INTERNET OF THINGS (IoT) - VIVA PREPARATION GUIDE

T.Y.BSc.DS - Academic Year 2025-26

⚡ PRACTICAL 1: STARTING RASPBIAN OS, FAMILIARISING WITH RASPBERRY PI COMPONENTS AND CONNECTING TO ETHERNET, MONITOR, USB

Key Definitions:

Raspberry Pi: A low-cost, credit-card sized single-board computer developed by Raspberry Pi Foundation that plugs into a monitor and uses a keyboard and mouse.

Raspbian OS (Raspberry Pi OS): The official Linux-based operating system for Raspberry Pi, based on Debian. Designed specifically to run on Raspberry Pi hardware.

Single Board Computer (SBC): A complete computer built on a single circuit board with microprocessor, memory, input/output, and other features.

GPIO (General Purpose Input/Output): Programmable pins on Raspberry Pi used to communicate with electronic components like LEDs, sensors, and motors.

SSH (Secure Shell): A network protocol that allows secure remote login and control of Raspberry Pi from another computer.

HDMI: High-Definition Multimedia Interface - used to connect Raspberry Pi to a monitor or TV.

SD Card: Storage medium used in Raspberry Pi instead of a hard drive, contains the OS.

Expected Viva Questions:

1. **Q: What is Raspberry Pi?** A: Raspberry Pi is a small, affordable single-board computer developed by the Raspberry Pi Foundation in the UK. It runs Linux-based OS and is used for learning programming and IoT applications.

2. **Q: What is Raspbian OS?** A: Raspbian (now called Raspberry Pi OS) is the official operating system for Raspberry Pi. It is based on Debian Linux and comes with pre-installed software for education and programming.

3. **Q: List the main components of Raspberry Pi.** A:
   - Processor (ARM CPU)

- RAM
- GPIO pins (40 pins)
- HDMI port
- USB ports (2-4)
- Ethernet port
- MicroSD card slot
- Audio jack
- Camera and Display ports (CSI/DSI)
- Wi-Fi and Bluetooth (Pi 3/4)

4. **Q: What is the purpose of GPIO pins?** A: GPIO (General Purpose Input/Output) pins allow the Raspberry Pi to interface with external hardware components like LEDs, sensors, motors, and other circuits. They can be configured as input or output.

5. **Q: How many GPIO pins does Raspberry Pi have?** A: Raspberry Pi has 40 GPIO pins arranged in 2 rows of 20 pins each.

6. **Q: What is the difference between Raspberry Pi and Arduino?** A: | Raspberry Pi | Arduino | |—|—| | Single-board computer | Microcontroller board | | Runs full OS (Linux) | No OS (direct code) | | High processing power | Low processing power | | Great for complex apps | Great for simple tasks | | Has RAM, storage | Minimal memory |

7. **Q: What is SSH and how is it used with Raspberry Pi?** A: SSH (Secure Shell) is a network protocol that allows you to remotely log into and control Raspberry Pi from another computer over a network, without needing a physical keyboard/monitor.

8. **Q: What type of power supply does Raspberry Pi use?** A: Raspberry Pi typically uses a 5V micro-USB or USB-C power supply. Pi 3 needs at least 2.5A and Pi 4 needs at least 3A current.

9. **Q: What is the operating system stored on in Raspberry Pi?** A: The OS is stored on a microSD card. Minimum 8GB card is recommended, 16GB+ is preferred.

10. **Q: How do you connect Raspberry Pi to the internet?** A:

    - Through Ethernet cable (wired)
    - Through Wi-Fi (wireless) - available on Pi 3/4
    - Through USB Wi-Fi dongle

11. **Q: What is VNC and how is it used?** A: VNC (Virtual Network Computing) allows you to remotely access and control the Raspberry Pi's graphical desktop from another computer.

12. **Q: What are the different models of Raspberry Pi?** A: Raspberry Pi 1, 2, 3, 4, 5, Zero, Zero W, Pico. Each has different specs, speeds, and features.

⚡ PRACTICAL 2: DISPLAYING DIFFERENT LED PATTERNS WITH ARDUINO

Key Definitions:

Arduino: An open-source electronics platform with easy-to-use hardware and software used for building digital devices and interactive objects.

Arduino UNO: The most popular Arduino board based on the ATmega328P microcontroller.

LED (Light Emitting Diode): A semiconductor device that emits light when current flows through it.

Resistor: A component that limits current flow. Used with LEDs to prevent them from burning out.

Digital Pin: Arduino pin that can be either HIGH (5V) or LOW (0V).

Sketch: The name given to a program written for Arduino.

void setup(): Function that runs once when Arduino starts, used for initialization.

void loop(): Function that runs repeatedly in a continuous loop.

digitalWrite(): Function to set a digital pin HIGH or LOW.

delay(): Function to pause the program for a specified number of milliseconds.

Breadboard: A solderless device used to create temporary circuits for prototyping.

Expected Viva Questions:

1. **Q: What is Arduino?** A: Arduino is an open-source electronics platform combining easy-to-use hardware (microcontroller board) and software (IDE). It's used to create interactive electronic projects and prototype IoT devices.

2. **Q: What is Arduino UNO?** A: Arduino UNO is the most popular Arduino board, based on the ATmega328P microcontroller. It has 14 digital I/O pins, 6 analog input pins, a 16MHz oscillator, USB connection, and a power jack.

3. **Q: What is an LED and how does it work?** A: LED (Light Emitting Diode) is a semiconductor that emits light when current flows through it from anode (+) to cathode (-). It requires a resistor in series to limit current.

4. **Q: Why is a resistor used with LED?** A: A resistor limits the current flowing through the LED. Without it, excessive current would destroy the LED. Typical value is 220Ω to 330Ω for 5V systems.

5. **Q: What is the structure of an Arduino program?** A:

```
void setup() {
  // Runs once - initialize pins, serial, etc.
  pinMode(13, OUTPUT);
}
void loop() {
  // Runs forever - main logic
  digitalWrite(13, HIGH);
  delay(1000);
  digitalWrite(13, LOW);
  delay(1000);
}
```

6. **Q: Explain the code to blink an LED.** A:

```
void setup() {
  pinMode(13, OUTPUT);  // Set pin 13 as output
}
void loop() {
  digitalWrite(13, HIGH);  // Turn LED ON
  delay(1000);              // Wait 1 second
  digitalWrite(13, LOW);   // Turn LED OFF
  delay(1000);              // Wait 1 second
}
```

7. **Q: What does pinMode() do?** A: pinMode() configures a specific pin as either INPUT or OUTPUT. Example: `pinMode(13, OUTPUT);`

8. **Q: What is the difference between HIGH and LOW in Arduino?** A: HIGH means 5V (pin is ON), LOW means 0V (pin is OFF).

9. **Q: How many digital pins does Arduino UNO have?** A: 14 digital I/O pins (pins 0-13). 6 of them (pins 3,5,6,9,10,11) support PWM.

10. **Q: What is PWM?** A: PWM (Pulse Width Modulation) is a technique to simulate analog output by rapidly switching a digital signal between HIGH and LOW, used to control LED brightness and motor speed.

11. **Q: How do you control LED brightness?** A: Using analogWrite() with PWM pins (values 0-255): `cpp  analogWrite(9, 127);  // 50% brightness`

12. **Q: What is the built-in LED pin of Arduino UNO?** A: Pin 13 has a built-in LED on Arduino UNO.

13. **Q: How do you connect multiple LEDs to Arduino?** A: Each LED connects from its own digital pin through a resistor to ground (GND). Each LED needs its own resistor.

14. **Q: What is a breadboard?** A: A breadboard is a solderless prototyping board where components can be inserted and connected without soldering, ideal for temporary circuits.

⚡ PRACTICAL 3: DISPLAYING TIME OVER 4-DIGIT 7-SEGMENT DISPLAY

Key Definitions:

7-Segment Display: An electronic display device that uses 7 LED segments (a,b,c,d,e,f,g) to display digits 0-9 and some characters.

4-Digit 7-Segment Display: Contains four 7-segment displays controlled by multiplexing to show 4-digit numbers like time (HH:MM).

Multiplexing: A technique to control multiple displays with fewer pins by rapidly switching between digits at high speed.

Common Cathode: All LED cathodes are connected together; segments are turned ON by HIGH signal.

Common Anode: All LED anodes are connected together; segments are turned ON by LOW signal.

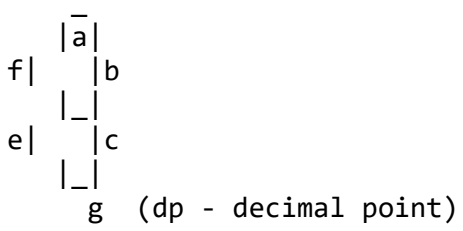
TM1637: A popular driver chip for 4-digit 7-segment displays, requiring only 2 wires (CLK and DIO).

Real Time Clock (RTC): A module (e.g., DS1307, DS3231) that keeps accurate time even when the main power is off.

Expected Viva Questions:

1. **Q: What is a 7-segment display?** A: A 7-segment display is an electronic display using 7 LED segments (labeled a through g) arranged to display digits 0-9 and letters A-F. A decimal point (dp) is sometimes the 8th segment.

2. **Q: Name the 7 segments and draw them.** A:



Segments: a (top), b (top-right), c (bottom-right), d (bottom), e (bottom-left), f (top-left), g (middle)

3. **Q: What is the difference between common anode and common cathode?** A:
   - **Common Cathode**: All negative (cathode) terminals share a common ground. Send HIGH to turn on a segment.
   - **Common Anode**: All positive (anode) terminals share a common 5V. Send LOW to turn on a segment.
4. **Q: How does multiplexing work in a 4-digit display?** A: Multiplexing rapidly turns on one digit at a time in sequence (faster than human eye can see ~50+ times/sec). This makes all 4 digits appear ON simultaneously while using fewer control pins.
5. **Q: What is the TM1637 module?** A: TM1637 is an integrated circuit that drives a 4-digit 7-segment display. It communicates with Arduino using just 2 wires - CLK (clock) and DIO (data).
6. **Q: How do you display time using Arduino?** A: Using a RTC module (DS1307/DS3231) to get current time, then displaying hours and minutes on the 4-digit 7-segment display through TM1637 library.
7. **Q: What library is used for TM1637?** A: `TM1637Display` library. Include it as:

   ```
   #include <TM1637Display.h>
   ```

8. **Q: What is a Real Time Clock (RTC)?** A: An RTC is a clock module that tracks actual time even when the microcontroller is off, using its own battery. Common models: DS1307 (I2C), DS3231.
9. **Q: Which segments light up for digit "8"?** A: All 7 segments (a, b, c, d, e, f, g) light up to display "8".
10. **Q: Which segments light up for digit "1"?** A: Segments b and c light up to display "1".
11. **Q: What protocol does the RTC module use to communicate?** A: I2C (Inter-Integrated Circuit) protocol using SDA (data) and SCL (clock) lines.
12. **Q: How many pins does a 7-segment display have?** A: A single 7-segment display has 10 pins - 7 for segments + 1 for decimal point + 2 common pins.

⚡ PRACTICAL 4: IMPLEMENT ARDUINO UNO BASED OSCILLOSCOPE

Key Definitions:

Oscilloscope: An instrument that displays electrical signals as a waveform over time on a screen, used to analyze signal behavior.

Analog-to-Digital Converter (ADC): Converts analog voltage to digital values. Arduino UNO has a 10-bit ADC (0-1023 values for 0-5V).

Waveform: A graphical representation of a signal's variation over time.

Sampling Rate: The number of measurements taken per second (samples per second / Hz).

Analog Pin: Arduino pin (A0-A5) that reads analog voltages between 0-5V.

analogRead(): Function that reads analog voltage on analog pins, returns 0-1023.

Serial Monitor: Tool in Arduino IDE to view serial data from Arduino.

Serial.println(): Sends data from Arduino to computer via serial communication.

Processing: A programming language/environment used to create visualizations from serial data.

Expected Viva Questions:

1. **Q: What is an oscilloscope?** A: An oscilloscope is a test instrument that displays electrical signals as a waveform (voltage vs time graph), allowing engineers to analyze signal characteristics like frequency, amplitude, and shape.

2. **Q: How can Arduino act as an oscilloscope?** A: Arduino reads analog signals through its ADC on pins A0-A5, sends data via Serial communication to a computer, and software (like Processing or Serial Plotter) displays it as a waveform.

3. **Q: What is ADC?** A: ADC (Analog-to-Digital Converter) converts analog voltage into a digital number. Arduino UNO has a 10-bit ADC that converts 0-5V into values from 0 to 1023.

4. **Q: What is the resolution of Arduino's ADC?** A: 10-bit resolution → 2^10 = 1024 values (0-1023). Voltage resolution = 5V / 1024 ≈ 4.9 mV per step.

5. **Q: How do you read an analog signal in Arduino?** A:

```
int sensorValue = analogRead(A0);  // Reads 0-1023
float voltage = sensorValue * (5.0 / 1023.0);  // Convert to volt
age
```

6. **Q: What is the maximum input voltage for Arduino analog pins?** A: 0 to 5V for Arduino UNO. Do NOT exceed 5V as it will damage the board.

7. **Q: What is the Serial Monitor?** A: Serial Monitor is a built-in tool in Arduino IDE that displays text data sent from Arduino via serial communication (USB). Used for debugging and monitoring values.

8. **Q: What is sampling rate?** A: Sampling rate is the number of measurements taken per second. Higher sampling rate captures faster signals more accurately. Arduino UNO ADC can sample at ~10,000 samples/second max.

9. **Q: What types of waveforms can be analyzed?** A: Sine wave, square wave, triangle wave, sawtooth wave, pulse signals, and noise.

10. **Q: What is the Serial Plotter in Arduino IDE?** A: Serial Plotter is a tool that graphically plots values received from Arduino serial communication in real-time, useful for visualizing sensor data and waveforms.

11. **Q: What are the limitations of Arduino as an oscilloscope?** A: Limited sampling rate (~10kHz), limited voltage range (0-5V), only handles low-frequency signals, less accurate than real oscilloscopes.

12. **Q: What is the code to implement a basic oscilloscope?** A: `cpp void setup() { Serial.begin(9600); } void loop() { int reading = analogRead(A0); Serial.println(reading); delay(1); }`

⚡ PRACTICAL 5: CONTROLLING ARDUINO UNO WITH WHATSAPP

Key Definitions:

IoT (Internet of Things): A network of physical devices connected to the internet that can collect and exchange data.

ESP8266/ESP32: Low-cost Wi-Fi microchips used to give Arduino internet connectivity.

Twilio API: A cloud communication platform API that can be used to send/receive WhatsApp messages programmatically.

Callmebot API: A free API service that enables sending WhatsApp messages to Arduino/ESP devices.

Webhook: A way for apps to communicate using HTTP callbacks when events happen.

HTTP Request: A request made over the internet using HTTP protocol (GET/POST).

REST API: An interface that allows two applications to communicate over the internet using HTTP.

WiFi.h: Arduino library for Wi-Fi connectivity (ESP8266/ESP32).

Expected Viva Questions:

1. **Q: How can you control Arduino with WhatsApp?** A: By connecting Arduino to internet via ESP8266/ESP32 Wi-Fi module, and using WhatsApp APIs (like Callmebot) or Twilio to send/receive messages that trigger actions on Arduino.

2. **Q: What is IoT?** A: Internet of Things (IoT) is a system of interconnected physical devices (sensors, actuators, machines) embedded with software and connectivity to collect and exchange data over the internet.

3. **Q: What is ESP8266?** A: ESP8266 is a low-cost Wi-Fi microchip made by Espressif that gives Arduino internet connectivity. It has its own microprocessor and can run Arduino code directly.

4. **Q: What is an API?** A: API (Application Programming Interface) is a set of rules/protocols that allows different software applications to communicate with each other. WhatsApp uses API for messaging integration.

5. **Q: What is Callmebot?** A: Callmebot is a free API service that allows you to send WhatsApp messages to your phone, and you can integrate it with microcontrollers like ESP8266/ESP32 to control devices.

6. **Q: What protocol is used to make API calls?** A: HTTP/HTTPS protocol. Typically GET or POST requests.

7. **Q: What is a webhook?** A: A webhook is an HTTP callback that occurs when something happens. When a WhatsApp message is received, a webhook triggers a function to process the message and take action.

8. **Q: What hardware is required for this practical?** A: Arduino UNO + ESP8266 (or ESP32), components to control (LED, relay, motor), power supply, internet connection.

9. **Q: What are real-world applications of WhatsApp-controlled IoT?** A: Home automation, remote switching, security alerts, monitoring systems, smart appliances, industrial automation.

10. **Q: What is NodeMCU?** A: NodeMCU is an open-source IoT platform based on the ESP8266 chip. It runs on the ESP-12 module and can be programmed using Arduino IDE or Lua language.

⚡ PRACTICAL 6: FINGERPRINT SENSOR INTERFACING WITH ARDUINO UNO

Key Definitions:

Fingerprint Sensor: A biometric device that captures and identifies fingerprints for security and authentication.

Biometrics: Measurement and analysis of unique physical characteristics (fingerprint, face, iris) used for authentication.

Optical Fingerprint Sensor: Uses light to capture a fingerprint image (e.g., R305, R307, R503).

Capacitive Fingerprint Sensor: Uses electrical signals to map fingerprint ridges and valleys.

Enrollment: The process of registering a fingerprint in the sensor's memory for future comparison.

Verification: Comparing a scanned fingerprint to stored templates (1:1 comparison).

Identification: Searching all stored templates to find a match (1:N comparison).

Adafruit_Fingerprint.h: Arduino library for interfacing with fingerprint modules.

UART (Universal Asynchronous Receiver/Transmitter): Serial communication protocol used by fingerprint sensors to communicate with Arduino.

Expected Viva Questions:

1. **Q: What is a fingerprint sensor?** A: A fingerprint sensor is a biometric device that captures a user's fingerprint image, converts it into a digital template, and stores it for future comparison and authentication.
2. **Q: What are the types of fingerprint sensors?** A:
   - **Optical**: Uses light and camera to capture fingerprint image
   - **Capacitive**: Uses electrical signals; most secure
   - **Ultrasonic**: Uses sound waves; works through dirt/water
   - **Thermal**: Uses temperature differences
3. **Q: Name common fingerprint sensor modules used with Arduino.** A: R305, R307, R503, AS608, FPM10A, GT-521F32.
4. **Q: What is the communication protocol of fingerprint sensors?** A: UART (Serial communication) using TX and RX pins at typically 57600 baud rate.

5. **Q: What library is used for fingerprint sensor in Arduino?** A: `Adafruit_Fingerprint.h` library:

```
#include <Adafruit_Fingerprint.h>
```

6. **Q: Explain the enrollment process.** A:

    1. Place finger on sensor (1st scan)
    2. Sensor captures image
    3. Lift finger
    4. Place same finger again (2nd scan)
    5. Both images compared and template created
    6. Template stored with an ID number

7. **Q: What is the difference between verification and identification?** A:

    - **Verification (1:1)**: Compares finger to ONE specific stored template (e.g., ID 5)
    - **Identification (1:N)**: Scans all stored templates to find ANY match

8. **Q: What are the real-world applications of fingerprint sensors?** A: Attendance systems, door lock systems, banking, mobile phone security, exam authentication, access control.

9. **Q: How many templates can common fingerprint modules store?** A: Most common modules store 127-1000 fingerprint templates in their internal flash memory.

10. **Q: How do you connect a fingerprint sensor to Arduino?** A:

    - Sensor VCC → Arduino 5V
    - Sensor GND → Arduino GND
    - Sensor TX → Arduino digital pin (e.g., pin 2)
    - Sensor RX → Arduino digital pin (e.g., pin 3) Use SoftwareSerial for the connection.

11. **Q: What is biometrics?** A: Biometrics is the measurement and statistical analysis of people's unique physical and behavioral characteristics like fingerprints, face, iris, voice for authentication purposes.

12. **Q: What is FAR and FRR in biometrics?** A:

    - **FAR (False Acceptance Rate)**: Wrong person accepted as valid
    - **FRR (False Rejection Rate)**: Valid person rejected as invalid

---

⚡ PRACTICAL 7: ARDUINO UNO GPS MODULE INTERFACING

Key Definitions:

GPS (Global Positioning System): A satellite-based navigation system that provides location and time information anywhere on Earth.

GPS Module: A hardware device that receives signals from GPS satellites to determine position (latitude, longitude, altitude).

NMEA (National Marine Electronics Association): A standard format for GPS data strings ($GPGGA, $GPRMC, etc.).

Latitude: Angular distance north or south of equator (0° to ±90°).

Longitude: Angular distance east or west of prime meridian (0° to ±180°).

NEO-6M: A popular, affordable GPS module widely used with Arduino.

TinyGPS++ Library: Arduino library for parsing GPS NMEA data.

Geofencing: Using GPS to define virtual geographic boundaries and trigger events.

Baud Rate: Speed of serial communication (GPS typically uses 9600 bps).

Expected Viva Questions:

1. **Q: What is GPS?** A: GPS (Global Positioning System) is a U.S. satellite-based navigation system that uses a network of 24+ satellites orbiting Earth to provide accurate location (latitude, longitude, altitude) and time information anywhere on Earth.

2. **Q: How does GPS work?** A: A GPS receiver picks up signals from at least 4 satellites. By calculating the time for signals to travel from each satellite, it determines its position through a process called trilateration.

3. **Q: What is the NEO-6M GPS module?** A: NEO-6M is a popular, compact GPS module made by u-blox. It communicates via UART (serial) at 9600 baud and outputs NMEA data strings with position and time information.

4. **Q: What is NMEA format?** A: NMEA (National Marine Electronics Association) is a standard data format for GPS output. Data comes as sentences like:

   - **$GPGGA** - Fix data (position, time, altitude)
   - **$GPRMC** - Recommended minimum data (position, speed)
   - **$GPGSV** - Satellite information

5. **Q: What library is used for GPS in Arduino?** A: `TinyGPS++` library:

```
#include <TinyGPSPlus.h>
```

6. **Q: What is latitude and longitude?** A:
   - **Latitude**: North/South position (0° at equator, ±90° at poles)
   - **Longitude**: East/West position (0° at Greenwich, ±180°) Together they uniquely identify any location on Earth.
7. **Q: How do you connect a GPS module to Arduino?** A:
   - GPS VCC → 3.3V or 5V
   - GPS GND → Arduino GND
   - GPS TX → Arduino RX (or SoftwareSerial)
   - GPS RX → Arduino TX (or SoftwareSerial)
8. **Q: How many satellites does GPS need for accurate positioning?** A: Minimum 4 satellites for 3D positioning (latitude, longitude, altitude). 3 satellites give 2D position only.
9. **Q: What is geofencing?** A: Geofencing is a virtual boundary drawn around a geographic area. When a GPS device enters or leaves this area, an alert or action is triggered. Used in vehicle tracking, child safety, fleet management.
10. **Q: What are the applications of GPS in IoT?** A: Vehicle tracking, fleet management, asset tracking, navigation apps, weather balloons, drone navigation, animal tracking, smart agriculture.
11. **Q: What is the difference between GPS accuracy in open sky vs indoor?** A: GPS works best outdoors with clear sky view (~3-10m accuracy). Indoors, signals are blocked by buildings causing poor or no GPS fix.
12. **Q: How do you extract latitude and longitude from GPS in Arduino?** A:

```
cpp     TinyGPSPlus gps;     if (gps.location.isValid())
{       float lat = gps.location.lat();       float lng =
gps.location.lng();       Serial.print("Lat: ");
Serial.println(lat, 6);       Serial.print("Lng: ");
Serial.println(lng, 6);     }
```

⚡ PRACTICAL 8: IoT BASED WEB CONTROLLED HOME AUTOMATION USING ARDUINO UNO

Key Definitions:

Home Automation: Use of smart technology to control home appliances and devices automatically or remotely.

Web Server: Software that serves web pages over the internet. Arduino/ESP can act as a mini web server.

HTML: HyperText Markup Language - the standard language for creating web pages.

Wi-Fi Module (ESP8266/ESP32): Provides Arduino with internet/Wi-Fi connectivity.

HTTP (HyperText Transfer Protocol): Protocol used to communicate between web browsers and servers.

IP Address: A unique numerical label assigned to each device on a network (e.g., 192.168.1.100).

Relay Module: An electrically operated switch that allows Arduino to control high-voltage AC appliances safely.

IoT Platform: Cloud-based service for IoT device management (e.g., Blynk, ThingSpeak, IFTTT, Cayenne).

Blynk: A popular IoT platform with mobile app that lets you control Arduino over internet.

Expected Viva Questions:

1. **Q: What is home automation?** A: Home automation is the use of smart technology to automatically control home appliances, systems (lighting, HVAC, security), allowing remote or automated management via smartphones, voice, or programmed schedules.

2. **Q: How does IoT-based home automation work?** A: Sensors and actuators in the home connect to Arduino/Raspberry Pi with Wi-Fi module. These devices connect to cloud platforms or local servers. Users control them through mobile apps or web interfaces via internet.

3. **Q: What is a relay module?** A: A relay is an electrically-operated switch. A relay module allows low-voltage Arduino (5V) to safely switch high-voltage AC devices (fans, lights, TVs) ON/OFF. It acts as an interface between Arduino and mains electricity.

4. **Q: How does a relay work?** A: When Arduino sends HIGH signal to relay, an electromagnet closes the switch connecting the AC load circuit. When signal goes LOW, the switch opens, disconnecting the load.

5. **Q: What is ESP8266 and how is it used in home automation?** A: ESP8266 is a Wi-Fi microchip that gives Arduino internet connectivity. With it, the Arduino can host a web server or connect to cloud platforms to receive control commands remotely.

6. **Q: What is a web server?** A: A web server is software/hardware that accepts HTTP requests and sends back HTML web pages. ESP8266/ESP32 can act as a mini web server to serve control interface pages.

7. **Q: What is Blynk?** A: Blynk is an IoT platform with a drag-and-drop mobile app builder. It allows users to create a smartphone app to control Arduino/ESP devices over the internet.

8. **Q: What is IFTTT?** A: IFTTT (If This Then That) is a free automation platform that connects different apps and devices. Example: "If temperature > 30°C, then turn on fan".

9. **Q: What is ThingSpeak?** A: ThingSpeak is an IoT analytics platform by MathWorks that allows you to collect, analyze, and visualize data from IoT sensors in the cloud.

10. **Q: What are the components needed for IoT home automation?** A: Arduino UNO, ESP8266/ESP32, relay module, appliances (LED/fan/bulb), power supply, Wi-Fi router, smartphone with app.

11. **Q: What is MQTT protocol?** A: MQTT (Message Queuing Telemetry Transport) is a lightweight publish-subscribe messaging protocol designed for IoT devices with limited resources and low bandwidth. Uses Broker-Publisher-Subscriber model.

12. **Q: What is the difference between Wi-Fi and Bluetooth in IoT?** A: | Wi-Fi | Bluetooth | |——-|———–| | Longer range (~100m) | Short range (~10m) | | Higher power usage | Lower power usage | | Internet connectivity | Device-to-device only | | Higher data rates | Lower data rates |

⚡ PRACTICAL 9: INTERFACING ARDUINO UNO WITH RFID

Key Definitions:

RFID (Radio Frequency Identification): Technology that uses electromagnetic fields to identify and track tags attached to objects.

RFID Tag: A small electronic device (card, key fob, sticker) containing a chip and antenna that transmits data when powered by RFID reader.

RFID Reader/Module: A device that emits radio waves to power and read RFID tags.

RC522: Popular Arduino-compatible RFID reader module operating at 13.56MHz.

UID (Unique Identifier): The unique identification number stored on an RFID tag.

SPI (Serial Peripheral Interface): Communication protocol used by RC522 to communicate with Arduino.

Active RFID: Tags have their own battery and transmit continuously (longer range).

Passive RFID: Tags have no battery; powered by reader's radio waves (shorter range, cheaper).

NFC (Near Field Communication): A subset of RFID operating at 13.56MHz for very short range (~4cm).

Expected Viva Questions:

1. **Q: What is RFID?** A: RFID (Radio Frequency Identification) is a wireless technology that uses electromagnetic fields to automatically identify and track tags attached to objects. The tag contains electronically stored information read by an RFID reader.

2. **Q: How does RFID work?** A: An RFID reader emits radio waves that power a passive RFID tag. The tag's antenna receives energy, powers the chip, and transmits its stored UID back to the reader. The reader captures and processes this data.

3. **Q: What are the types of RFID?** A:

   - **Low Frequency (LF)**: 125-134 kHz, short range, animal tracking
   - **High Frequency (HF)**: 13.56 MHz, ~1m range (RC522, NFC cards)
   - **Ultra High Frequency (UHF)**: 860-960 MHz, longer range, supply chain

4. **Q: What is the RC522 module?** A: RC522 is an RFID reader/writer module operating at 13.56MHz. It uses SPI protocol to communicate with Arduino and can read/write MIFARE RFID cards and key fobs.

5. **Q: What is the difference between active and passive RFID?** A: | Active RFID | Passive RFID | |—|—| | Has own battery | No battery | | Long range (100m+) | Short range (1-10m) | | Expensive | Cheap | | Transmits continuously | Powered by reader |

6. **Q: What is UID?** A: UID (Unique Identifier) is a unique number programmed into an RFID tag during manufacturing. It's used to identify each tag uniquely, like a serial number.

7. **Q: Which communication protocol does RC522 use?** A: SPI (Serial Peripheral Interface) protocol with 4 lines: MOSI, MISO, SCK, SS.

8. **Q: What library is used for RFID in Arduino?** A: MFRC522 library:

   ```
   #include <MFRC522.h>
   #include <SPI.h>
   ```

9. **Q: How do you connect RC522 to Arduino UNO?** A:

   - SDA → Pin 10

- SCK → Pin 13
- MOSI → Pin 11
- MISO → Pin 12
- GND → GND
- RST → Pin 9
- 3.3V → 3.3V (NOT 5V!)

10. **Q: What are real-world applications of RFID?** A: Access control, attendance systems, library book tracking, supermarket checkout, animal tracking, passport/ID cards, toll collection, contactless payments.

11. **Q: What is NFC and how is it related to RFID?** A: NFC (Near Field Communication) is a short-range (4cm) wireless technology based on RFID standards operating at 13.56MHz. Used in smartphones for contactless payments, file transfer.

12. **Q: How can RFID be used for attendance system?** A: Each student has an RFID card with unique UID. When card is tapped on reader, Arduino reads UID, matches it against a registered database, marks attendance, and displays name/time on LCD.

⚡ PRACTICAL 10: INSTALLING WINDOWS 10 IoT CORE ON RASPBERRY PI

Key Definitions:

Windows 10 IoT Core: A version of Windows 10 designed for small embedded devices like Raspberry Pi. It is headless (no standard UI) or uses Universal Windows Platform apps.

IoT Core: A trimmed-down version of Windows optimized for IoT devices with limited resources.

Universal Windows Platform (UWP): A development platform for building apps that run across all Windows 10 devices including IoT.

Windows IoT Dashboard: A Windows application used to setup, manage, and connect to Windows IoT devices.

Headless Mode: Device runs without a display/monitor (controlled remotely).

Headed Mode: Device runs with a display, showing apps on screen.

Device Portal: A web-based management interface for Windows IoT Core devices, accessible through a browser.

WinSCP: A Windows application for secure file transfer to/from IoT device.

Visual Studio: Microsoft's IDE used to develop and deploy UWP apps to Windows IoT Core.

Expected Viva Questions:

1. **Q: What is Windows 10 IoT Core?** A: Windows 10 IoT Core is a version of Windows 10 optimized for smaller devices like Raspberry Pi. It is designed for IoT applications, supports Universal Windows Platform (UWP) apps, and can run headless or with display.

2. **Q: What is the difference between Windows 10 IoT Core and regular Windows 10?** A: | Windows 10 IoT Core | Windows 10 (Desktop) | |—|—| | Minimal OS for IoT devices | Full-featured desktop OS | | Runs one UWP app at a time | Runs multiple apps | | No Start menu or desktop | Full GUI with desktop | | For ARM/small devices | For PCs | | Free to use | Licensed |

3. **Q: How do you install Windows 10 IoT Core on Raspberry Pi?** A:

    1. Download Windows IoT Dashboard
    2. Connect microSD card to PC
    3. In Dashboard, select "Setup a new device"
    4. Choose Raspberry Pi and Windows IoT Core
    5. Flash the image to SD card
    6. Insert SD card into Pi and power on
    7. Connect via Device Portal or Dashboard

4. **Q: What is the Windows IoT Dashboard?** A: Windows IoT Dashboard is a free Windows application that simplifies setting up, configuring, and managing Windows IoT Core devices. It helps flash OS to SD card and discover devices on the network.

5. **Q: What is the Device Portal?** A: Device Portal is a web-based management interface for Windows IoT Core. Accessible at device's IP address, it allows managing apps, settings, debugging, and monitoring without physical access.

6. **Q: What are UWP apps?** A: UWP (Universal Windows Platform) apps are applications built with Microsoft's framework that run on all Windows 10 devices - PC, tablet, phone, IoT devices - using the same code base.

7. **Q: What is headless mode in Windows IoT Core?** A: Headless mode means the device runs without a display. All management is done remotely through the network (Device Portal, PowerShell, SSH).

8. **Q: What programming languages can be used with Windows IoT Core?** A: C#, C++, Python, and JavaScript (Node.js) are supported through Visual Studio and other development tools.

9. **Q: What are the system requirements for Windows 10 IoT Core on Raspberry Pi?** A:
    - Raspberry Pi 2, 3, or 4
    - MicroSD card (8GB minimum, class 10)
    - Windows PC (for setup)
    - Internet connection
    - Optional: HDMI monitor, USB keyboard
10. **Q: What are advantages of Windows IoT Core over Raspbian?** A:
    - Familiar Windows environment for .NET developers
    - Integration with Azure IoT Hub
    - Visual Studio development support
    - UWP app ecosystem But Raspbian has more community support and software.
11. **Q: What is Azure IoT Hub?** A: Azure IoT Hub is Microsoft's cloud service for bidirectional communication between IoT applications and devices. It allows connecting, monitoring, and managing IoT devices at scale.
12. **Q: How do you deploy an app to Windows IoT Core?** A: Using Visual Studio, select the target as Remote Machine and enter device's IP address. Visual Studio deploys the UWP app wirelessly to the device.

GENERAL IoT CONCEPTS - MUST KNOW FOR VIVA

Key Definitions:

IoT (Internet of Things): Network of physical objects embedded with sensors, software, and connectivity to collect and exchange data over the internet.

Sensor: A device that detects and measures physical parameters (temperature, pressure, light).

Actuator: A device that performs physical action based on electrical signals (motor, LED, relay).

Protocol: Rules that define how data is transmitted between devices (HTTP, MQTT, CoAP, Zigbee).

Cloud Computing: Delivering computing services (storage, processing) over the internet.

Edge Computing: Processing data near the source (device) rather than sending all to cloud.

Firmware: Permanent software stored on a microcontroller's flash memory.

Microcontroller: A small computer on a single IC containing processor, memory, and I/O (e.g., ATmega328P in Arduino).

Expected Viva Questions:

1. **Q: What is IoT?** A: IoT (Internet of Things) is a network of physical devices (things) embedded with sensors, software, and connectivity that enables them to collect and exchange data over the internet without human interaction.

2. **Q: What are the layers of IoT architecture?** A:

   - **Layer 1 - Perception Layer**: Physical sensors, devices, data collection
   - **Layer 2 - Network Layer**: Internet connectivity, gateways, routers
   - **Layer 3 - Processing Layer**: Data storage, cloud processing
   - **Layer 4 - Application Layer**: User interfaces, apps, services

3. **Q: Give examples of IoT applications.** A:

   - **Smart Home**: Smart lighting, AC, security cameras
   - **Healthcare**: Wearables, patient monitoring
   - **Agriculture**: Soil sensors, automated irrigation
   - **Industry**: Factory automation, predictive maintenance
   - **Transport**: Vehicle tracking, smart traffic lights

4. **Q: What is the difference between microcontroller and microprocessor?** A: | Microcontroller | Microprocessor | |—|—| | All-in-one (CPU+RAM+ROM+I/O) | CPU only | | Designed for specific tasks | General purpose | | Low power | Higher power | | e.g., ATmega328 (Arduino) | e.g., Intel i5 |

5. **Q: What is MQTT and why is it used in IoT?** A: MQTT (Message Queuing Telemetry Transport) is a lightweight publish-subscribe protocol designed for IoT. Used because it requires minimal bandwidth, works on unreliable networks, and is perfect for resource-constrained devices.

6. **Q: What communication protocols are used in IoT?** A: Wi-Fi, Bluetooth, Zigbee, Z-Wave, LoRa, NB-IoT, 4G/5G, MQTT, HTTP, CoAP.

7. **Q: What is a gateway in IoT?** A: An IoT gateway is a device that connects IoT endpoints (sensors/devices) to the internet/cloud. It translates protocols between the IoT network and the internet.

8. **Q: What is edge computing?** A: Edge computing processes data at or near the IoT device itself rather than sending everything to the cloud. This reduces latency, bandwidth usage, and improves response time.

9. **Q: What are the security challenges in IoT?** A: Lack of encryption, default/weak passwords, unpatched firmware, physical security, data privacy, DDoS attacks, man-in-the-middle attacks.

10. **Q: What is a digital twin?** A: A digital twin is a virtual representation of a physical IoT device/system that simulates its real-world behavior in real-time. Used for monitoring, testing, and predictive maintenance.

COMPARISON TABLES - EXAM FAVOURITES

Arduino vs Raspberry Pi

Feature	Arduino UNO	Raspberry Pi 4
Type	Microcontroller	Single-board computer
OS	No OS	Linux / Windows IoT
Processor	ATmega328P (8-bit)	ARM Cortex-A72 (64-bit)
RAM	2KB SRAM	2/4/8 GB
Storage	32KB Flash	MicroSD card
Power	5V, low power	5V/3A, more power
Best for	Simple control tasks	Complex applications
GPIO	14 digital, 6 analog	40 GPIO pins
Cost	~₹500	~₹3000+

Wired vs Wireless IoT Communication

Feature	Wired (Ethernet)	Wireless (Wi-Fi)
Speed	High	Medium
Reliability	Very High	Moderate
Mobility	Fixed	Mobile
Setup	Complex wiring	Easy setup
Range	Limited by cable	~50-100m

IMPORTANT PROTOCOLS & PINOUTS

Arduino UNO Important Pins:

- **Pin 0 (RX), 1 (TX)** - Serial communication
- **Pin 2, 3** - Interrupt pins
- **Pin 3,5,6,9,10,11** - PWM pins (~)
- **Pin 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK)** - SPI bus
- **Pin A4 (SDA), A5 (SCL)** - I2C bus
- **Pin 13** - Built-in LED

I2C vs SPI vs UART:

Feature	I2C	SPI	UART
Wires	2 (SDA, SCL)	4 (MOSI, MISO, SCK, SS)	2 (TX, RX)
Devices	Multiple (using addresses)	Multiple (using SS)	Point-to-point
Speed	Medium	Fast	Slow-Medium
Used by	RTC, LCD, GPS	RFID, SD card	GPS, Fingerprint

QUICK FORMULA & VALUES REFERENCE

Item	Value
Arduino UNO Voltage	5V
Raspberry Pi GPIO Voltage	3.3V
Arduino ADC Resolution	10-bit (0-1023)
Standard LED Resistor	220Ω to 330Ω
GPS Default Baud Rate	9600 bps
RC522 RFID Frequency	13.56 MHz
RFID (LF) Frequency	125-134 kHz
Arduino Flash Memory	32KB
Arduino SRAM	2KB
Arduino EEPROM	1KB
Arduino Clock Speed	16 MHz

✓ TOP 20 MOST LIKELY VIVA QUESTIONS - KNOW THESE BY HEART

1. What is IoT? Give examples.
2. What is the difference between Arduino and Raspberry Pi?
3. What is GPIO? How many pins does Raspberry Pi have?
4. Why do we connect a resistor with LED?
5. What is a 7-segment display and name its 7 segments?
6. What is the difference between common anode and common cathode?
7. What is ADC? What is the resolution of Arduino's ADC?
8. What is RFID? How does it work?
9. What is the difference between active and passive RFID?
10. What is a fingerprint sensor and how does it work?
11. What is a relay module and why is it used?
12. What is ESP8266 and why is it used?
13. What is GPS? What is NMEA format?
14. What is Blynk / MQTT / ThingSpeak?
15. What is Windows 10 IoT Core?

16. What is PWM? How is it used to control LED brightness?
17. What communication protocols are used by RC522, GPS, RTC?
18. What is an oscilloscope? How does Arduino act as one?
19. What is SPI protocol?
20. What are the real-world applications of IoT home automation?

✅ PRO TIPS FOR IoT VIVA

1. **Always start with a clear definition**
2. **Give a practical example or real-world application**
3. **Know the pin connections for all modules**
4. **Remember the libraries used for each module**
5. **Understand what each component does physically**
6. **Be ready to explain the circuit you built**
7. **Know the difference between similar concepts** (e.g., RFID vs NFC, I2C vs SPI)
8. **Draw diagrams when explaining circuits**
9. **Know both code and hardware aspects**
10. **Stay calm and be confident!**

✅ FINAL CHECKLIST BEFORE VIVA

✓ Review all 10 practical aims ✓ Know what each sensor/module does ✓ Memorize key definitions ✓ Know pin connections for each module ✓ Know the library names used ✓ Understand the difference between similar concepts ✓ Be ready to explain your circuit ✓ Know IoT applications of each practical ✓ Review Arduino vs Raspberry Pi differences ✓ Get good sleep before exam!

BEST OF LUCK FOR YOUR IoT VIVA!

"IoT connects the physical world to the digital world. Understanding this connection is the key to your exam!"