Branch	**Computer Science and Engineering,**	Programme	**B.Tech**
Course Name	**Practical Robotics Projects with Arduino**	Semester	**7th**
Course Code	**CSE-4571**	Academic Year	**2025/Odd**
Assignment-1	Topic- <u>WiFi-Communication</u>		

Learning Level (LL)	**L1**: Remembering	**L3:** Applying	**L5**: Evaluating
	L2: Understanding	**L4:** Analyzing	**L6**: Creating

To interface an ESP01 Wi-Fi module with an Arduino Uno to establish wireless communication for transmitting real-time sensor data to a remote server or monitoring system. This involves utilizing serial communication between Arduino and ESP01, configuring Wi-Fi connectivity using AT commands, and implementing TCP/IP protocols to enable accurate and reliable wireless data transmission for IoT-based applications.

| Q's	Questions	COs	LL
1	What is the primary purpose of the ESP8266 ESP-01 module when interfaced with an Arduino UNO? Ans:	**CO1**	**L1 &L2**
2	Why does the ESP-01 module operate at 3.3V instead of 5V, and what could happen if connected directly to 5V? Ans:	**CO1**	**L1 &L2**
3	List all 8 pins of the ESP-01 module and briefly describe the function of each. Ans:	**CO1**	**L1 &L2**

4	What is the default baud rate of the ESP-01 module, and why is it important to match it with the Arduino serial communication settings? Ans:	CO1	L1 &L2
5	Differentiate between Station (STA) and Access Point (AP) Wi-Fi modes, as well as both (STA+AP) Wi-Fi modes, supported by the ESP-01. Ans:	CO2	L2 & L3
6	Explain the function of the AT commands AT, AT+RST, and AT+CWJAP. Ans:	CO2	L2 & L3
7	Explain the role of the ESP01 Wi-Fi module in an IoT-based Arduino system. Ans:	CO2	L2 & L3
8	Describe how serial communication occurs between the Arduino UNO and the ESP01 module. Ans:	CO2	L2 & L3
9.	Why is a voltage divider circuit or logic level converter required when connecting Arduino TX (5V) to ESP-01 RX (3.3V)? Ans:	CO2	L2 & L3
10	What is the purpose of using the AT+CWJAP="SSID","PASSWORD"command during ESP01 configuration. Ans:	CO2	L2 & L3

Using the ESP8266 ESP-01 Wi-Fi module and Arduino UNO, demonstrate how to configure the module in Station (STA) mode to connect to an existing Wi-Fi network.
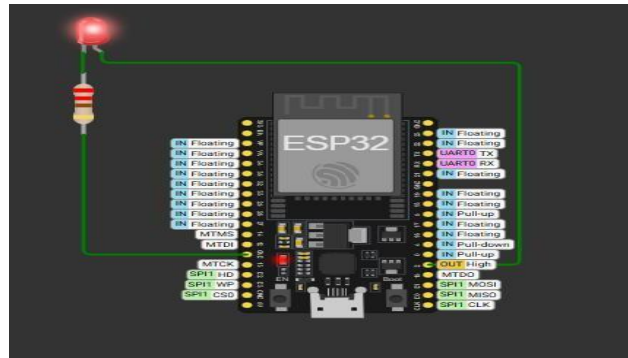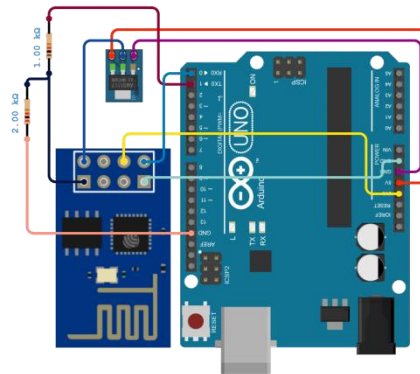

(a) Draw the circuit diagram showing the connection between Arduino UNO and ESP-01 module for Station mode configuration.
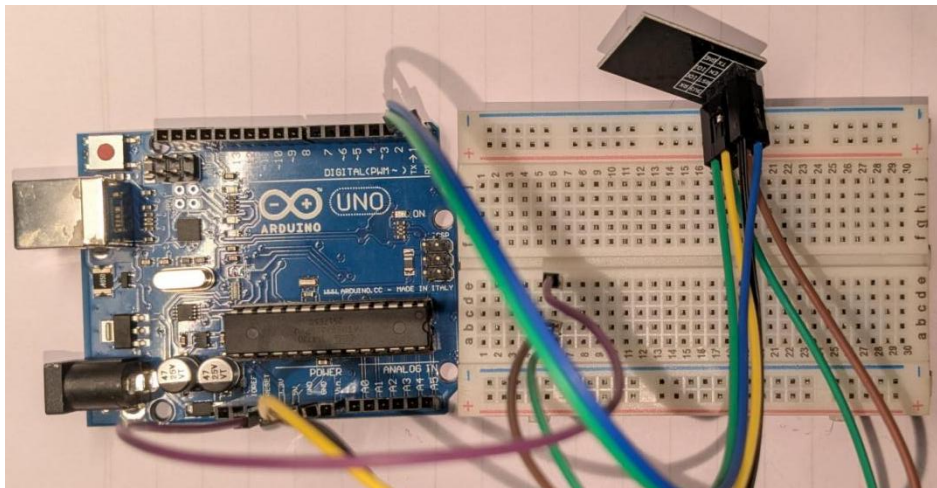
Ans:

11.

(b) Perform a Simulation-based Configuration to connect ESP-01 to a Wi-Fi network and record your observations.
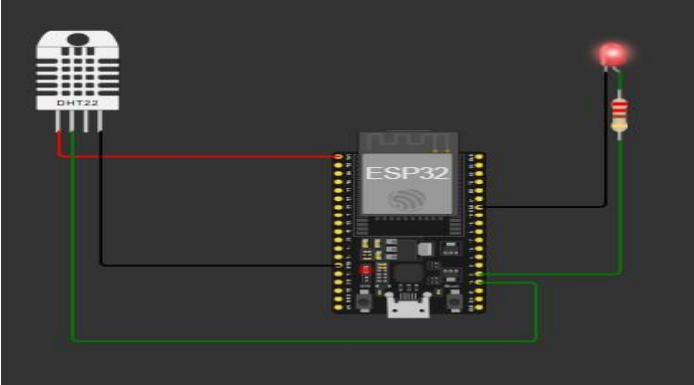
Ans:


```
> AT
OK
> AT+CWMODE=1
OK
> AT+CWJAP="SOA BH-12A","iter@bh12"
WIFI CONNECTED
WIFI GOT IP
OK
> AT+CIFSR
+CIFSR:STAIP,"192.168.1.xxx"
OK
```

(c) Perform Hardware-based Configuration to connect ESP-01 to a Wi-Fi network and capture the serial monitor output.
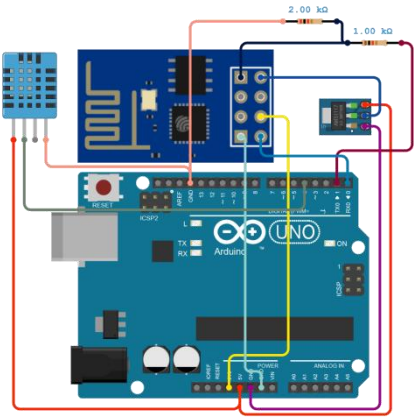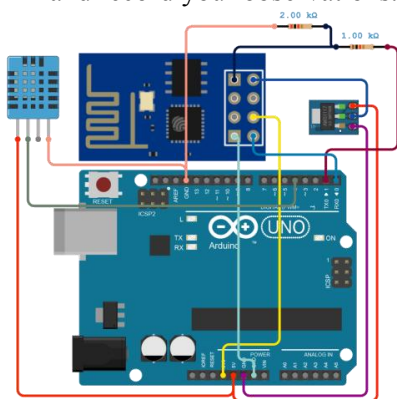
Ans:

CO3

L3 & L4

| | ```
> AT
OK
> AT+CWMODE=1
OK
> AT+CWJAP="SOA BH-12A","iter@bh12"
WIFI CONNECTED
WIFI GOT IP
OK
> AT+CIFSR
+CIFSR:STAIP,"192.168.1.xxx"
OK
``` | | |
| **12** | Using the ESP8266 ESP-01 Wi-Fi module and Arduino Uno, demonstrate how to wirelessly transmit sensor data from Arduino to a remote server or monitor through both simulation and hardware implementation.<br><br><br><br>(a) Draw the circuit diagram showing the connection between Arduino UNO, ESP-01 module, and a sensor for wireless data transmission.<br>Ans:<br><br><br><br>(b) Write a short Arduino code snippet to send temperature or humidity data from a DHT11/LM35 sensor through ESP-01 to a remote server.<br>Ans:<br><br>```
#include <SoftwareSerial.h>
#include <DHT.h>
#define DHTPIN 4
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
SoftwareSerial esp(2,3); // RX, TX
const char ssid[] = "YOUR_SSID";
const char pass[] = "YOUR_PASS";
const char host[] = "example.com"; // remote server (IP or domain)
const int  port = 80;
unsigned long lastSend = 0;
void setup(){
  Serial.begin(9600);
  esp.begin(9600);
  dht.begin();
  delay(1000);
``` | **CO4** | **L4 & L5** |

```
  sendAT("AT");
  sendAT("AT+CWMODE=1");
  connectWiFi();
}
void loop(){
 if (millis() - lastSend >= 15000){
  float t = dht.readTemperature();
  if (isnan(t)){ lastSend = millis(); return; }
  sendHTTP(t);
  lastSend = millis();
 }
 while(esp.available()){ Serial.write(esp.read()); }
 while(Serial.available()){ esp.write(Serial.read()); }
}
void sendAT(const char *cmd){
 esp.println(cmd);
 waitFor("OK",3000);
}
void connectWiFi(){
 String cmd = String("AT+CWJAP=\"") + ssid + "\",\"" + pass + "\"";
 esp.println(cmd);
 waitFor("WIFI GOT IP",15000);
}
void sendHTTP(float temp){
 String payload = String("GET /update?temp=") + String((int)temp) + "
HTTP/1.1\r\nHost: " + host + "\r\nConnection: close\r\n\r\n";
 String cipstart = String("AT+CIPSTART=\"TCP\",\"") + host + "\"," + port;
 esp.println(cipstart);
 if (!waitFor("CONNECT",5000)) return;
 String cipsend = String("AT+CIPSEND=") + payload.length();
 esp.println(cipsend);
 if (!waitFor(">",3000)) return;
 esp.print(payload);
 waitFor("CLOSED",5000);
 esp.println("AT+CIPCLOSE");
 waitFor("OK",2000);
}
bool waitFor(const char *pat, unsigned long timeout){
 unsigned long t0 = millis();
 String r;
 while(millis()-t0 < timeout){
  while(esp.available()) r += (char)esp.read();
  if (r.indexOf(pat) != -1) return true;
 }
 return false;
}
```
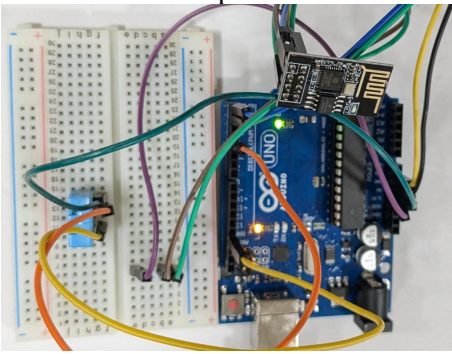
(c) Perform Simulation-based implementation to wirelessly transmit sensor data and record your observations.


```
> AT
OK
> AT+CWMODE=1
OK
> AT+CWJAP="MySSID","MyPass"
WIFI CONNECTED
WIFI GOT IP
> AT+CIPSTART="TCP","example.com",80
CONNECT
> AT+CIPSEND=58
> GET / HTTP/1.1\r\nHost: example.com\r\n\r\n
+IPD,123:HTTP/1.1 200 OK
+CLOSED
```
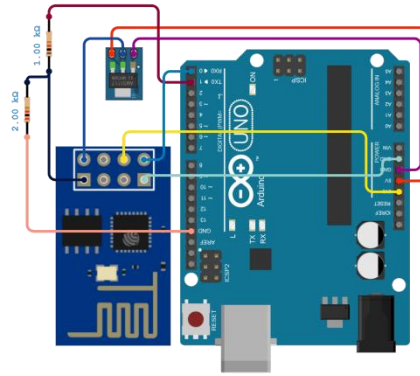
(d) Perform Hardware-based implementation to demonstrate actual data transmission and capture the serial monitor output or server response.


```
> AT
OK
> AT+CWMODE=1
OK
> AT+CWJAP="MySSID","MyPass"
WIFI CONNECTED
WIFI GOT IP
> AT+CIPSTART="TCP","example.com",80
CONNECT
> AT+CIPSEND=58
> GET / HTTP/1.1\r\nHost: example.com\r\n\r\n
+IPD,123:HTTP/1.1 200 OK
+CLOSED
```

Ans:

(e) Conclude your observations by discussing how reliable data transmission can be achieved using ESP-01 in IoT applications.

Ans:

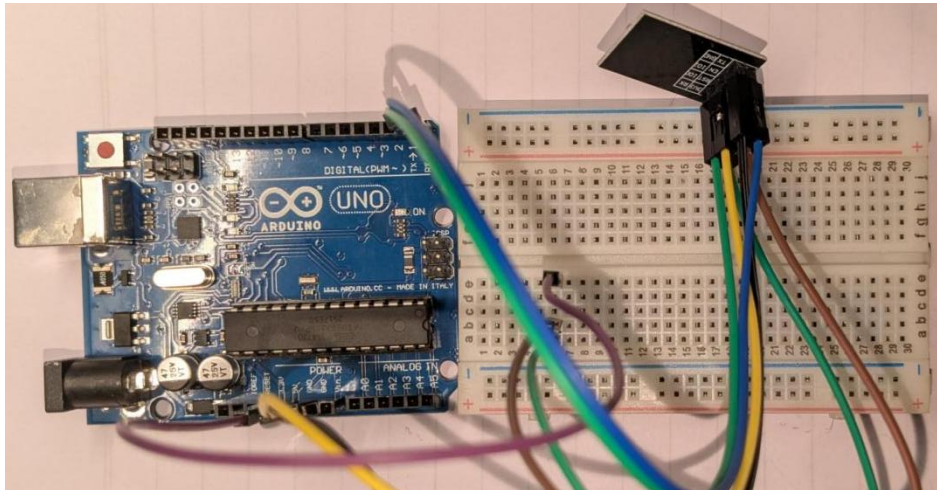| 13	Using the ESP8266 ESP-01 Wi-Fi module and Arduino UNO, demonstrate how to implement TCP/IP protocols for client-server communication through hardware-based development		
	(a) How can you configure the ESP-01 module as a TCP client to connect with a remote server? Ans:	**CO5**	**L4 & L5**
	(b) Write the AT commands sequence for establishing a TCP connection and transmitting data. Ans:		

(c) Draw the circuit diagram showing the setup for client-server communication between Arduino UNO and ESP-01.

Ans:

(d) Perform Hardware-based development to implement TCP/IP communication and record the serial monitor output.

Ans:


```
Starting ESP-01 TCP Client Sequence...

Sending: AT
OK

Sending: AT+RST
OK
ets Jan  8 2013,rst cause:2, boot mode:(3,6)

ready

Sending: AT+CWMODE=1
OK

Connecting to WiFi...
Sending: AT+CWJAP="MySSID","MyPassword"
WIFI CONNECTED
WIFI GOT IP
OK

Requesting IP address...
Sending: AT+CIFSR
+CIFSR:STAIP,"192.168.1.107"
+CIFSR:STAMAC,"84:f3:eb:2a:65:9c"
OK

Setting single connection mode...
Sending: AT+CIPMUX=0
OK

Opening TCP connection...
Sending: AT+CIPSTART="TCP","192.168.1.100",5000
CONNECT
OK

Sending data...
Sending: AT+CIPSEND=19
>
Hello from ESP-01
SEND OK

Remote server replied:
+IPD,16:Echo: Hello ESP

Closing connection...
Sending: AT+CIPCLOSE
CLOSED

TCP Client Operation Complete.
```

14	Explain how ESP-01 Wi-Fi communication contributes to the development of smart IoT systems with an example. Ans:	CO6	L5 & L6

Assignment 1	**Topic: WiFi-Communication**	**Date of Assignment1: 08.11.2025**	**Date of Submission: ---.11.2025**

Note:

1. Assignment carries a weightage of **20 marks out of 100**
2. Course outcome CO1 to CO2 was covered.

Course Outcomes	CO1	Understand the fundamentals of Arduino hardware and software.
	CO2	Interface various sensors and actuators with Arduino.
	CO3	Apply programming logic to control robotic systems.
	CO4	Design and build basic to intermediate level robotics projects.
	CO5	CO5 Demonstrate problem-solving and debugging skills in robotics.
	CO6	Collaborate in teams to plan, execute, and present robotics projects