



IOT assignment

Computer Science & Engineering (Lakshmi Narain College of Technology)



Scan to open on Studocu

Q1. To write a program to sense the available networks using Arduino.

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

void setup() {
    // initialize serial and wait for the port to open:
    Serial.begin(9600);
    while (!Serial)
        ;

    // attempt to connect using WEP encryption:
    Serial.println("Initializing Wifi...");
    printMacAddress();
    // scan for existing networks:
    Serial.println("Scanning available networks...");
    listNetworks();
}

void loop() {
    delay(10000);
    // scan for existing networks:
    Serial.println("Scanning available networks...");
    listNetworks();
}

void printMacAddress() {
    // the MAC address of your Wifi shield
    byte mac[6];
    // print your MAC address:
    WiFi.macAddress(mac);
    Serial.print("MAC: ");
    Serial.print(mac[5], HEX);
    Serial.print(":");
    Serial.print(mac[4], HEX);
    Serial.print(":");
    Serial.print(mac[3], HEX);
    Serial.print(":");
    Serial.print(mac[2], HEX);
    Serial.print(":");
    Serial.print(mac[1], HEX);
    Serial.print(":");
    Serial.print(mac[0], HEX);
}
```

```

    Serial.print(mac[1], HEX);
    Serial.print(":");
    Serial.println(mac[0], HEX);
}

void listNetworks() {
    // scan for nearby networks:
    Serial.println("** Scan Networks **");
    byte numSsid = WiFi.scanNetworks();
    // print the list of networks seen:
    Serial.print("number of available networks:");
    Serial.println(numSsid);
    // print the network number and name for each network found:
    for (int thisNet = 0; thisNet < numSsid; thisNet++) {
        Serial.print(thisNet);
        Serial.print(" ");
        Serial.print(WiFi.SSID(thisNet));
        Serial.print("\tSignal: ");
        Serial.print(WiFi.RSSI(thisNet));
        Serial.print(" dBm");
        Serial.print("\tEncryption: ");
        Serial.println(WiFi.encryptionType(thisNet));
    }
}

```

Q2. To write a program to measure the distance using ultrasonic sensor and make LED blink using Arduino.

```
// constants
const int TRIG_PIN = 6; // Arduino pin connected to Ultrasonic
Sensor's TRIG pin
const int ECHO_PIN = 7; // Arduino pin connected to Ultrasonic
Sensor's ECHO pin
const int LED_PIN = 3; // Arduino pin connected to LED's pin
const int DISTANCE_THRESHOLD = 50; // centimeters

// variables
float duration_us, distance_cm;
void setup() {
    Serial.begin(9600); // initialize serial port
    pinMode(TRIG_PIN, OUTPUT); // set arduino pin to output mode
    pinMode(ECHO_PIN, INPUT); // set arduino pin to input mode
    pinMode(LED_PIN, OUTPUT); // set arduino pin to output mode
}
void loop() {
    // generate 10-microsecond pulse to TRIG pin
    digitalWrite(TRIG_PIN, HIGH);
    delayMicroseconds(10);
    digitalWrite(TRIG_PIN, LOW);

    // measure duration of pulse from ECHO pin
    duration_us = pulseIn(ECHO_PIN, HIGH);
    // calculate the distance
    distance_cm = 0.017 * duration_us;

    if (distance_cm < DISTANCE_THRESHOLD)
        digitalWrite(LED_PIN, HIGH); // turn on LED
    else
        digitalWrite(LED_PIN, LOW); // turn off LED

    // print the value to Serial Monitor
    Serial.print("distance: ");
    Serial.print(distance_cm);
    Serial.println(" cm");
    delay(500);
}
```

Q3. To write a program to detects the vibration of an object with sensor using Arduino.

```
int vib_pin=7;
int led_pin=13;
void setup() {
  pinMode(vib_pin, INPUT);
  pinMode(led_pin, OUTPUT);
}

void loop() {
  int val;
  val=digitalRead(vib_pin);
  if(val==1)
  {
    digitalWrite(led_pin, HIGH);
    delay(1000);
    digitalWrite(led_pin, LOW);
    delay(1000);
  }
  else
  digitalWrite(led_pin, LOW);
}
```

Q4. To write a program to connect with the available Wi-Fi using Arduino

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

void setup() {
  char ssid[] = "Wifi name";
  char pass[] = "secret password";

  Serial.begin(9600);
  while (!Serial);

  int status = WL_IDLE_STATUS;
  while (status != WL_CONNECTED) {
    Serial.print("Connecting to ");
    Serial.println(ssid);
    status = WiFi.begin(ssid, pass);
    delay(5000);
  }

  Serial.print("IP address: ");
  Serial.println(WiFi.localIP());
}

void loop() {}
```

Q5. To write a program to sense a finger when it is placed on the board, Arduino.

```
#include <CapacitiveSensor.h>

const int touchPin = 2;
const int touchThreshold = 50;

CapacitiveSensor touchSensor = CapacitiveSensor(0, touchPin);

void setup() {
  Serial.begin(9600);
}

void loop() {
  long touchValue = touchSensor.capacitiveSensor(30);
  if (touchValue > touchThreshold) {
    Serial.println("Finger detected!");
    delay(1000); // wait for 1 second to prevent multiple detections
  }
}
```

Q6. To write a program to get temperature notification using Arduino.

```
#define ADC_VREF_mV 5000.0 // in millivolt
#define ADC_RESOLUTION 1024.0
#define PIN_LM35 A0

void setup() { Serial.begin(9600); }

void loop() {
    // get the ADC value from the temperature sensor
    int adcVal = analogRead(PIN_LM35);
    // convert the ADC value to voltage in millivolt
    float milliVolt = adcVal * (ADC_VREF_mV / ADC_RESOLUTION);
    // convert the voltage to the temperature in Celsius
    float tempC = milliVolt / 10;
    // convert the Celsius to Fahrenheit
    float tempF = tempC * 9 / 5 + 32;

    // print the temperature in the Serial Monitor:
    Serial.print("Temperature: ");
    Serial.print(tempC); // print the temperature in Celsius
    Serial.print("°C");
    Serial.print(" ~ "); // separator between Celsius and Fahrenheit
    Serial.print(tempF); // print the temperature in Fahrenheit
    Serial.println("°F");

    delay(1000);
}
```


Q7. To write a program for LDR to vary the light intensity of LED using Arduino.

```
int sensor=A0;
int output=9;
void setup()
{
  pinMode(output, OUTPUT);
}
void loop()
{
  int reading=analogRead(sensor);
  int bright=reading/4;
  delay(500);
  analogWrite(output, bright);
}
```

Q8. To write a program to install MySQL database in Raspberry Pi.

```
#!/bin/bash
# Update the system
sudo apt update && sudo apt upgrade -y

# Install MySQL server
sudo apt install mysql-server -y

# Secure the installation
sudo mysql_secure_installation

# Create a database and a user for it
sudo mysql -e "CREATE DATABASE raspberrypi;"
sudo mysql -e "CREATE USER 'pi'@'localhost' IDENTIFIED BY
'password';"
sudo mysql -e "GRANT ALL PRIVILEGES ON raspberrypi.* TO
'pi'@'localhost';"
sudo mysql -e "FLUSH PRIVILEGES;"

# Test the connection
mysql -u pi -p raspberrypi
```

Q9. To write a program to work with basic MySQL queries by fetching data from database in Raspberry Pi.

```
import mysql.connector

# Create a connection object to the MySQL database
conn = mysql.connector.connect(
    host="localhost",
    user="pi",
    password="password",
    database="raspberrypi"
)

# Create a cursor object to execute queries
cursor = conn.cursor()

# Execute query
cursor.execute("SELECT * FROM mytable;")
result = cursor.fetchall()

# Print the result
for row in result:
    print(row)
```

Q10. To write a program to switch light on when the input is 1 and switch the light off when the input is 0 using Raspberry Pi.

```
// Include the wiringPi library
#include <iostream>
#include <wiringPi.h>
using namespace std;

// Define the pin number for the LED
#define LED 0

// Define the main function
int main() {
    // Initialize the wiringPi library
    wiringPiSetup();

    // Set the LED pin as output
    pinMode(LED, OUTPUT);

    // Declare a variable to store the input
    int input;

    // Loop forever
    while (true) {
        // Prompt the user to enter 1 or 0
        cout << "Enter 1 to turn on the light or 0 to turn off the
light: ";
        cin >> input;

        // Check if the input is valid
        if (input == 1 || input == 0) {
            // Write the input value to the LED pin
            digitalWrite(LED, input);
        } else {
            // Display an error message
            cout << "Invalid input. Please enter 1 or 0." << endl;
        }
    }

    return 0;
}
```

