed of DC motors
  - LED Dimming : Adjust brightness of LEDs
  - servo motor control : controlling angle/position
  - signal Generation : creating audio tones or frequencies

• Analog Input :

• Think about our voice
• voice is nothing but analog signal
• Then it is digitized and Stone on transmit

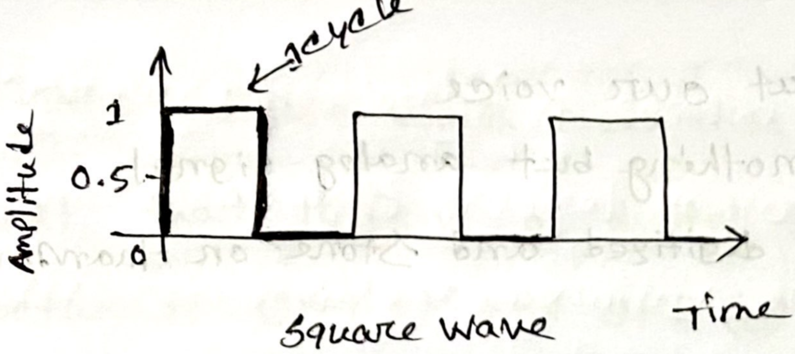
• Arduino Analog Input :



• Resolution : The number of different voltage level used to digitize an input signal.

• Sampled Point : single measurement of an analog voltage at a specific moment.

cycle :



Duty cycle : It is a measure of active time of the system.

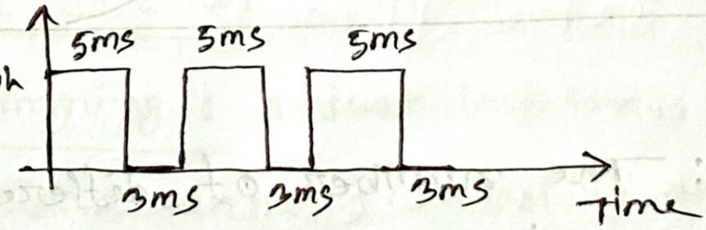
$$\text{Duty cycle} = \frac{T_{high}}{\text{Total time period}} \times 100\ \%$$

Here,

T_{high} = pulse G? High time

Total time = $T_{high} + T_{Low}$

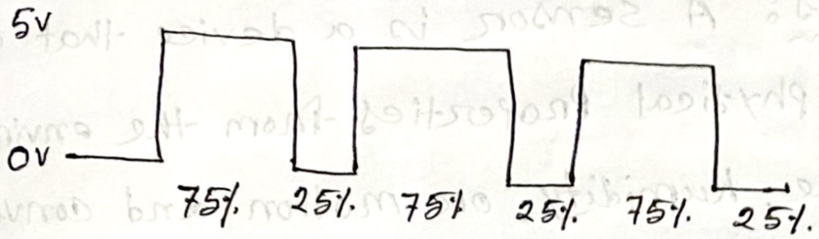
Ex:



$$\therefore \text{Duty cycle} = \frac{5}{5+3} \times 100\ \%$$

$$= 62.5\ \%$$

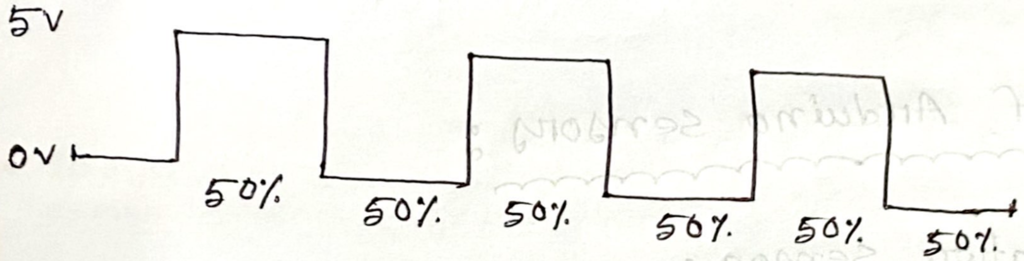
• output voltage $= \dfrac{T_{high}}{\text{total time period}} \times$ maximum voltage



Voltage : $5 \times 75\%$

$\Rightarrow 5 \times 75/100$

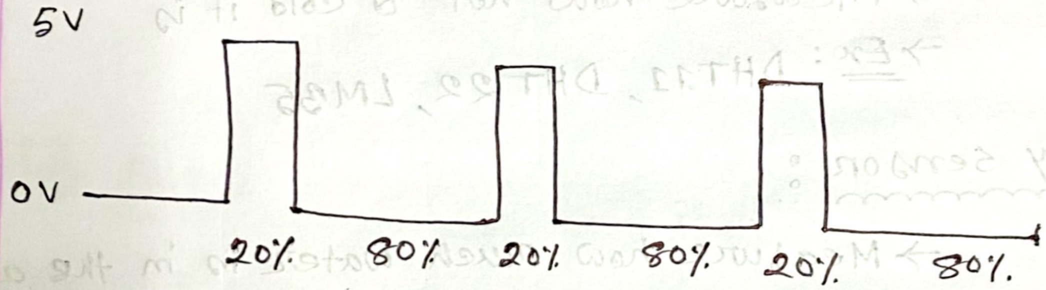
$= 3.75$ V



Voltage : $5 \times 50\%$

$= 5 \times \dfrac{50}{100}$

$= 2.5$ V



Votage : $5 \times 20\%$

$= 5 \times \dfrac{20}{100}$

$= 1$ V

Lecture 5 : Arduino sensors and Actuators

• **Arduino sensors:** A sensor is a device that detects and measures physical properties from the environment like Temperature, humidity, or motion and converts them into data for further process.

Types of Arduino sensors :

• **Temperature sensor :**

→ Measure how hot & cold it is

→ Ex : DHT11, DHT22, LM35

• **Humidity sensor :**

→ Measure how much water is in the air.

→ Ex: DHT 11, DHT 22

• **Light sensor :**

→ measures How bright the light is.

→ Ex : LDR, BH1750

• **Motion sensor :**

→ Detect movement or body motion

→ Ex: PIR, MPU6050

- **Proximity & Distance sensors:**

  → Measure how close or far an object is.

  Ex: HC-SR04, IR Sensor

- **Gas and Air Quality sensors:**

  → Detect smoke, gases or air pollution

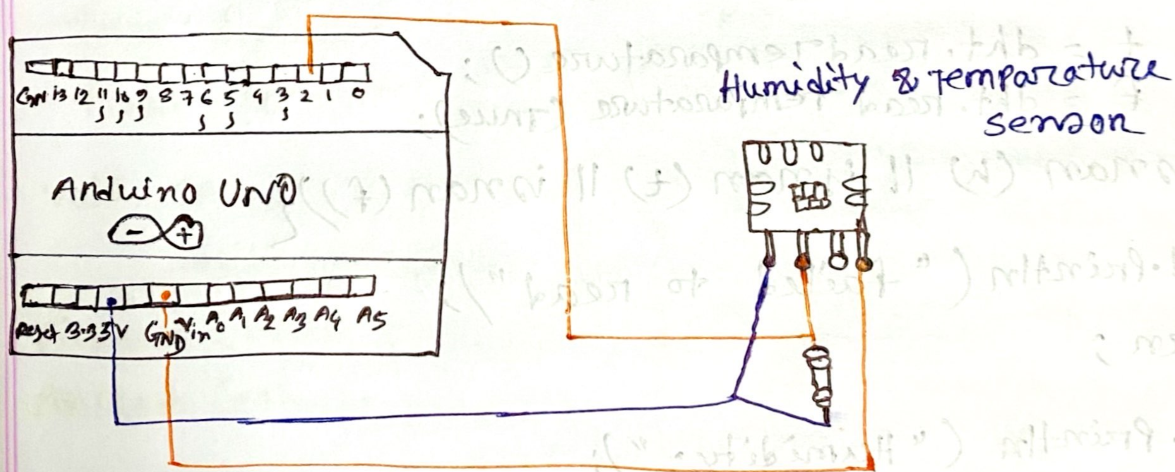
  Ex: MQ-2, MQ-135

- **Pressure sensors:**

  → Measure air or water pressure

  Ex: BMP 180, BMP 280, MPX 5700

DHT 11 Interfacing Diagram:



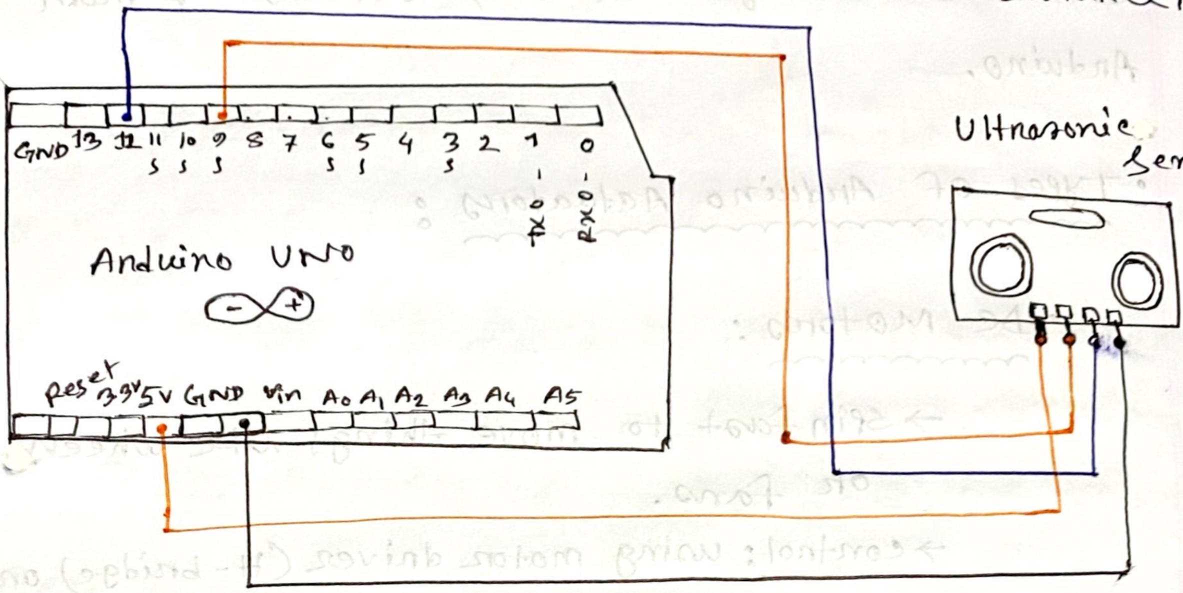
- DHT 11 Interfacing code :

```
#include "DHT.h"
#define DHTPIN 2
#define DHTTYPE DHT11

DHT dht (DHTPIN, DHTTYPE);

void setup (){
    serial. begin (9600);
    serial. Println ("DHT11 Test");
    dht. begin (); }

void loop (){
        delay (2000);
float h = dht. readHumidity();
float t = dht. readTemparature ();
float f = dht. read Temparature (True);
if (isnan (h) || isnan (t) || isnan (f)){
 serial. Println ("failed to read");
 return;
serial. Print ("Humidity: ");
serial. Print (h);
serial. Print ("Temperature: ");
serial. Print (t);
```

Scanned with CamScanner

HC-SR04 Interfacing Diagram:

→ object detection
→ Distance measure



HC-SR04 Interfacing Code:

```
int trigPin = 9;
int echopin = 12;
long duration;
float cm;

void setup() {
    serial.begin (9600);
pinMode (trigpin, OUTPUT);
pinMode (echopin, Input); }

void loop() {
digitalwrite (trigpin, Low);
delayMicroseconds (5);
digitalwrite (trigpin, High);
delaymicroseconds (10);
digitalwrite (trigpin, Low);

duration = pulseIn(echopin, HIGH);
  cm = (duration/2.0)/29.1;

serial.print ("Distance: ");
serial.print (cm);
delay (250);
}
```

Arduino Actuators: It is a device that move or control something when they get signals from the Arduino.

- **Types of Arduino Actuators:**

  - **DC Motors:**
    - → Spin fast to move things like wheels or fans.
    - → Control: using motor driver (H-bridge) or transistor.
    - → Ex: Robotic car, fan.

  - **Servo motors:**
    - → Rotate to a specific angle (0-180°)
    - → Control: PWM signal & servo library
    - → Ex: Robotics arm, grippers.

  - **Stepper motors:**
    - → Turn in small steps for accurate rotation
    - → Control: Stepper motor driver, step signal
    - → Ex: 3D printer, CNC machines.

- **Solenoids:**
  - → Push or pull with a magnetic force
  - → Control: transistor or rely
  - → Ex: Locks, valves, Pinball machines

- **Relays:**
  - → Switch high power devices using low Arduino power.
  - → Control: Triggered by signal through transistor.
  - → Ex: controlling high power devices like Light, fan, motor.

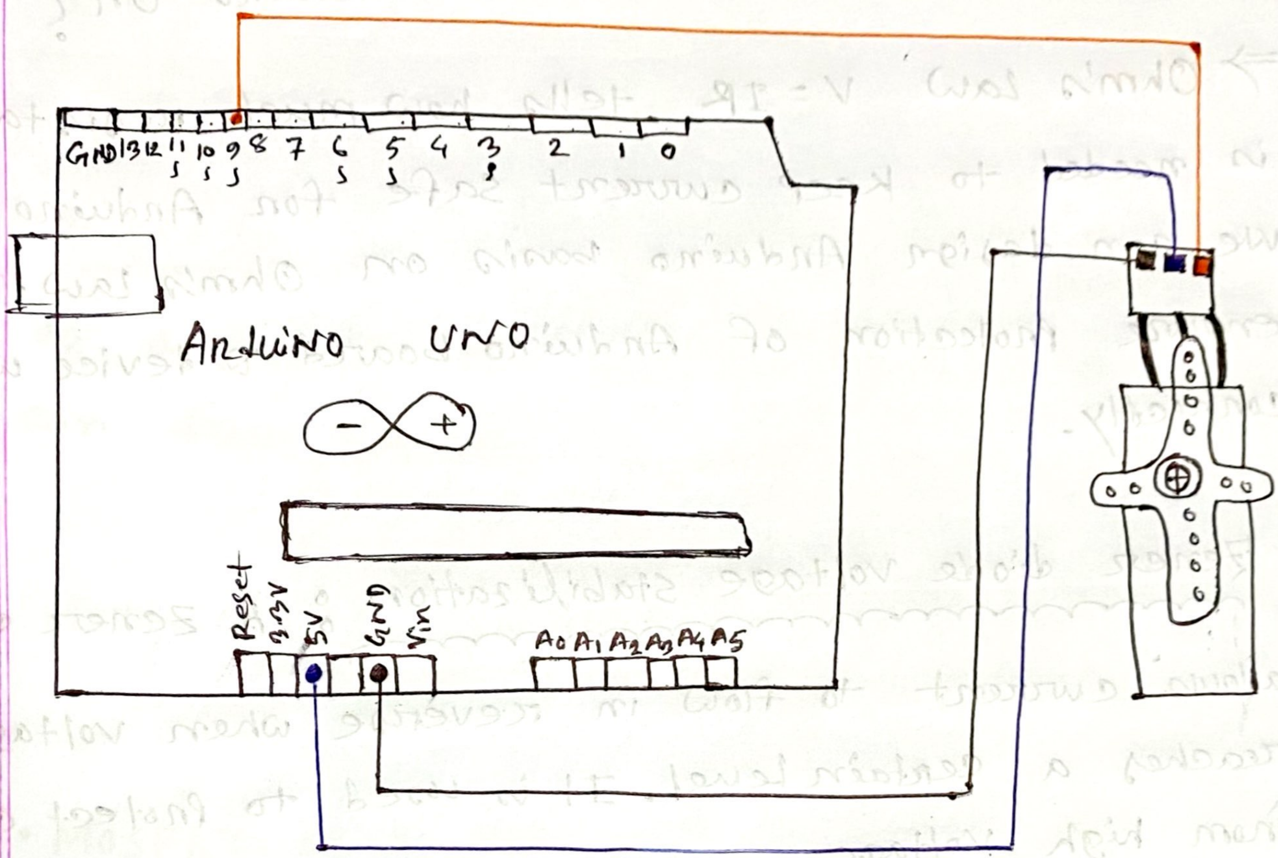
- **Piezo Buzzers/vibration motors:**
  - → Make sound & give notification, Alert with vibration
  - → control: digital pin or PWM
  - → Ex: notification, Alert

- **Heating Elements:**
  - → Produce heat for tools or device
  - → control: relay or MOSFET for more power
  - → Ex: soldering iron control.

DC Motor Interfacing Code:

```
int enable1 = 11;
int in1 = 10;
int in2 = 9;

void setup(){

pinMode(enable1, OUTPUT);
pinMode(in1, OUTPUT);
pinMode(in2, OUTPUT);
}

void loop(){
digitalWrite(enable2, HIGH);
digitalWrite(in1, HIGH);
digitalWrite(in2, LOW);
}
```

SG90/MG996R Servo Motor:



Code:

```
#include <servo.h>
 Servo myServo;
void setup(){
    myServo.attach(9);}
void loop(){
   for(int i=0; i<180; i++){
 myServo.write(i);
 delay(10);}

For(int i= 179; i>0; i--){
  myServo.write(i);
  delay(10);
   }
  }
```

Some short Questions:

- How Ohm's Law is related to Arduino or?

⇒ Ohm's law $V = IR$ tells how much resistance is needed to keep current safe for Arduino components. We can design Arduino basis on Ohm's law. It ensure protection of Arduino board & device work correctly.

- Zener diode voltage stabilization: A zener diode allows current to flow in reverse when voltage reaches a certain level. It is used to protect circuits from high voltage.

Zener diode keeps the output voltage constant even if the input voltage changes. It is used in power supply circuits to give steady voltage to microcontroller.

• **MOSFET**: A MOSFET (Metal-Oxide-Semiconductor field-Effect transistor) is an electronic component used to control current in circuits.

• **MOSFET as a switch**: When a MOSFET is used as a switch, the Gate pin controls if electricity can flow between the Drain and source pin.

if Gate = HIGH (Ex: 5V) → ON → current flow

if Gate = Low (Ex: 0V) → OFF → no current.

• **MOSFET as a Amplifier**:

→ In this case, mosfet use in linear (Active) mode

→ The ammount of current flow depends on the Gate voltage

→ for amplification, it receive low signal

→ Control a large current this low signal from Drain to source

→ output strong signal taken from Drain.

CEP Problem Using Arduino

CEP Problem Based on Embedded System and IoT

Q1.

How would you design a smart irrigation system for an agricultural field using appropriate microcontrollers, sensors, and actuators to monitor soil moisture levels, weather conditions, and plant growth? The system should automatically regulate water flow and trigger alerts when critical thresholds are crossed, ensuring optimal water usage and plant health. Develop a detailed schematic of the system and provide a comprehensive list of required components.

CEP Problem Based on Embedded System and IoT

Solution:

Problem Overview:

The goal is to design a smart irrigation system that monitors soil moisture levels, weather conditions, and plant growth, automatically regulating water flow to ensure optimal water usage and plant health. The system will also trigger alerts when critical thresholds are crossed. Below is a detailed solution covering both hardware and software components.

CEP Problem Based on Embedded System and IoT

Solution:

1. System Requirements

The system should be capable of:

- **Monitoring soil moisture** to determine when watering is required.
- **Monitoring weather conditions** (rain, temperature, humidity) to adjust irrigation based on environmental conditions.
- **Monitoring plant growth** using relevant sensors to optimize watering based on plant maturity or health.
- **Controlling water flow** using solenoid valves to open or close irrigation lines.
- **Sending alerts** if soil moisture drops below a critical level or if weather conditions suggest extreme dryness or flooding.
- **Logging data** to cloud storage for analysis and long-term optimization of irrigation patterns.

CEP Problem Based on Embedded System and IoT

Solution:

2. Components Required

Microcontroller:

- **ESP32 or Arduino**: A microcontroller with built-in Wi-Fi for remote monitoring and control. The ESP32 is recommended for its better processing power, dual-core functionality, and built-in wireless capabilities.

Sensors:

- **Soil Moisture Sensor (Capacitive Type)**: Measures soil moisture to determine when to irrigate. Example: Capacitive Soil Moisture Sensor (YL-69).

- **Temperature and Humidity Sensor (DHT22 or SHT31)**: Monitors the ambient temperature and humidity to optimize irrigation based on weather conditions.

CEP Problem Based on Embedded System and IoT

Solution:

- **Rain Sensor**: Detects rainfall to prevent unnecessary watering.
- **Ultrasonic Distance Sensor** (optional): Monitors plant height/growth or could be replaced by more advanced sensors like NDVI (Normalized Difference Vegetation Index) sensors for plant health.
- **Weather Station** (optional): Integrates wind speed, rainfall, and UV sensors to optimize irrigation during extreme weather.

Actuators:

- **Solenoid Valve**: Controls the water flow in the irrigation system. It opens or closes based on signals from the microcontroller.
- **Relay Module**: Acts as a switch for high-power devices (solenoid valves or water pumps).

CEP Problem Based on Embedded System and IoT

Solution:

Power Supply:

- **12V or 24V DC Power Supply**: Powers the solenoid valves and the entire system. Solar power could be used for energy efficiency in remote fields.

- **5V Step-Down Converter**: To provide power to the microcontroller and sensors from the 12V/24V supply.

Communication:

- **Wi-Fi or LoRa Module**: Provides remote communication for data logging and alerting. ESP32 has built-in Wi-Fi, while LoRa is suitable for long-range applications.

Cloud Platform:

- **ThingSpeak/AWS IoT/MQTT Protocol**: For remote data logging and real-time alerting. The microcontroller sends sensor data to the cloud for analysis and triggers notifications via the cloud platform if required.

CEP Problem Based on Embedded System and IoT

<u>**Solution:**</u>

Power Supply:

- **12V or 24V DC Power Supply**: Powers the solenoid valves and the entire system. Solar power could be used for energy efficiency in remote fields.

- **5V Step-Down Converter**: To provide power to the microcontroller and sensors from the 12V/24V supply.

Communication:

- **Wi-Fi or LoRa Module**: Provides remote communication for data logging and alerting. ESP32 has built-in Wi-Fi, while LoRa is suitable for long-range applications.

Cloud Platform:

- **ThingSpeak/AWS IoT/MQTT Protocol**: For remote data logging and real-time alerting. The microcontroller sends sensor data to the cloud for analysis and triggers notifications via the cloud platform if required.

CEP Problem Based on Embedded System and IoT

Solution:

3. System Design

Main System Components:

1. **ESP32 Microcontroller** is the central brain that collects data from various sensors, processes it, and controls the solenoid valves via relays.
2. **Sensors**:
    - The **Soil Moisture Sensor** is placed near the plants and connected to the microcontroller’s analog input to continuously monitor soil water content.
    - The **Temperature and Humidity Sensor (DHT22)** and **Rain Sensor** are placed in open-air locations to monitor environmental conditions.

CEP Problem Based on Embedded System and IoT

Solution:

1. **Water Flow Control**: The solenoid valves are connected to water pipes that distribute irrigation. The microcontroller, using a **Relay Module**, controls these valves based on sensor data.

2. **Power Supply**: A DC power supply unit is used to power the solenoid valves and sensors. The microcontroller is powered through a step-down converter.

CEP Problem Based on Embedded System and IoT

Solution:

4. Software Design

The software needs to perform the following tasks:

- **Read sensor data** for soil moisture, temperature, humidity, and rain status.
- **Process the sensor data** and decide when to water the plants.
- **Control solenoid valves** based on sensor thresholds and weather conditions.
- **Send alerts** if any parameters (like low soil moisture) exceed the predefined thresholds.
- **Log data to the cloud** for analysis and future optimization.

CEP Problem Based on Embedded System and IoT

Solution:

4.1 System Flow

1. **Sensor Data Collection**:
    - Soil moisture is measured in real-time.
    - Temperature, humidity, and rain sensors provide context for adjusting watering schedules based on environmental conditions.
2. **Decision Making**:
    - If the soil moisture falls below a certain threshold, and the weather sensors indicate no rain, the system opens the solenoid valves to irrigate the field.
    - If it's raining or soil moisture is adequate, irrigation is skipped.

CEP Problem Based on Embedded System and IoT

Solution:

3. Water Control:

- The relay module is triggered to open/close the solenoid valve for watering.
- Valves are automatically closed once soil moisture reaches a safe level.

4. Data Logging and Alerts:

- Data is logged to the cloud using an IoT platform (like ThingSpeak or AWS IoT).
- Alerts (via email, SMS, or app notification) are triggered if soil moisture or temperature reaches critical levels.

4.2 Pseudo Code

CEP Problem Based on Embedded System and IoT

Solution:

5. Conclusion

This smart irrigation system integrates sensors, actuators, and cloud connectivity to manage irrigation in an automated and efficient way. By monitoring soil moisture, weather conditions, and plant growth, the system optimizes water usage while ensuring plant health. Real-time alerts and data logging to the cloud make it robust, allowing remote monitoring and adjustments.

This system demonstrates how modern agricultural practices can be enhanced using IoT, real-time data, and automation to conserve water, improve crop yield, and reduce manual labor.

CEP Problem Based on Embedded System and IoT

Q2.

How would you design a smart energy management system for a commercial building using appropriate microcontrollers, sensors, and actuators to monitor and optimize energy consumption? The system should track occupancy, lighting, HVAC (Heating, Ventilation, and Air Conditioning) usage, and power loads from various appliances, automatically adjusting based on occupancy and time of day. It should also trigger alerts when energy usage exceeds certain limits or when equipment failures are detected. Develop a detailed schematic of the system and provide a comprehensive list of required components