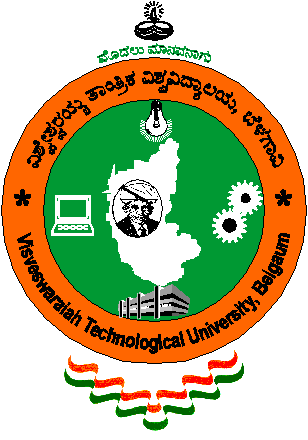
**INTERNSHIP - I**

******(Inter / Intra Institutional Internship)**

**An**

**Internship Report Submitted To**

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**Belagavi, Karnataka**

*For The Award of Degree*

***Bachelor of Engineering***

By

**Miss. Dhanashree Patil**

**USN: 2JI22EC402**

Guide

**Prof. Veeresh Sawant**

Professor

Department of Electronics and communication Engineering

Jain College of Engineering, Belagavi



**Department of Electronics and Communication Engineering**

**JAIN COLLEGE OF ENGINEERING (JCE)**

**Belagavi, Karnataka – 590008**

**2022-2023**

**JGI’s**

**JAIN COLLEGE OF ENGINEERING (JCE)**

**BELAGAVI**



**DEPARTMENT OF ELECTRONICS AND COMMUNICATON ENGINEERING**

**Certificate**

**Certified that Miss. Dhanashree Patil,** anundergraduate student bearing **USN 2JI22EC402** has satisfactorily completed the **Internship – I** on **“AIPS – ARDUINO PROGRAMMING, INTRODUCTION TO IOT USING RASPBERRY PI, 3D-PRINTING AND SPREADSHEETS”,** submitted to **Visvesvaraya Technological University, Belagavi** in partial fulfillment for the award of **Bachelors** in **Electronics and Communication Engineering.**

**Guide Internship Co-ordinator**

**HOD Principal & Director**

**Visvesvaraya Technological University, Belagavi**

****

**CERTIFICATE**

**Certified that the Internship - I**

**“AIPS – ARDUINO PROGRAMMING, INTRODUCTION TO IOT USING RASPBERRY PI, 3D-PRINTING AND SPREADSHEETS”**

is a bonafide work carried out by

**Mss. Dhanashree Patil**

**USN: 2JI22EC402**

*In partial fulfillment for the award of* **BACHELORS IN ELECTRONICS AND COMMUNICATION ENGINEERING** *of the* ***Visvesvaraya Technological University, Belagavi.*** *The report has been approved as it satisfies the academic requirements in respect of Internship Work prescribed for the said degree.*

**Name of the Examiners Signature with date**

**Internship Certificate**

**Acknowledgements**

I am grateful and would like to express my sincere gratitude to guide Prof. Vinayak Sawant for his untiring guidance, continuous encouragement and constant support which made this Internship study possible. Sir has impressed me with his outstanding professional conduct and his strong conviction towards the Internship study. I am truly obliged for his progressive vision throughout the internship.

I would also like to thank Principal and Director of Jain College of Engineering, Belagavi Dr. J. Shivakumar, HOD Mechanical Engineering, Dr. B. V. H5ubballi, Faculty of Mechanical Engineering Department and Management of Jain College of Engineering, Belagavi, for their wisdom and valuable suggestions during my internship work. I extend my gratitude to the, Internship Co-Ordinator and Staff of Jain College of Engineering, Belagavi, for their support and cooperation during the study.

I finally acknowledge my sincere indebtedness and gratitude to my parents and family for their constant support and encouragement throughout the Internship.

Dhanashree Patil

**TABLE OF CONTENTS**

**CONTENTS Page. No**

|  |  |
| --- | --- |
| Acknowledgement |  |
| Abstract | i |
| Table of Contents | ii |
| List of Figures | v |
| List of Tables | xii |
| List of Nomenclature | xvi |
| List of Abbreviations | xvii |

**Chapter 1: Introduction 1-9**

1.1: Introduction 1

1.2: Internship study 3

1.3: Aims and Objectives 7

1.4: Report Outline 9

**Chapter 2: Literature Review 10-11**

2.1: Arduino Programming 10

2.2: Introduction to IOT using Raspberry Pi 10

2.3: 3D- Printing 11

2.4: Microsoft Excel and Spreadsheets 11

2.5 Concluding Remarks 11

**Chapter 3: Arduino Programming**  **12-29**

3.1 Introduction to Embedded System 12

3.1.1 Embedded system Introduction 13

3.1.2 Microcontroller 14

3.1.3 Input and output devices 14

3.1.4 Arduino Uno Board 15

3.1.5 Open-source Hardware and software 16

3.2 Tinkercad Online Simulator 17

3.3 Arduino Uno Programming and Simulation 18

3.3.1 Program to blink a LED and a Buzzer simultaneously 18

and continuously

3.3.2 ON & OFF all LEDs in sequence 18

3.3.3 Traffic Light controller one side with 3 LEDs 19

3.4 Serial Communication Interface 20

3.4.1 Functions 21

3.4.2 Message display on Serial monitor “Hello World! 22

3.5 Potentiometer Interface as Analog Input and LED Blinking 23

3.6 Arduino Uno Interface with LDR (light dependent resistor/photo resistor) 24

3.6.1 Observe the sensor Value on Serial Monitor and on the LED 24

3.7 Arduino with Ultrasonic Sensor 25

3.7.1 Distance of object to display on Serial Monitor 25

3.7.2 Distance is less than defined value ON led else OFF Led 25

3.8 Arduino Uno Interface with LCD Display 26

3.8.1 Display “Hello World!” on LCD. 27

3.8.2 Custom Character Display on LCD 27

3.8.3 Program for scrolling Numbers 28

3.8.4 Program for scrolling Characters 28

3.8.5 Program for displaying character on 29

LCD sent from serial monitor

3.9 Module Outcome

**Chapter 4: Introduction to IOT using Raspberry Pi**   **30-36** 4.1: Introduction to Raspberry Pi 64.2 IOT Components 30

4.2.1 LED 31

4.2.2 Resistor 31

4.2.3 Jumper Wires 31

4.2.4 PIR (Motion Sensors) 31

4.2.5 Gas Sensors (MQ – 2) 31

4.2.6 Button Switches 32

4.2.7 Buzzers 32

4.2.8 Pi Camera Module Interface with Raspberry Pi using Python 32

4. 3 Introduction to Python 33

4.4 Program to Flash an LED at a given on time and off time cycle, 33

where the two times are taken from a file

4.5 Program to Get Input from Two Switches and Switch on Corresponding LEDs 34

4.6 Program to Access an Image Through a Pi Web Cam 34

4.7 Program to Get an Alarm from a Remote Area (Through LAN) 35

if Smoke is Detected

4.8 Module Outcome 36

**Chapter 5: 3D- Printing 37-40**

5.1 3D Printing 37

5.2 Technologies 37

5.3 Applications 38

5.4 modeling software 39

5.5 Module Outcome 40

**Chapter 6: Microsoft Excel and Spreadsheets 41-44**

6.1: Microsoft PowerPoint 41

6.2: Microsoft Word 42

6.3: Microsoft Excel 42

6.4: Outcome 43

**Chapter 7: Conclusion and Future Scope 44**

7.1: Conclusions 44

7.2: Future Scope 44

**CHAPTER 1: *Introduction***

## 1.1 Introduction:

During my internship, I had the opportunity to dive into Arduino programming, IoT using Raspberry Pi, 3D printing, and Microsoft Excel. In Arduino programming, I gained proficiency by learning how to connect various components, such as sensors, actuators, and displays, to an Arduino board and write code to control them. This involved understanding the syntax and structure of the Arduino programming language, as well as utilizing the extensive library of functions and examples provided by Arduino.

These experiences during my internship provided me with a solid foundation in Arduino programming, IoT using Raspberry Pi, 3D printing, and Microsoft Excel. I gained practical skills, problem-solving abilities, and a deeper understanding of the applications and potential of these technologies. This equipped me with valuable technical skills that will undoubtedly contribute to my future endeavors in the field of electronics, IoT, and data analysis.

# **1.2: Internship study:**

# Module 1: Arduino Programming

Module 2: Introduction to IOT using Raspberry Pi

Module 3: 3D Printing

Module 4: Spread Sheets for Beginners and UHV

## 1.3: Aims and Objectives:

The aim of this internship experience was to gain proficiency in Arduino programming, IoT using Raspberry Pi, 3D printing, and Microsoft Excel.

**Objective:**

1. Gain proficiency in Arduino programming, IoT using Raspberry Pi, 3D printing, and Microsoft Excel.
2. Develop practical skills in connecting components, writing code, and setting up Raspberry Pi for IoT projects.
3. Acquire hands-on experience in designing and printing 3D models using CAD software and 3D printers.
4. Enhance data analysis and manipulation skills in Microsoft Excel.
5. Apply acquired skills in real-world projects to develop problem-solving abilities.
6. Develop a comprehensive understanding of the applications and potential of these technologies for future career growth.

**1.4: Report Outline:**

* This report consists of seven chapters
* Chapter 1: Introduction
* Chapter 2: Literature Review
* Chapter 3: Arduino Programming
* Chapter 4: Introduction to IOT using Raspberry Pi
* Chapter 5: 3D - Printing
* Chapter 6: Microsoft Office
* Chapter 7: Conclusion & Future Scope

**CHAPTER 2: *Literature Review***

**2.1: Arduino programing:**

Many studies have focused on the educational benefits of Arduino programming, particularly in STEM education. These studies highlight how Arduino promotes hands-on learning, problem-solving skills, and creativity among students of different age groups. Arduino's ease of use and accessibility have made it a popular choice for introducing programming and electro

In terms of applications, the literature covers a wide range of projects and domains where Arduino programming is utilized. These include home automation, robotics, environmental monitoring, medical devices, and more. Arduino's versatility and open-source nature have made it a preferred platform for prototyping and developing low-cost solutions in various fields.

The literature also discusses programming techniques and best practices for Arduino. This includes topics such as coding standards, efficient memory management, using libraries, and optimizing power consumption. Additionally, there are resources available that provide tutorials and examples for beginners to learn and apply Arduino programming concepts.nics to beginners.

# **2.2: Introduction to IOT using Raspberry Pi:**

## A literature survey on IoT using Raspberry Pi reveals its versatility and cost-effectiveness for developing IoT applications. Studies explore its use in smart homes, healthcare, agriculture, and industrial automation. The literature covers topics such as communication protocols, security, data analytics, and educational applications. Future directions include edge computing and integration with emerging technologies. Overall, Raspberry Pi offers immense potential in building practical IoT solutions and has sparked significant research and development in the field.

## 2.3: 3D-printing:

## A literature survey on 3D printing reveals its wide range of applications across industries. Research covers materials, printing techniques, design considerations, and post-processing. 3D printing has revolutionized manufacturing, enabling customization and complex geometries. The literature showcases its potential for creating prototypes, personalized products, and advancing various fields such as healthcare and aerospace.

## 2.4: Microsoft Office:

## A literature survey on Microsoft Office highlights the use of Microsoft Word for document creation and collaboration, and Microsoft Excel for data analysis and visualization. Research explores features such as templates, styles, formulas, and pivot tables, emphasizing their significance in professional tasks.

## 2.5: Concluding Remarks:

This chapter has the literature review of topic. The next chapters will conclude the topic

**CHAPTER 3: *Arduino Programming***



## 3.1: Introduction:

Arduino programming is a key component of the Arduino platform, an open-source electronics platform designed for building interactive projects. Arduino boards are equipped with microcontrollers that can be programmed to control and interact with various electronic components such as sensors, motors, and displays.

Arduino programming utilizes a simplified version of the C/C++ programming language, making it accessible to beginners and experienced programmers alike. It offers a wide range of pre-built libraries and functions that facilitate the development process by abstracting complex tasks.

With Arduino programming, users can create projects that respond to inputs, collect data, and perform specific actions. This can range from simple tasks like blinking an LED or reading sensor data to more complex projects such as robotics, home automation, or IoT applications.

Arduino's versatility, ease of use, and extensive online community support have made it a popular choice for hobbyists, students, and professionals. Its open-source nature encourages collaboration, allowing users to share and learn from each other's projects and code.

By learning Arduino programming, individuals can gain valuable skills in electronics, coding, and project development. It provides a solid foundation for exploring the world of physical computing and empowers users to bring their creative ideas to life.

**3.1.1 Embedded System:**

Embedded systems are specialized computer systems designed to perform specific tasks or functions within a larger system. Unlike general-purpose computers, embedded systems are dedicated to executing specific functions with specific requirements.

Embedded systems are found in various everyday devices, ranging from household appliances and mobile phones to automotive systems and industrial machinery. They are designed to provide intelligence, control, and connectivity to these devices, enabling them to perform their intended functions efficiently and reliably.

Embedded systems typically consist of three essential components: a microcontroller or microprocessor, software, and external components. The microcontroller is the core of the system, responsible for executing the program instructions and controlling the device's operation. The software, specifically tailored for the embedded system, determines its behavior and functionality. External components such as sensors, actuators, and communication modules enable interaction with the surrounding environment.

The design and development of embedded systems require a deep understanding of hardware, software, and system integration. Low power consumption, real-time response, and reliability are crucial considerations in embedded system design. As technology advances, embedded systems are becoming increasingly interconnected through network interfaces, leading to the emergence of the Internet of Things (IoT) and the integration of embedded systems into a larger networked ecosystem.

Embedded systems programming often involves working with programming languages such as C/C++, as well as specialized tools and development environments. Developers must possess a solid understanding of hardware architectures, communication protocols, and system-level design principles.

With their wide range of applications and increasing integration into our daily lives, embedded systems play a vital role in various industries, including automotive, healthcare, aerospace, and consumer electronics. They enable automation, improve efficiency, and enhance the functionality and performance of countless devices and systems.

In conclusion, embedded systems are the backbone of numerous electronic devices, providing dedicated computing power and functionality. They offer customized solutions tailored to specific requirements, ensuring reliable and efficient operation. The field of embedded systems continues to advance, driving innovation and shaping the technologies that shape our modern world.

**3.1.2: Microcontroller:**

A microcontroller is a small computer on a single integrated circuit (IC) that is designed to perform specific tasks within an embedded system. It consists of a microprocessor core, memory (both volatile and non-volatile), input/output (I/O) peripherals, and various communication interfaces. Microcontrollers are commonly used in embedded systems where they provide intelligence and control to devices or machines.

The microprocessor core of a microcontroller executes the program instructions stored in its memory, which can be either flash memory or read-only memory (ROM). The program instructions determine the behavior and functionality of the embedded system.

Microcontrollers are designed to interface with external devices through various I/O peripherals such as digital input/output (GPIO) pins, analog-to-digital converters (ADC), timers, and serial communication interfaces (UART, SPI, I2C). These peripherals allow the microcontroller to interact with sensors, actuators, displays, and other external components.

The choice of microcontroller depends on the specific requirements of the embedded system, such as processing power, memory capacity, I/O capabilities, power consumption, and cost. Popular microcontroller families include Atmel AVR, Microchip PIC, ARM Cortex-M, and Texas Instruments MSP430.

Programming microcontrollers typically involves writing code in low-level languages such as C or assembly language. Integrated development environments (IDEs) and compilers specifically tailored for the target microcontroller simplify the development process.

Microcontrollers are widely used in various applications, including home automation, robotics, medical devices, automotive systems, and industrial control. Their small size, low power consumption, and cost-effectiveness make them ideal for embedded system designs where space and power constraints are present.

**3.1.3: Input and Output Devices:**

Arduino boards are designed to interface with various input and output (I/O) devices, allowing users to interact with the physical world and control external components. Here are some commonly used input and output devices with Arduino:

Input Devices:

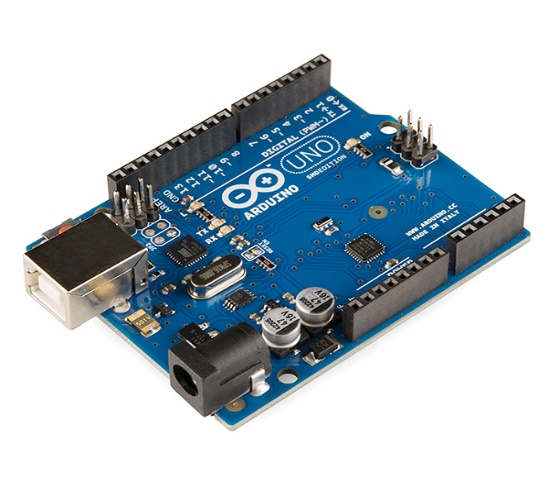
1. Push Buttons: These are simple mechanical switches that can be used as digital input devices, detecting when they are pressed or released.
2. Potentiometers: Potentiometers are variable resistors that allow users to provide analog input by adjusting the position of a knob or lever.
3. Sensors: Arduino can interface with a wide range of sensors such as temperature sensors, light sensors, humidity sensors, motion sensors, and more, enabling the collection of data from the environment.
4. Keypads: Keypads provide a set of buttons arranged in a grid, allowing users to input characters or numbers.
5. Switches: Switches can be used as digital input devices to detect the state of a specific condition or toggle between two states.

Output Devices:

1. LEDs: Light Emitting Diodes (LEDs) are commonly used as visual indicators or to provide simple output signals.
2. LCD Displays: Liquid Crystal Displays (LCDs) allow for text and graphic output, providing a visual interface to communicate information.
3. Motors: Arduino can control different types of motors, including DC motors, servo motors, and stepper motors, enabling the movement of physical objects.
4. Speakers/Buzzers: These devices produce sound output and can be used for audio alerts, melodies, or generating tones.
5. Relays: Relays are electromechanical switches that can be used to control high-power devices or to switch on/off external circuits.

These are just a few examples of the input and output devices that can be used with Arduino. The flexibility and versatility of Arduino allow for the connection and control of a wide range of other devices and components based on specific project requirements.

**3.1.4: Arduino Uno Board:**



The Arduino Uno board is a versatile microcontroller board based on the ATmega328P. It offers 14 digital I/O pins, 6 analog input pins, USB interface for programming, and compatibility with various expansion modules. It is widely used for prototyping and DIY projects, making it popular among beginners and enthusiasts.

The Arduino Uno board finds applications in various projects and domains. Here are some common applications:

* **Robotics**
* **Home Automation**
* **IoT (Internet of Things)**
* **Prototyping**
* **Education**
* **Environmental Monitoring**
* **Automation Systems**
* **Data Logging**

## 3.1.5: INTRODUCTION TO BREADBOARD:

## Breadboard Images – Browse 9,934 Stock Photos, Vectors, and Video | Adobe Stock

A breadboard is a plastic board with a grid of interconnected holes used for prototyping and testing circuits without soldering. Components can be inserted into the holes and connected using jumper wires, allowing for quick and temporary circuit connections. It provides a flexible and versatile platform for experimenting with circuit designs before creating permanent versions. Breadboards are widely used by electronics enthusiasts and professionals for rapid prototyping and testing of circuit ideas.

* Using a breadboard is a simple and straightforward process:

1. Insert Components: Insert the electronic components, such as resistors, capacitors, and integrated circuits (ICs), into the breadboard's holes. Make sure the component leads are aligned with the appropriate rows or columns.
2. Create Connections: Use jumper wires to establish connections between components. Insert one end of the wire into a hole with a desired component lead, and the other end into another hole to connect to another component or the power rails.
3. Utilize Power Rails: The breadboard typically has two rows called power rails along the sides. Connect the power source, such as a battery or power supply, to the appropriate rails. The positive voltage (usually 5V or 3.3V) connects to the rail on the same side as the red line, and the ground (GND) connects to the rail on the same side as the blue or black line.
4. Test and Iterate: Once the connections are made, apply power to the breadboard. Test the circuit's functionality and make any necessary modifications by rearranging or replacing components.

**3.2: Tinker cad Online Simulator:**

Tinkercad is an online simulator and design tool that allows users to create, simulate, and prototype electronic circuits and 3D designs. It provides a user-friendly interface and a range of features to support learning, experimentation, and project development.

* Key features of Tinkercad Online Simulator:

1. Circuit Design: Tinkercad offers a virtual workspace where users can design electronic circuits using a wide range of components, including resistors, capacitors, LEDs, sensors, and microcontrollers. Components can be easily placed and connected on a virtual breadboard or through a drag-and-drop interface.
2. Simulation: Tinkercad allows users to simulate their circuits in real-time to test their functionality and behavior. This feature helps in identifying and troubleshooting issues before building physical prototypes.
3. Code Development: Tinkercad supports programming through a block-based coding environment or Arduino code. Users can write code to control the behavior of their circuits and interact with components.
4. 3D Design: In addition to circuit design, Tinkercad offers 3D design capabilities. Users can create and modify 3D models using a variety of tools and shapes, enabling them to design enclosures, housings, or other mechanical components for their projects.
5. Collaboration and Sharing: Tinkercad allows users to collaborate with others by sharing their designs and projects. This feature promotes teamwork, knowledge sharing, and feedback exchange among users.
6. Educational Resources: Tinkercad provides tutorials, projects, and lesson plans to support learning and inspire creativity. It is widely used in educational settings to teach electronics, coding, and 3D design.

## 3.3 Arduino Uno Programming and Simulation:

Arduino Uno programming and simulation involves writing code for the board and testing it virtually. The Arduino IDE and online simulators like Tinkercad offer platforms for code development and testing. Users can verify code functionality and behavior before deploying it on the physical board. These tools streamline the development and debugging processes for Arduino Uno projects.

They provide a virtual environment for code verification, saving time and effort. Arduino programming allows control of various components like motors, sensors, and actuators. Simulation helps ensure correct circuit behavior. Code can be written in the Arduino IDE and tested in a virtual environment before uploading to the Arduino Uno board. Online simulators offer convenient prototyping and testing without physical components. Arduino Uno programming and simulation enhance efficiency and accuracy in project development.

**3.3.1 Program to blink a LED and a Buzzer simultaneously** **and continuously:**

void setup() {

pinMode(13, OUTPUT);

pinMode(5, OUTPUT);

}

// the loop function runs over and over again forever

void loop() {

digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)

digitalWrite(5,LOW);

delay(1000); // wait for a second

digitalWrite(13, LOW); // turn the LED off by making the voltage LOW

digitalWrite(5,HIGH);

delay(1000); // wait for a second

}

**3.3.2 ON & OFF all LEDs in sequence:**

Void setup() {

pinMode(7, OUTPUT);// LED1

pinMode(8, OUTPUT);// LED2

pinMode(9, OUTPUT);// LED3

}

void loop() {

digitalWrite(7, HIGH); digitalWrite(8, LOW); digitalWrite(9, LOW); delay(1000);

digitalWrite(7, HIGH); digitalWrite(8, HIGH); digitalWrite(9, LOW); delay(1000); digitalWrite(7, HIGH); digitalWrite(8, HIGH); digitalWrite(9, HIGH); delay(1000); digitalWrite(7, LOW); digitalWrite(8, HIGH); digitalWrite(9, HIGH); delay(1000); digitalWrite(7, LOW); digitalWrite(8, LOW); digitalWrite(9, HIGH); delay(1000); digitalWrite(7, LOW); digitalWrite(8, LOW); digitalWrite(9, LOW); delay(1000);

}

**3.3.3 Traffic Light controller one side with 3 LEDs:**

void setup() {

// configure the output pins

pinMode(2,OUTPUT);

pinMode(3,OUTPUT);

pinMode(4,OUTPUT);

pinMode(5,OUTPUT);

pinMode(6,OUTPUT);

pinMode(7,OUTPUT);

pinMode(8,OUTPUT);

pinMode(9,OUTPUT);

pinMode(10,OUTPUT);

void loop()

digitalWrite(2,1); //enables the 1st set of signals

digitalWrite(7,1);

digitatWrite(10,1);

digitalWrite(4,0);

digitalWrite(6,0):

digitalWrite(3,0);

digitalWrite(8,0);

digitalWrite(9,0);

digitalWrite(5,0);

delay(5000);

digitalWrite(3,1); //enables the yellow lights

digitalWrite(6,1);

digitalWrite(2,0);

digitalWrite(7.0);

delay(1000);

digitalWrite(10,1)

digitalWrite(4,1); //enables the 2nd set of signals

digitalWrite(5,1):

digitalWrite(2,0);

digitalWrite(6,0);

digitalWrite(7.0)

digitalWrite(3,0);

digitalWrite(8,0);

digitalWrite(9,0);

delay(5000);

digitalWrite(9,1); //enables the yellow lights

digitalWrite(6,1);

digitalWrite(10,0);

digitalWrite(5,0);

digitalWrite(4,0);

delay(1000);

digitalWrite(8,1); //enables the 3rd set of signals

digitalWrite(4,1);

digitalWrite(7,1);

digitalWrite(2,0);

digitalWrite(3,0);

digitalWrite(5,0);

digitalWrite(6,0);

digitalWrite(9,0):

digitalWrite(10,0);

delay(5000);

digitalWrite(9,1); //enables the yellow lights

digitalWrite(3,1);

digitalWrite(7,0);

digitalWrite(8,0);

digitalWrite(4,0);

delay(1000);

## 3.4 Serial Communication Interface

void setup() {

Serial.begin(9600);

}

void loop(){

Serial.print(“Hello World”); delay(1000);

}

void setup() {

Serial.begin(9600);

}

void loop() {

Serial.println(“Hello World”);

delay(1000);

## 3.5 Potentiometer Interface as Analog Input and LED Blinking:

## PROGRAM FOR LED BLINKING USING POTENTIOMETER

## int SensorValue=0;

## void setup() {

## Serial.begin(9600);

## pinMode( 13, OUTPUT);

## }

## void loop() {

## SensorValue = analogRead(A0);

## Serial.print(SensorValue);

## digitalWrite(13,HIGH);

## delay(SensorValue);

## digitalWrite(13,LOW);

## delay(SensorValue);

## }

**3.6 Arduino Uno Interface with LDR (light dependent resistor/photo resistor):**

**INTRODUCTION TO LIGHT DEPENDENT RESISTOR**

LDR, which stands for Light Dependent Resistor, is a passive electronic component that exhibits changes in its electrical resistance based on the intensity of light falling on its surface. It is also commonly referred to as a photoresistor or photocell. LDRs are widely used in various applications where light detection or measurement is required.

The structure of an LDR typically consists of a semiconductor material, such as cadmium sulfide (CdS), that undergoes changes in conductivity when exposed to light. When light falls on the LDR, the photons absorbed by the semiconductor material cause the electrons to move from the valence band to the conduction band, resulting in a decrease in resistance.

The resistance of an LDR can range from several kilohms in bright light conditions to several megohms or higher in darkness. This property makes LDRs highly useful for detecting and measuring light levels in a wide range of applications. They are particularly popular in automatic light control systems, photography equipment, burglar alarm systems.

**Analog input code Read value in serial monitor**

void setup(){

Serial.begin(9600);

}

void loop(){

int sensorValue = analogRead(A0); Serial.println(sensorValue); delay(1);

}

void loop(){

SensorValue=analogRead(A0); Serial.print(SensorValue); digitalWrite(13,HIGH); delay(SensorValue);

digitalWrite(13,LOW);

delay(SensorValue);

}

int input=A0;

int LED=10;

int sensorvalue=0;

void setup(){

Serial.begin(9600);

pinMode (LED, OUTPUT);

}

void loop() {

sensorvalue=analogRead(input); Serial.println("Sensor value ="); Serial.println(sensorvalue);

if(sensorvalue > 512)

{

digitalWrite(LED,HIGH);

}

else

{

Digital write(LED LOW);

}

delay(50);

}

**INTRODUCTION TO ULTRASONIC SENSOR:**

An ultrasonic sensor is a device that uses ultrasonic sound waves to detect and measure distance to objects. It is a type of non-contact proximity sensor that emits high-frequency sound waves and analyses the echoes reflected from nearby objects.

The sensor consists of two main components: a transducer and a receiver. The transducer emits ultrasonic waves, usually in the range of 20 kHz to 200 kHz, which are inaudible to humans. These waves travel through the air and when they encounter an object, they bounce back as echoes. The receiver then detects these echoes and measures the time it takes for the waves to return. Overall, ultrasonic sensors offer reliable and versatile solutions for detecting and measuring distances, making them popular in various industries and applications.

**3.7 Arduino with Ultrasonic Sensor**:

/\*Ultrasonic Sensor HC-SR04 and Arduino Tutorial\*/

// defines pins numbers

const int trigPin = 9;

const int echoPin = 10;

// defines variables

long duration;

int distance;

void setup() {

pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output

pinMode(echoPin, INPUT); // Sets the echoPin as an Input

Serial.begin(9600); // Starts the serial communication

}

void loop(){

// Clears the trigPin

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

// Sets the trigPin on HIGH state for 10 micro seconds

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

// Reads the echoPin, returns the sound wave travel time in microseconds

duration = pulseIn(echoPin, HIGH);

// Calculating the distance

distance= duration\*0.034/2;

// Prints the distance on the Serial Monitor

Serial.print("Distance: ");

Serial.println(distance);

delay(1000);

}

**3.8 Arduino Uno Interface with LCD Display:**

An LCD (Liquid Crystal Display) is a flat panel display technology that uses liquid crystals to display images or text. LCD displays are commonly used in electronic devices such as televisions, computer monitors, smartphones, and, in this case, with Arduino microcontrollers.

There are different types of LCD displays, but one commonly used with Arduino is the character LCD display. Character LCD displays come in various sizes, such as 16x2 or 20x4, indicating the number of characters they can display per line and the number of lines.

Character LCD displays are controlled using parallel or serial communication protocols. The parallel communication method requires several pins on the Arduino, while the serial communication method requires fewer pins but requires an I2C or SPI interface.

To interface an Arduino with an LCD display, you typically need to connect power (VCC and GND), data (data pins or I2C/SPI interface), and control signals (such as enable, register select, and read/write pins). Additionally, some LCD displays have backlight control pins.

**3.8.1 Display “Hello World!” on LCD:**

Void setup() {

Serial.begin(9600);

Serial.println(“\*\*\*\*\*\*\*\*\*\*\*\*\*\*Hello JCE!\*\*\*\*\*\*\*\*\*\*\*\*\*\*”);

Serial.println(“\*\*\*\*Welcome to Arduino Programming\*\*\*\*”);

}

void loop(){

}

### **3.8.2 Custom Character Display on LCD:**

#include <LiquidCrystal.h>

const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;

LiquidCrystal lcd ( rs,en,d4,d5,d6,d7);

byte custom[8] = {

B00001,

B00010,

B00100,

B01000,

B10000,

B01000,

B00100,

B00010};

void setup() {

lcd.begin(16,2);

lcd.clear();

lcd.createChar(1,custom);

lcd.setCursor(0,0);

lcd.print("CUSTOM!");

delay(3000);

}

void loop() {

for(int i=0;i<=15;i++)

{

lcd.setCursor(i,1);

lcd.write(1);

delay(1000);

}

}

### **3.8.3 Program for scrolling Numbers:**

#include <LiquidCrystal.h>

const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;

LiquidCrystal lcd (rs, en, d4, d5, d6, d7);

void setup() {

lcd.begin(16, 2);

}

void loop() {

lcd.setCursor(0, 0);

// print from 0 to 9:

for (int thisChar = 0; thisChar < 10; thisChar++) {

lcd.print(thisChar);

delay(500);

{

lcd.setCursor(16, 1);

// set the display to automatically scroll:

lcd.autoscroll();

print from 0 to 9:

for (int thisChar = 0; thisChar < 10; thisChar++) {

lcd.print(thisChar);

delay(500);

}

// turn off automatic scrolling

lcd.noAutoscroll();

// clear screen for the next loop:

lcd.clear();

}

**3.8.4 Program for scrolling Characters:**

#include <LiquidCrystal.h>

const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2; LiquidCrystal lcd (  rs,en,d4,d5,d6,d7); byte custom[8] ={

B00001,

B00010,

B00100,

B01000,

B10000,

B01000,

B00100, B00010};

void setup() {

lcd.begin(16,2);

lcd.clear();

lcd.createChar(1,custom);

lcd.setCursor(0,0);

lcd.print("CUSTOM!");

delay(3000);

}

void loop() {

for(int i=0;i<=15;i++)

 {

lcd.setCursor(i,1);

lcd.write(1);

delay(1000);

}

**3.8.5 Program for displaying character on LCD sent from serial monitor:**

#include <LiquidCrystal.h>//Library for LCD

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);//LCD pins at which it is attached to the Arudino

void setup()//method used to run the code for once

{

lcd.begin(16, 2);//LCD order

lcd.print("                  ");//prints on LCD

// lcd.setCursor(0,1);//Setting the cursor on LCD

//lcd.print("");//prints on LCD

delay(500);//delay of 1 second

}

void loop() //used to run the code repeatedly

{

for(int PositionCount=0; PositionCount<13; PositionCount++)

{

lcd.scrollDisplayRight(); //builtin command to scroll right the text

delay(150);//delay of 150 msec

}

for(int PositionCount=0; PositionCount<13; PositionCount++)//loop for scrolling the text

{

lcd.scrollDisplayLeft();//builtin command to scroll the text left again

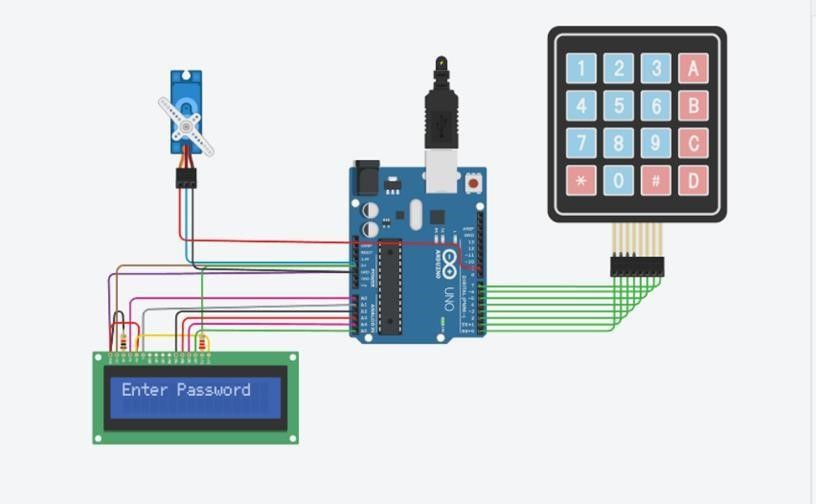
delay(150);//delay of 150 msec

}

}

**Mini project on thinker cad using Arduino board:**

**digital door lock opener using Arduino:**



## 

A digital door lock opener using Arduino combines various components to create a functional system. At the core of this system is the Arduino microcontroller, which acts as the central processing unit. It receives inputs from a keypad or an RFID (Radio Frequency Identification) reader, allowing users to enter a password or use an RFID card for access.

The Arduino code implements a validation mechanism to compare the user's input with authorized codes or credentials. If the input matches an authorized code, the Arduino sends signals to control the lock mechanism, which can be a servo motor or a solenoid. The lock mechanism physically controls the door, unlocking it to grant access or locking it to deny access.

Optionally, an LCD display can be integrated into the system to provide feedback and display relevant information to the user. This can include prompts for input, error messages, or status updates on the locking/unlocking process.

**Here are the main theoretical concepts involved:**

1. **Arduino Microcontroller:** The Arduino serves as the brain of the system. It is responsible for processing inputs, making decisions, and controlling the lock mechanism based on predefined conditions.
2. **Keypad or RFID Reader:** The input device allows users to enter a password or use an RFID card to gain access. The keypad can be a matrix of buttons, while the RFID reader communicates with RFID tags or cards.
3. **Password/RFID Validation:** The Arduino code implements a validation mechanism to compare the user's input (password or RFID tag) against authorized codes or credentials. If the input matches an authorized code, access is granted; otherwise, access is denied.
4. **Servo Motor or Solenoid Lock:** The lock mechanism physically controls the door, either by rotating a servo motor or activating a solenoid. The Arduino sends signals to the lock mechanism to unlock or lock the door based on the validation result.

**The working of a locker using Arduino involves the following steps:**

1. User provides authentication credentials through an input interface such as a keypad, RFID reader, or biometric scanner.
2. Arduino validates the credentials against authorized data stored in its memory.
3. If the credentials are valid, Arduino sends signals to the lock control mechanism to unlock the locker.
4. Optionally, feedback on the locker's status can be provided through an LCD display or LEDs.
5. Security measures should be implemented to protect the system and data, such as encryption and secure communication protocols.

**3.9 Module Outcome:**

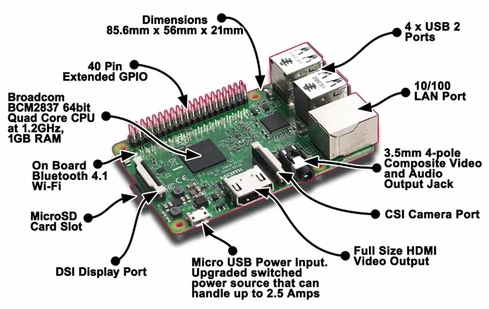
From this module The Students will be able To Design and Program the Arduino

Circuits & The Monitor.

**CHAPTER 4:**

Introduction to IOT using Raspberry pi

## 4.1: Introduction to Raspberry pi:



Raspberry Pi is a credit card-sized single-board computer that offers a wide range of possibilities for electronic projects and applications. It is known for its versatility, affordability, and accessibility. Raspberry Pi boards come with various models, each offering different specifications and capabilities.

Key Features:

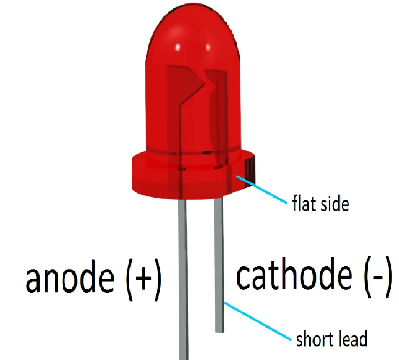
* Hardware: Raspberry Pi boards include a processor, memory, storage, GPIO pins, USB ports, HDMI output, and other essential components.
* Operating System: Raspberry Pi supports various operating systems, such as Linux-based distributions like Raspbian, Ubuntu, and others.
* Connectivity: Raspberry Pi boards offer connectivity options like Ethernet, Wi-Fi, Bluetooth, and USB, allowing for interaction with other devices and the internet.
* GPIO Pins: Raspberry Pi includes GPIO pins that allow for easy interfacing with external devices, enabling control and sensing capabilities.
* Programming: Raspberry Pi supports multiple programming languages, including Python, C/C++, and Java, making it accessible to a wide range of developers.

**Applications of Raspberry Pi:**

* Home Automation
* Internet of Things (IoT) Projects
* Media Centers
* Education
* Robotics

**4.2: IOT Components:**

**4.2.1: LED:**

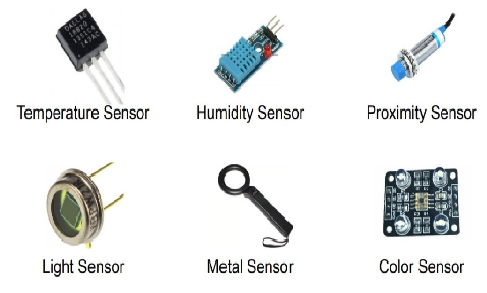
An LED, or Light-Emitting Diode, is a semiconductor device that emits light when an electric current passes through it. It is a solid-state technology that has revolutionized lighting due to its numerous advantages. LEDs are highly energy-efficient, converting a higher percentage of electrical energy into light compared to traditional incandescent bulbs.

**4.2.2: Resistors:**

A resistor is a passive electronic component that opposes or restricts the flow of electric current in a circuit. It is designed to introduce a specific amount of resistance to the flow of electrons, which in turn affects the voltage and current levels within the circuit.

**4.2.3: Jumper Wires:**

The primary purpose of jumper wires is to create temporary or permanent connections between various components, such as microcontrollers, sensors, actuators, and other electronic devices. They eliminate the need for soldering and provide flexibility for quick modifications or experimentation in circuits.

**4.2.4: Sensors:**

Sensors are devices that detect physical changes and convert them into electrical signals. They are used to measure parameters like temperature, pressure, light, and motion. Sensors play a crucial role in electronics, automation, and various applications by providing information for decision-making and control.

**4.2.5 Gas Sensors (MQ – 2):**

Gas sensors, specifically the MQ-2 gas sensor, are electronic devices used to detect and measure the presence of various gases in the surrounding environment. The MQ-2 sensor is commonly used for detecting gases such as methane, propane, butane, alcohol, smoke, and other flammable gases.

The MQ-2 gas sensor operates on the principle of metal oxide semiconductors. It consists of a sensing element that contains a metal oxide material. When the target gas comes into contact with the sensing element, it undergoes a chemical reaction that causes a change in the resistance of the sensor. This change in resistance is then measured and converted into an electrical signal, indicating the presence and concentration of the gas.

### **4.2.6 Button Switches:**

Button switches, also known as tactile switches or push buttons, are ubiquitous electronic components that serve as user input devices. They consist of a housing, contacts, and a button or actuator. When the button is pressed, it activates the internal mechanism, causing the contacts to momentarily connect and complete an electrical circuit. When the button is released, the contacts separate, breaking the circuit.

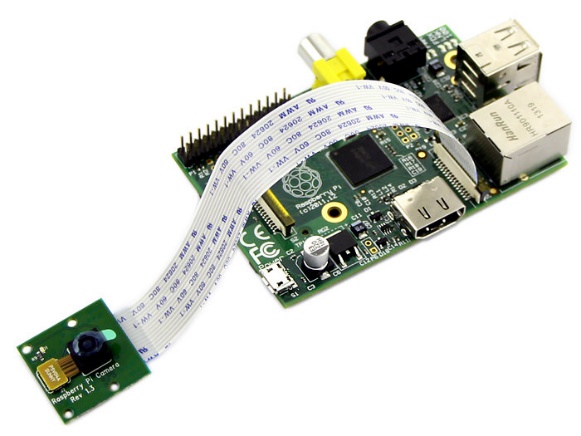
Button switches are designed to provide a tactile feedback or click sensation when pressed, giving users a physical confirmation of the switch activation. This feedback is often accompanied by an audible click sound. The tactile response makes button switches useful for applications that require precise user input or control, such as keyboards, gaming controllers, industrial control panels, and various electronic devices.

**4.2.7 Buzzers:**

Buzzers are electronic devices that produce audible sound or tones when an electric current is applied to them. They are commonly used for audio alerts, notifications, alarms, and signalling purposes.

Buzzers consist of a piezoelectric element or an electromagnetic coil and a diaphragm. When an electric current is passed through the buzzer, the piezoelectric element vibrates or the coil generates a magnetic field, causing the diaphragm to move back and forth rapidly. This movement creates sound waves, producing an audible tone or buzz.

### **4.2.8 Pi Camera Module Interface with Raspberry Pi using Python 68:**



The Pi Camera Module is an accessory specifically designed for the Raspberry Pi single-board computer. It is a small, high-quality camera that can be easily connected to the Raspberry Pi, enabling users to capture photos and videos directly from their projects.

The Pi Camera Module is compact and lightweight, making it ideal for embedding in various applications, such as robotics, security systems, home automation, and more. It connects to the Raspberry Pi through a ribbon cable, providing a convenient and straightforward interface.

The camera module offers high-resolution imaging capabilities, with different versions available that support varying resolutions and features. For instance, the original Pi Camera Module (v1) had a fixed focus lens and offered a resolution of 5 megapixels, while the later Pi Camera Module v2 provided an 8-megapixel resolution and added features like adjustable focus.

**4.3: Introduction to Python:**



Python is a popular and versatile programming language known for its simplicity and readability. It is widely used for web development, data analysis, artificial intelligence, scripting, and more. Python has a large and active community that contributes to its extensive library of packages and resources. It is cross-platform compatible and offers seamless integration with other languages. Python's ease of learning and powerful capabilities makes it a top choice for both beginners and experienced developers. Its broad range of applications and thriving ecosystem make it a valuable tool in the world of programming.

### **4.4 Program to Flash an LED at a given on time and off time cycle, Where the two times are taken from a file.**

#define redLed 5

#define bluLed 6

long previousMillis[2]; //[x] = number of leds

void setup() {

pinMode(redLed, OUTPUT);

pinMode(bluLed, OUTPUT);

}

void loop()

{

//OnOffBlink(which led?, tOn, tOff, One of the previousMillis);

OnOffBlink(redLed, 300, 500, 0); //array must be different

OnOffBlink(bluLed, 300, 300, 1); //for each led

}

void OnOffBlink(int led, int tOn, int tOff, int array)

{

static int timer=tOn;

if ((millis() - previousMillis[array]) >= timer)

{

if (digitalRead(led) == HIGH) {

timer = tOff;

}

else

{

timer = tOn;

}

digitalWrite(led, !digitalRead(led));

previousMillis[array] = millis();

}

}

**4.5 Program to Get Input from Two Switches and Switch on:**

**Corresponding LED**

**s**import RPi.GPIO as GPIO #Import GPIO library

import time #Import time library

GPIO.setmode(GPIO.BOARD) #Set GPIO pin numbering

GPIO.setup(12, GPIO.IN, pull\_up\_down=GPIO.PUD\_UP) #Enable input and pull up resistors while True:

input\_state = GPIO.input(12) #Read and store value of input to a variable

if input\_state == False: #Check whether pin is grounded print('Button

Pressed') #Print 'Button Pressed' time.sleep(0.3) #Delay of 0.3

**4.6 Program to Access an Image through a Pi Web Cam:**

from picamera import PiCamera import time camera = PiCamera() time.sleep(2) camera.resolution = (1280, 720) camera.vflip = True camera.contrast = 10 file\_name = "/home/pi/Pictures/video\_" + str(time.time()) + ".h264" print("Startrecording...")

camera.start\_recording(file\_name)

camera.wait\_recording(5) camera.stop\_recording()

print("Done.")

**4.7 Program to Get an Alarm from a Remote Area (Through LAN)**

**If Smoke is detected**

import socket

import Adafruit\_MCP3008 import Adafruit\_GPIO.SPI as SPI import time

HOST = ‘192.168.234.121’

PORT = 3000

SPI\_DEVICE = 0 SPI\_PORT = 0 mcp = Adafruit\_MCP3008.MCP3008(spi=SPI.SpiDev(SPI\_PORT,SPI\_DEVICE)) try: with socket.socket(socket.AF\_INET,socket.SOCK\_STREAM) as s: s.bind((HOST,PORT)) s.listen()

conn,addr = s.accept() with conn:

print(‘connected by’,addr) while True: value = mcp.read\_adc(0)

print(“Gas value”,value, “units”) if (value>80): data = “Alert”.encode( ‘utf-8’ ) conn.sendall(data) Client:

import socket import RPi.GPIO as GPIO import time Buzzer = 35

HOST = ‘192.168.234.121’ PORT = 3000

GPIO.setmode(GPIO.BOARD)

GPIO.setup(36,GPIO.OUT) GPIO.setup(False) try: with socket.socket (socket.AF\_INET,socket.SOCK\_STREAM) as s: s.connect((HOST,PORT)) while True:

data = s.recv(1024).decode(‘utf-8’) if (str(data) == ‘Alert’) : print(“Alert!gas leakage detected”) GPIO.output(35,True) time.sleep(3) GPIO.output(36,False) time.sleep(3) except KeyboardInterrupt: s. close() GPIO.cleanup()

**4.8**: **Module Outcome:**

* Raspberry Pi proves to be a smart, economic and efficient platform for implementing The home automation.
* The system is Flexible and programmable and has wide range applications and Supports wide Varity of peripherals and accessories
* The system can be accessed from any internet based device including handheld

Devices such as mobile phones.

**CHAPTER 5:**

3-D PRINTING

3D printing, also known as additive manufacturing, is a revolutionary technology that enables the creation of three-dimensional objects by building them layer by layer. Instead of traditional subtractive manufacturing processes that involve cutting, drilling, or molding, 3D printing adds material to create objects.

The process starts with a digital model of the object, which is sliced into thin layers. These layers serve as instructions for the 3D printer, which deposits or solidifies material, typically plastic or resin, layer by layer until the complete object is formed.

3D printing offers several advantages. It allows for the production of complex and intricate designs that would be challenging or impossible with traditional methods. It enables rapid prototyping, reducing time and costs in the product development cycle. Customization and personalization are key benefits, as objects can be tailored to individual needs or specific requirements.

**5.1: TECHNOLOGIES:**

Each technology has its strengths, limitations, and suitable applications, and the choice of technology depends on factors such as desired print quality, material compatibility, cost, and intended use of the printed objects. Here are some of the key technologies in 3D printing:

* **Selective Laser Sintering (SLS):** SLS employs a high-power laser to selectively fuse powdered materials, such as nylon or metal, layer by layer. The laser heats the powdered material, causing it to fuse and solidify. Unfused powder supports the structure during printing, eliminating the need for additional support structures.
* **Digital Light Processing (DLP):** DLP is similar to SLA, but instead of using a laser, it uses a digital light projector to cure a liquid resin. The entire layer is exposed to light simultaneously, allowing for faster print times compared to SLA.
* **Binder Jetting:** Binder Jetting involves the deposition of a liquid binder onto thin layers of powdered material, such as sand or metal. The binder binds the particles together, layer by layer, forming the object. After printing, additional post-processing, such as sintering, may be required for improved strength and durability.
* **Material Jetting:** Material Jetting works by jetting or depositing droplets of liquid photopolymer or wax materials onto the build platform. These droplets solidify through UV light exposure or cooling, forming the object layer by layer. Material Jetting can produce multi-material and multi-color prints simultaneously.
* **Direct Metal Laser Sintering (DMLS)/Selective Laser Melting (SLM):** DMLS/SLM uses a high-power laser to selectively melt and fuse metal powders together, creating fully dense metal parts. The laser melts the metal powder particles, which solidify upon cooling. DMLS/SLM is primarily used for producing metal components with complex geometries.

**5.2: Applications:**

3D printing has a wide range of applications across various industries. Here are some notable applications of 3D printing:

* **Prototyping and Product Development:** 3D printing allows rapid prototyping, enabling designers and engineers to quickly iterate and test their designs before mass production. It speeds up the product development cycle and reduces costs.
* **Healthcare and Medical Applications:** 3D printing has revolutionized healthcare by enabling the production of patient-specific medical devices, prosthetics, and implants. It allows for personalized and tailored solutions, improving patient outcomes. It is also used for creating anatomical models for surgical planning and education.
* **Aerospace and Automotive:** 3D printing is used in aerospace and automotive industries for rapid prototyping, manufacturing lightweight components, and producing complex geometries. It helps reduce weight and improve fuel efficiency.
* **Architecture and Construction:** 3D printing is utilized in architecture and construction for creating intricate models, prototypes, and even full-scale structures. It enables faster and more cost-effective construction methods.
* **Education and Research:** 3D printing has educational applications by providing hands-on learning experiences. It allows students to bring their designs to life and enhances understanding of concepts in various fields. It is also used in scientific research for creating custom lab equipment, models, and prototypes.

These are just a few examples of the wide-ranging applications of 3D printing.

**5.3: Modeling Software:**

### **SOLID EDGE**

Solid Edge is a professional 3D computer-aided design (CAD) software developed by Siemens Digital Industries Software. It offers a comprehensive suite of tools for designing, simulating, and manufacturing mechanical and electrical systems. With its user-friendly interface and robust functionality, Solid Edge is widely used across industries such as mechanical engineering, automotive, aerospace, and consumer products.

Key features of Solid Edge include powerful 3D modeling capabilities, allowing users to create complex parts and assemblies using parametric, direct, and synchronous modeling techniques. It offers specialized tools for sheet metal design, simulation and analysis for structural and thermal performance, drafting and documentation for detailed drawings, and electrical design capabilities for creating electrical schematics and harness designs. Here are some features on solid edge:

* Advanced 3D modeling techniques for precise part and assembly creation.
* Specialized tools for sheet metal design and automatic unfolding.
* Assembly design and management capabilities, including motion simulation and interference detection.
* Simulation and analysis tools for structural, thermal, and fluid flow performance evaluation.
* Comprehensive drafting tools for detailed 2D drawings with annotations and dimensions.
* Electrical design capabilities for creating schematics, wiring diagrams, and harness designs.
* Support for additive manufacturing, including design optimization and support structure generation.
* Built-in data management for collaboration, version control, and integration with Siemens' Teamcenter.
* Seamless integration with CAM software for CNC machining and manufacturing processes.

**5.4 Module Outcome:**

3D printing can be a great teaching tool to enhance students' understanding of complex concepts and processes, which should presumably improve learning performance, problem solving and critical thinking skills, communication and collaboration skills, and attitudes, engagement, and motivation.

## CHAPTER 6:

## MICROSOFT OFFICE

Microsoft Office is a suite of productivity applications developed by Microsoft. It includes widely used programs such as Word for word processing, Excel for spreadsheets, PowerPoint for presentations, Outlook for email and personal information management, and OneNote for note-taking. Additional applications like Access for database management, Publisher for desktop publishing, and Teams for communication and collaboration are also part of the suite.

Microsoft Office is designed to enhance productivity and streamline tasks in various professional and personal settings. It provides tools for creating, editing, and formatting documents, managing data and calculations, designing presentations and publications, organizing emails and schedules, and facilitating collaboration among team members.

The suite is known for its user-friendly interface, extensive functionality, and compatibility across different platforms and devices. Microsoft Office has become a standard in many industries and educational institutions, empowering users to create professional-quality content, analyse data, communicate effectively, and streamline workflows.

**6.1: Microsoft Word:**

Microsoft Word is a word processing program that is part of the Microsoft Office suite. It is one of the most widely used applications for creating, editing, and formatting documents.

Microsoft Word provides a user-friendly and feature-rich environment for creating and editing text-based documents. It offers a wide range of tools and functions to help users format text, add images and tables, and customize the layout of their documents. Word supports various file formats, allowing seamless compatibility with other word processing software.

With Microsoft Word, users can create professional-looking documents for various purposes, such as reports, letters, resumes, and manuscripts. It offers features like spell checking, grammar correction, and thesaurus to enhance the quality and accuracy of written content. Users can also collaborate with others in real-time, making it easy to review and edit documents as a team

**6.2: Microsoft Excel:**

Microsoft Excel is a powerful spreadsheet program that allows users to organize, analyse, and manipulate data effectively. It offers a range of features and functions that make it a versatile tool for various tasks.

It enables users to input data, organize it in rows and columns, and perform calculations using built-in mathematical formulas. Excel supports a vast number of functions, ranging from basic arithmetic operations to advanced statistical and financial calculations.

One of Excel's key features is its ability to handle large amounts of data. Users can easily sort, filter, and format data to create meaningful insights. Excel also supports the creation of charts and graphs, making it easier to visualize data trends and patterns.

Excel's functionality extends beyond basic calculations. It offers tools for data analysis, including pivot tables, which allow users to summarize and analyse large datasets quickly. Users can also create custom macros and automate repetitive tasks to improve efficiency.

**6.3: Microsoft Power point:**

Microsoft PowerPoint is a presentation software developed by Microsoft as part of the Microsoft Office suite. It is widely used to create visually appealing and engaging presentations for various purposes. It allows users to combine text, images, videos, and other multimedia elements to convey information effectively. With PowerPoint, users can craft visually stunning slideshows that capture the attention of their audience.

PowerPoint provides a variety of slide templates and themes to choose from, making it easy to create consistent and visually appealing presentations. Users can add text, format it with different fonts, colours, and styles, and arrange it on slides to create a logical flow of information.

The software offers tools for inserting and editing images, charts, tables, and SmartArt graphics to enhance the visual impact of presentations. Users can also incorporate audio and video files, transitions, animations, and slide timings to create engaging and interactive presentations. PowerPoint enables users to deliver presentations in various formats, including on-screen slideshows, printed handouts, or even as self-running presentations. It provides features for rehearsing timings, adding speaker notes, and controlling the presentation flow during live delivery.

Collaboration is made easy with PowerPoint's cloud-based features. Multiple users can work on the same presentation simultaneously, and changes are synced in real-time. PowerPoint also integrates with other Office applications, allowing seamless insertion of content from Word or Excel, and easy sharing and collaboration with colleagues.

Microsoft PowerPoint is widely used in business, education, and other professional settings for presentations, training sessions, sales pitches, and more. Its intuitive interface, extensive customization options, and multimedia capabilities make it a go-to tool for creating impactful and visually appealing slideshows.

**6.4 Module Outcome:**

Microsoft Office offers enhanced productivity, streamlined communication and collaboration, effective data management and analysis, professional document creation, efficient presentations, and access to a comprehensive suite of applications that support various tasks and requirements.

## CHAPTER 7:

## CONCLUSION AND FUTURE SCOPE

**7.1 Conclusion:**

It provided valuable real-world work experience, practical skills development, industry exposure, networking opportunities, and a pathway to future career success. They offer a chance to apply knowledge, gain insights into specific industries, and make informed career decisions. The given modules having a very good scope in industry, will be helpful to get to know about the technical and practical knowledge about technology.

**7.2: Future Scope:**

**Arduino:** Arduino has a bright future with growing interest in electronics and IoT. It offers endless possibilities for innovation and automation in various industries, making it a valuable skillset for aspiring engineers and inventors.

**Raspberry Pi:** Raspberry Pi's versatility and affordability make it well-positioned for future applications in home automation, robotics, IoT, and educational projects. Its continuous development and expanding community ensure an exciting future for this powerful single-board computer.

**3D Printing:** The future of 3D printing looks promising as it revolutionizes manufacturing and customization. Advancements in materials, speed, and accessibility will lead to wider adoption in industries like healthcare, aerospace, and consumer products, shaping the future of production.

**Microsoft Office:** Microsoft Office remains an essential suite for productivity and communication. Its continuous updates and integration with cloud services ensure it will adapt to evolving work environments, enabling efficient document creation, data analysis, collaboration, and professional communication.