Q1. What is Wokwi?

Ans. Wokwi is an online platform for simulating electronic projects, particularly those involving microcontrollers like Arduino, ESP32, and Raspberry Pi Pico. It allows users to design, test, and debug circuits in a virtual environment, providing a web-based interface for writing and running code, connecting virtual components, and visualizing the behavior of the circuit without needing physical hardware.

2. What is Tinkercad?

Ans. Tinkercad is a free, web-based application for 3D design, electronics, and coding. It is particularly popular for educational purposes and among beginners. Users can create 3D models, design electronic circuits with components like Arduino, and simulate their operation. Tinkercad's intuitive interface makes it accessible for users of all ages and skill levels.

3. Difference between Wokwi and Tinkercad.

Ans.

Feature	Wokwi	Tinkercad
① Primary Focus	Electronics simulation, particularly microcontrollers (Arduino, ESP32, Raspberry Pi Pico)	3D design, electronics simulation, and coding
② Interface	Web-based, code-based	Web-based, visual and code-focused
③ Supported Microcontrollers	Arduino, ESP32, Raspberry Pi Pico	Arduino

(4) 3D Design	No	Yes
(5) Component Library	Extensive, focused on electronic components	Extensive, includes electronic and mechanical parts
(6) Simulation	Detailed, code-centric	Visual and interactive
(7) Education Tools	Limited	Extensive, with tutorials and lesson plans
(8) Community	Yes, with public projects sharing and examples	Yes, with public projects and gallery
(9) Cost	Free	Free

4. Which board can be used/simulated on Wokwi?

Ans. On Wokwi, we can simulate various microcontroller boards and modules. Some of the popular ones are:

(1) Arduino Uno
(2) Arduino Mega 2560
(3) ESP8266 NodeMCU
(4) ESP32 Dev Module
(5) Raspberry Pi Pico
(6) STM32 Blue Pill (STM32F103C8T6)
(7) Atmega328p (Generic 328p Board)

5. Which board can be simulated on Tinkercad?

Ans. On Tinkercad, we can simulate various microcontrollers boards which are as follows:

(1) Arduino Uno
(2) Arduino Mega
(3) Arduino Nano
(4) Arduino Leonardo
(5) Arduino Due

6. List sensors and actuators available in Wokwi.

Ans. Sensors:

1. HC-SR04 Ultrasonic Distance Sensor
2. DHT22
3. DS18B20 Temperature Sensor
4. PIR Motion Sensor
5. Analog Temperature Sensor (NTC)
6. Photoresistor (LDR) Sensor
7. MPU6050 Accelerometer + Gyroscope
8. DS1307 RTC

Actuators:

1. LEDs
2. Buzzer
3. Servo Motor
4. Bipolar Stepper Motor
5. A4988 Stepper Motor Driver
6. Biaxial Stepper Motor
7. WS2812 SPI Tri-Color LED Driver
8. Relay Module
9. DPDT Relay
10. LCD (16x2 and 20x4)
11. 7-Segment Display
12. Neopixel RGB LED.

7. List sensors and actuators available in Tinkercad.

Ans. Sensors:

① Potentiometer, ② Photoresistor, ③ Photodiode, ④ Ambient Light Sensor [Phototransistor], ⑤ Flex Sensor ⑥ Force Sensor ⑦ IR Sensor ⑧ Ultrasonic Distance Sensor, ⑨ PIR Sensor, ⑩ Soil Moisture Sensor, ⑪ Tilt Sensor, ⑫ Temperature Sensor [TMP36], ⑬ Gas Sensor.

Actuators:

① LEDs (Various colors), ② LED RGB, ③ Light Bulb ④ NeoPixel RGB LED, ⑤ Vibration Motor, ⑥ DC Motor ⑦ DC Motor with encoder, ⑧ Micro Servo, ⑨ Hobby Gearmotor, ⑩ Piezo buzzer, ⑪ 7 Segment Display, ⑫ LCD Display (16x2).

8. State limitations of Wokwi.

Ans. The limitations of Wokwi are as follows:

① Limited Component Library: Not all components and sensors are available.

② Real-World Accuracy: Simulations may not perfectly match real-world conditions.

③ Project Complexity: Handling very complex projects can be challenging.

④ Internet Dependency: Requires an internet connection to use.

⑤ Educational Focus: More suited for learning than for production-level testing.

⑥ Smaller Community: Fewer resources and support compared to more established tools.

9. State limitations of Tinkercad.

Ans. The limitations of Tinkercad are as follows:

① Component library: Limited selection of components compared to other more advanced simulation tools.

② Complex Projects: Handling and simulating very complex circuits can be difficult.

③ Internet Requirement: Must be online to use, which can be restrictive.

④ Feature set: Limited advanced features for professional or industrial use.

10. What is Raspberry Pi and its different types.

Ans. The Raspberry Pi is a series of small, affordable single-board computers developed by the Raspberry Pi Foundation. Raspberry Pi boards are widely used in both educational and hobbyist projects, as well as for practical applications such as home automation and IoT devices.

Types:

① Raspberry Pi 1 (Model A and B).
② Raspberry Pi 2 (Model B)
③ Raspberry Pi 3 (Model B and B+)
④ Raspberry Pi 4 (Model B)
⑤ Raspberry Pi 5
⑥ Raspberry Pi Pico

Q11. State advantages and disadvantages of Raspberry Pi.

Ans. Advantages:

(1) Affordable: Low-cost, making it accessible for a wide range of users.

(2) Compact: Small form factor allows for easy integration into various projects.

(3) Versatile: Suitable for a wide range of applications, from education to IoT and home automation.

(4) Community Support: Large and active community with abundant resources, tutorials, and forums.

Disadvantages:

(1) Performance Limitations: Not as powerful as traditional PCs, limiting its use for high-performance applications.

(2) Storage: Use microSD cards for storage, which can be slower and less reliable than SSDs.

(3) No Built-in Storage: Lacks built-in storage, requiring external solutions for larger data needs.

(4) Heat Management: Can overheat under heavy load without proper cooling solutions.

12. Compare RPi, Arduino, NodeMCU and ESP32.

Ans. Raspberry Pi:

Type: Single-board computer

CPU: ARM Cortex (varies by model)

OS: Linux-based (e.g., Raspbian)

Storage: microSD Card

Memory: 512 MB to 8GB RAM

I/O Pins: 40 GPIO pins

Connectivity: Ethernet, WiFi, Bluetooth

② Arduino:
Type : Microcontroller board
CPU : ATmega (varies by model)
OS : None (bare-metal programming)
Memory : 2KB to 8KB SRAM.
Storage : Flash memory (typically 32KB)
Connectivity : limited (shields available)
I/O Pins : Varies by model (around 20)

③ Node MCU :
Type : Microcontroller board
CPU : ESP8266
OS : None (Lua, Arduino IDE)
Memory : 80KB to 160KB SRAM
Storage : Flash memory (typically 4MB)
Connectivity : WiFi
I/O Pins : Around 11 GPIO pins

④ ESP32
Type : Microcontroller board
CPU : Dual-core Tensilica Xtensa LX6
OS : None (Arduino IDE, MicroPython, ESP-IDF)
Memory : 520KB SRAM
Storage : Flash memory (varies, typically 4MB)
Connectivity : WiFi, Bluetooth
I/O Pins : Around 34 GPIO pins

13. What is sensor? Give 5 examples.

Ans. A sensor is a device that detects and responds to some type of input from the physical environment. The input could be light, heat, motion, moisture, pressure, or any other environmental problem phenomena. The output is generally a signal that can be read by humans or by another system.

Examples:

1. Temperature Sensor (DS18B20)
2. LDR (Light Dependent Resistor)
3. PIR (Passive Infrared) Sensor
4. Humidity Sensor (DHT11)
5. Pressure Sensor (BMP180)

14. What is actuator? Give 5 examples.

Ans. An actuator is a device that converts energy (often electrical) into physical motion or action. It is typically used to move or control a mechanism or system. Actuators are integral to many systems, enabling automated and controlled physical responses to inputs.

Examples:

1. DC Motor
2. Buzzer
3. Servo Motor
4. Hydraulic Cylinder
5. Solenoid

15. What is digital sensor? Explain.

Ans. A digital sensor is a type of sensor that converts the physical parameter it measures into a digital signal. Unlike analog sensors, which output a continuous signal that represents the measured quantity, digital sensors output discrete digital signals, often in the form of binary data (0s and 1s).

16. What is analog sensor? Explain.

Ans. An analog sensor is a type of sensor that produces a continuous output signal (analog signal) that is proportional to the measured physical quantity. This output can vary over a range, rather than being limited to discrete levels, which is typical of digital sensors.

17. State full form of HDMI, USB, SD, DP, BCM, GPIO, ARM, RISC, CI

Ans. HDMI - High-Definition Multimedia Interface
USB - Universal Serial Bus
SD - Secure Digital
DP - Display Port
BCM - Broadcom
GPIO - General-Purpose Input/Output
ARM - Advanced RISC Machines
RISC - Reduced Instruction Set Computer
CISC - Complex Instruction Set Computer

18. State purpose of softwares:

Ans. ① Advanced IP Scanner: Scans and identifies devices on a local network.

② VNC Viewer: Allows remote control of another computer.

③ PuTTY: SSH client for remote terminal access to Unix/Linux systems.

④ FileZilla: Transfers files between local and remote servers via FTP/SFTP.

⑤ SD CARD Formatter: Formats SD cards for optimal performance and compatibility.

⑥ RPi Imager: Writes OS images to SD cards for Raspberry Pi setup.

19. Which Raspberry Pi was used in the Lab? State its configuration.

Ans. Raspberry Pi Model B was used in the Lab.

Configuration:

① CPU: Quad-core ARM Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz

② Memory: Options for 2GB, 4GB, or 8GB LPDDR4-3200 SDRAM

③ Storage: microSD card slot for loading operating system and data storage.

④ Connectivity: WiFi, Bluetooth, Gigabit Ethernet

⑤ USB Ports: 2 x USB 3.0 ports, 2x USB 2.0 ports

⑥ Video Output: 2x micro-HDMI ports (up to 4Kp60 supported)

⑦ GPIO: 40-pin GPIO header

⑧ Power: 5V DC via USB-C connector (minimum 3A).

20. Which OS is used in Raspberry Pi? State its configuration.

Ans. The most commonly used operating system for Raspberry Pi is Raspberry Pi OS (formerly known as Raspbian.

Configuration:

(1) Base: Debian Linux
(2) Desktop Environment: LXDE (Lightweight X11 Desktop
(3) Kernel: Linux Kernel
(4) Package Manager: APT (Advanced Package Tool)
(5) User Interface: GUI and CLI.
(6) Updates: Regular updates and security patches.

21. Which programming language is used to program hardware in Raspberry Pi, ESP32 and NodeMCU?

Ans. Python is the programming language used for programming Hardware in Raspberry Pi, ESP32 and NodeMCU.

21. Which programming language is used to program hardware in Raspberry Pi, ESP32 and NodeMCU?

Ans. Raspberry Pi - Python
ESP32 - C/C++
NodeMCU - C/C++

22. What is DHT sensor used for? State different sensors available on DHT? State types of DHT sensors with specifications.

Ans. DHT Sensor Usage:

DHT sensor is used to measure temperature and humidity.

Different Sensors Available on DHT:

- DHT11
- DHT22 (AM2302)

Types of DHT Sensors and Specifications:

1. DHT11:

- Temperature Range: 0 to 50°C
- Temperature Accuracy: ±2°C
- Humidity Range: 20% to 90% RH
- Humidy Accuracy: ±5% RH
- Sampling Rate: 1Hz (once per second)

2. DHT22 (AM2302):

- Temperature Range: -40 to 80°C
- Temperature Accuracy: ±0.5°C
- Humidity Range: 0% to 100% RH
- Humidity Accuracy: ±2-5% RH
- Sampling Rate: 0.5Hz (once every 2 seconds)

23. What is LDR? State purpose of LDR sensor.

Ans. An LDR, or Light Dependent Resistor, is a type of resistor whose resistance varies significantly with the amount of light falling on it. It is made of a high-resistance semiconductor.

Purpose:

① Photographic Exposure Meters: Measure the light intensity to determine the correct exposure settings for cameras.

② Consumer Electronics: Automate screen brightness in devices like smartphones, tablets, and laptops based on surrounding light conditions.

③ Industrial Automation: Detect the presence of objects and changes in light levels to control machinery and production processes.

24. Which sensors are used for sensing LPG Gas, CNG Gas, Air Quality and alcohol? State working principles of these sensors.

Ans. ① LPG Gas Sensor:

Sensor: MQ-2

Working Principle: Tin dioxide (SnO_2) layer changes resistance with LPG presence, producing an analog signal.

② CNG Gas Sensor:

Sensor: MQ-4

Working Principle: SnO_2 layer changes resistance with methane (CNG) giving an analog output.

③ Air Quality Sensor:
Sensor: MQ-135
Working Principle: SnO_2 layer changes resistance with various gases (NH_3, NO_x, alcohol, benzene, smoke, (O_2), producing an analog signal.

④ Alcohol Sensor:
Sensor: MQ-3
Working Principle: SnO_2 layer decreases resistance with alcohol vapors, resulting in an analog output.

25. State working of obstacle sensor. State its application in IoT. Which LED and diode are used in obstacle sensor.

Ans. Working of obstack sensor:
Obstacle sensors typically use infrared (IR) technology. They emit infrared light from an IR LED and then detect the reflected light using a photodiode or phototransistor. When an object is present, the reflected light intensity changes, causing a corresponding change in the output signal.

Applications:
Smart Security Systems: Detecting intruders or unauthorized entry.
Smart Home Automation: Triggering lights or appliances based on movement.

Components in Obstacle Sensor:
LED: Infrared (IR) LED emits infrared light.

- Photodiode or Phototransistor: Receives and detects reflected infrared light, producing a corresponding electrical signal.

26. State working of fire sensor. State its application in IoT. Which diode is used in fire sensor.

Ans. Working of Fire Sensor:

Working Principle: Fire sensors typically utilize either smoke detection or heat detection methods.

- Smoke Detection: Uses a smoke chamber where particles from smoke disrupt a light beam, triggering an alarm.
- Heat Detection: Measures temperature changes indicative of a fire.

Applications:

- Early Fire Detection: Sending alert alerts to mobile devices or IoT platforms for timely response.
- Industry Safety: Monitoring fire hazards in factories and warehouses.

Components in Fire Sensor:

- Photodiode: Used in smoke detection types to detect changes in light intensity due to smoke particles disrupting a light beam.

27. List functions with examples used to create project using RPi.

Ans. RPi.GPIO Basics:

-> Importing the library and setting up the GPIO mode:

```
import RPi.GPIO as GPIO
# Set the GPIO mode (BCM or BOARD)
GPIO.setmode(GPIO.BCM)
```

- Example: Setting up GPIO pin 17 as an output:
  GPIO.setup (17, GPIO.OUT)
- Example: Setting up GPIO pin 18 as an input:
  GPIO.setup (18, GPIO.IN)

② GPIO output:
→ Function: Setting a GPIO pin to High or Low:
GPIO.output (pin_number, GPIO.HIGH) # (3.3V)
GPIO.output (pin_number, GPIO.LOW) # (0V)
- Example: Turning an LED connected to GPIO pin 17 ON:
GPIO.output (17, GPIO.HIGH)
- Example: Turning the same LED off.
  GPIO.output (17, GPIO.LOW)

③ GPIO Input:
→ Function: Reading the state of an input GPIO pin:
input_state = GPIO.input (pin_number)
- Example: Checking if a button connected to GPIO pin 18 is pressed:

```
if GPIO.input(18) == GPIO.HIGH:
        print("Button pressed")
else:
        print("Button not pressed")
```

Q 28 List functions with examples used to create project using ESP32/ NodeMCU.

Ans. ① Setup function: Initialization of GPIO pins, serial communication, and other hardware settings.

.Example:

```
void setup() {
    pinMode(D5, OUTPUT); // Set GPIO D5 as output
    Serial.begin(9600); // Start serial communication
}
```

② Loop Function: Continuous execution of the main program logic.

.Example:

```
void loop() {
    digitalWrite(D5, HIGH); //Set GPIO D5 high (3.3V)
    delay(1000);            // Delay for 1 second
    digitalWrite(D5, LOW); // Set GPIO D5 low (0V)
    delay(1000);            // Delay for 1 second
}
```

③ Digital Input/Output: Control and read digital signals on GPIO pins.

.Example:

```
digitalWrite(pin_number, HIGH);
digitalWrite(pin_number, LOW);
digitalRead(pin_number);
```

(4) Analog Input/Output: Read analog sensor values or output analog signals (PWM).

- Example

```
analogRead(A0);
analogWrite(D6, 128); // Output PWM signal on GPIO D6 (50% duty cycle)
```

29. Which IDE is used to write program on RPi, ESP32, NodeMCU, Arduino?

Ans. Raspberry Pi - IDLE (Python)
ESP32 - Arduino IDE
NodeMCU - Arduino IDE
Arduino - Arduino IDE

30. What are settings needed for ESP32/NodeMCU to connect to a PC?

Ans. To connect an ESP32 or NodeMCU to a PC:
(1) USB Drivers: Install necessary USB drivers for your ESP32 or NodeMCU board on your PC.
(2) USB Cable: Use a micro-USB cable to connect the board to your PC.
(3) Development Environment: Set up Arduino IDE with correct board settings and libraries.
(4) Programming Mode: Ensure the board is in programming (bootloader) mode for code uploading via USB.
(5) Serial Communication: Use a serial monitor (e.g., Arduino Serial Monitor) for debugging and data exchange.
(6) Baud Rate: Set baud rate to match board settings (typically 115200 or 9600 baud).