INTERNET OF THINGS(IoT)

Experiment No. - 1

Q. Prepare a list of platforms used for IoT.

Prepare a list of features of above IoT platforms.

Prepare a list of features provided by python language for IoT applications.

Ans.- List of platforms used for IoT:

1. Amazon Web Services (AWS) IoT
2. Microsoft Azure IoT
3. Google Cloud IoT
4. IBM Watson IoT Platform
5. ThingWorx
6. Cisco IoT
7. Bosch IoT Suite
8. Oracle IoT
9. SAP IoT
10. GE Predix

List of features of the above IoT platforms:

1. AWS IoT: Device shadow, rule engine, device management, data analytics, security and privacy, and cloud integration.
2. Azure IoT: Device management, security, real-time analytics, event-driven actions, and cloud integration.
3. Google Cloud IoT: Device management, secure communication, data ingestion, data processing and analytics, and cloud integration.
4. IBM Watson IoT Platform: Device management, data management, analytics, security, and cloud integration.
5. ThingWorx: Rapid application development, connectivity to a variety of devices, data collection and analysis, and cloud integration.
6. Cisco IoT: Device management, security, real-time analytics, and cloud integration.
7. Bosch IoT Suite: Device management, connectivity, data collection and analysis, and cloud integration.
8. Oracle IoT: Device management, data collection and analysis, security, and cloud integration.
9. SAP IoT: Device management, data management, analytics, security, and cloud integration.
10. GE Predix: Device management, connectivity, data analysis, and cloud integration.

List of features provided by Python language for IoT applications:

1. Easy to learn and use
2. Large and active community
3. Availability of various libraries and frameworks for IoT development
4. Cross-platform support
5. Efficient memory management

6. Good support for data analysis and visualization
7. Ability to interface with hardware and sensors using modules such as GPIO, SPI, and I2C
8. Good support for networking and communication protocols such as MQTT, CoAP, and HTTP
9. Support for multi-threading and asynchronous programming
10. Availability of tools and platforms such as Flask, Django, and Jupyter Notebook for IoT development and deployment.

Experiment No. - 2

Q. Establish connectivity between various components of IoT.
Establish connection between Arduino and Bluetooth module.
Establish connection using WiFi

Ans. - To establish connectivity between various components of IoT, you will need to follow these general steps:

1. Choose a communication protocol: Decide which protocol you will use to establish communication between your IoT components. Some popular protocols include Wi-Fi, Bluetooth, Zigbee, LoRa, and MQTT.
2. Configure your hardware: Set up your hardware components to communicate with each other according to the chosen protocol. This may involve configuring networking settings, programming microcontrollers, and connecting sensors and actuators.
3. Write software to handle communication: Develop software that allows your IoT components to send and receive data. This can involve writing code for microcontrollers, cloud services, or mobile applications.
4. Test and deploy: Test your IoT system to ensure that it is working correctly and deploy it in a production environment.

To establish a connection between Arduino and Bluetooth module, follow these steps:

1. Connect the Bluetooth module to your Arduino board according to the manufacturer's instructions.
2. Install the necessary software libraries on your Arduino board to communicate with the Bluetooth module. One popular library is the SoftwareSerial library, which allows you to establish a serial connection with the Bluetooth module.
3. Write code on your Arduino board to communicate with the Bluetooth module using the installed libraries. You can use the Serial.println() function to send data to the Bluetooth module and the Serial.read() function to receive data from the module.
4. Pair the Bluetooth module with your mobile device or computer according to the manufacturer's instructions.
5. Send data between your Arduino board and the Bluetooth module using the paired device. You can use a mobile application or terminal emulator software to send and receive data.

To establish a connection using Wi-Fi, follow these steps:

1. Connect your device or microcontroller to a Wi-Fi network. Most microcontrollers, such as the ESP32 or ESP8266, have built-in Wi-Fi capabilities.
2. Configure your device to send and receive data over the Wi-Fi network. This may involve setting up networking settings, such as the SSID and password, and configuring the Wi-Fi module on your microcontroller.
3. Write software on your device or microcontroller to communicate over Wi-Fi. This can involve developing a web server or client to send and receive data using HTTP or MQTT.
4. Develop a mobile application or web application to interface with your device over Wi-Fi. You can use a variety of tools, such as the Flask or Django web frameworks, to develop your application.
5. Test your system to ensure that it is working correctly and deploy it in a production environment.

Experiment No. - 3

Q.

Publish data on the IoT platform.

Measure the temperature of a remotely located temperature sensor Using IOT based temperature data-monitoring system.

Measure the humidity of a remotely located humidity sensor Using IOT based humidity data-monitoring system.

Measure the pressure of a remotely located pressure sensor Using IOT based pressure data-monitoring System.

Ans.-

To publish data on an IoT platform, you can follow these general steps:

1. Create an account on your chosen IoT platform and set up the necessary configuration, such as the device registry and authentication settings.
2. Develop software to send data to the IoT platform. This can involve using an SDK or API provided by the platform to send data over MQTT, HTTP, or another protocol.
3. Create a device on the IoT platform to represent your connected sensor. This will typically involve registering a device ID, setting up authentication credentials, and configuring other device settings such as telemetry data formats.
4. Configure your sensor to send data to the IoT platform. This may involve setting up networking settings, such as the Wi-Fi or Ethernet connection, and configuring the sensor's firmware to send data according to the chosen protocol.
5. Test the system to ensure that data is being sent and received correctly. You can use the dashboard provided by the IoT platform to view the telemetry data.

To measure the temperature of a remotely located temperature sensor using an IoT-based temperature data-monitoring system, you can follow these steps:

1. Install a temperature sensor at the remote location, such as a thermistor or thermocouple, and connect it to a microcontroller board such as an Arduino or ESP32.
2. Set up the microcontroller board to send temperature data to an IoT platform, such as AWS IoT or Google Cloud IoT. This can involve configuring the Wi-Fi connection, installing the necessary software libraries, and writing code to send data over MQTT or HTTP.
3. Create a device on the IoT platform to represent the temperature sensor and configure the telemetry data format.
4. Test the system to ensure that the temperature data is being sent and received correctly. You can view the telemetry data on the IoT platform dashboard.

To measure the humidity of a remotely located humidity sensor using an IoT-based humidity data-monitoring system, you can follow these steps:

1. Install a humidity sensor at the remote location, such as a capacitive humidity sensor, and connect it to a microcontroller board.
2. Set up the microcontroller board to send humidity data to an IoT platform, such as AWS IoT or Google Cloud IoT. This can involve configuring the Wi-Fi connection, installing the necessary software libraries, and writing code to send data over MQTT or HTTP.
3. Create a device on the IoT platform to represent the humidity sensor and configure the telemetry data format.
4. Test the system to ensure that the humidity data is being sent and received correctly. You can view the telemetry data on the IoT platform dashboard.

To measure the pressure of a remotely located pressure sensor using an IoT-based pressure data-monitoring system, you can follow these steps:

1. Install a pressure sensor at the remote location, such as a piezoelectric sensor, and connect it to a microcontroller board.
2. Set up the microcontroller board to send pressure data to an IoT platform, such as AWS IoT or Google Cloud IoT. This can involve configuring the Wi-Fi connection, installing the necessary software libraries, and writing code to send data over MQTT or HTTP.

3. Create a device on the IoT platform to represent the pressure sensor and configure the telemetry data format.
4. Test the system to ensure that the pressure data is being sent and received correctly. You can view the telemetry data on the IoT platform dashboard.

Experiment No. - 4

Q. Download and Configure POSTMAN Application.
Verify REST APIs through POSTMAN.
Verify JSON APIs through POSTMAN.
Verify SOAP APIs through POSTMAN.

Ans.- To download and configure Postman application, you can follow these steps:

1. Go to the Postman website at https://www.postman.com/downloads/ and download the appropriate version of the application for your operating system.
2. Install the application by running the downloaded installer file and following the prompts.
3. Once the application is installed, open it and create an account or sign in with an existing account.
4. Configure any necessary settings, such as network proxies, request timeouts, or SSL certificates.

To verify REST APIs through Postman, you can follow these steps:

1. Open Postman and create a new request by clicking the "New" button in the top left corner of the screen.
2. Enter the URL of the REST API endpoint you want to test in the address bar at the top of the screen.
3. Select the appropriate HTTP method for the request, such as GET, POST, or PUT.
4. Enter any necessary request headers or parameters, such as authentication credentials or query parameters.
5. Click the "Send" button to send the request to the API endpoint.
6. Verify that the API response is returned correctly and contains the expected data.

To verify JSON APIs through Postman, you can follow these steps:

1. Open Postman and create a new request as described above.
2. Verify that the API endpoint accepts JSON data by checking the API documentation or by inspecting the request headers or parameters.
3. Create a JSON data object that matches the API endpoint's data format and include it in the request body.
4. Click the "Send" button to send the request to the API endpoint.
5. Verify that the API response is returned correctly and contains the expected JSON data.

To verify SOAP APIs through Postman, you can follow these steps:

1. Open Postman and create a new request as described above.
2. Select the "Raw" option for the request body and set the format to "XML".

3. Enter the SOAP envelope and body XML for the API request, including any necessary SOAP headers or authentication credentials.
4. Click the "Send" button to send the request to the API endpoint.
5. Verify that the API response is returned correctly and contains the expected data. You may need to parse the XML response to extract the data you are interested in.