Q.1 Explain and write a program for interfacing of ESP8266 WiFi Module with Ardunio.

→

```
#include "ESP8266_AT.h"
# define  SEND_DEMO
# define  DOMAIN
#define  PORT
# define  API_WRITE_KEY
# define  CHANNEL_ID
#define  SSID
# define  PASSWORD

Char buffer [150];
uin8_t  Connect_status;
#ifdef SEND_DEMO
uint8_t  Sample = 0;
#endif

Void setup() {
   Serial.begin (115200);
   while (!ESP8266_Begin());
   ESP8266_WIFIMode(BOTH STATION-AND ACCESPOINT);
   ESP 8266_ConnectionMode (SINGLE);
   ESP 8266_ApplicationMode (NORMAL));
   if (ESP8266_connected() == ESP8266_NOT_CONNECTED_
      TO_AP)
   ESP8266_JoinAccessPoint (SSID, PASSWORD);
   ESP 8266_Start(0, DOMAIN, PORT);
}
```

```
void loop(){
  Connect_Status = ESP8266_conneted();
  if (Connect_Status == ESP8266_NOT_CONNECTED_TO_AP)
  ESP8266_JoinAccessPoint (SSID, PASSWORD);
  if(Connect_Status == ESP8266_TRANSMISSION_DIS-
     CONNECTED)
  ESP8266_Start (0, DOMAIN, PORT);

  #ifdef SEND_DEMO
  memset (buffer, 0,150);
  sprintf(_buffer, "GET/update?Api-key =%s &field1=%d"
          API_WRITE_KEY, sample++)
  ESP8266_send (Buffer);
  delay (15000);
  #endif

 #ifdef RECEIVE_DEMO
  memset (_buffer, 0, 150);
  sprintf (_buffer, "GET/ channels, %s /feeds / last.txt",
          CHANNEL_ID);
  ESP8266_send (buffer);
  Read_Data ('buffer);
  delay (600);
 #endif
}
```

At client end, we need to check ESP8266 responses We can check it on the serial terminal of PC/ laptop. Connect ESP8266 module transmit

To recieve pin (Rx) of Ardunio UNO and to recieve pin (Rx) of USB to serial converter as Connect USB to serial converter to PC/laptop. Open the serial terminal on PC/Laptop to see the ESP8266 responses for the AT command sent from Ardunio UNO.

Q.2 → Explain and write program for interfacing of Relay and DHT11 with Raspberry Pi.

```
import time
import Adafruit_charLCD as LCD
import Adafruit_DHT
sensor_name = Adafruit_DHT.DHT11
sensor_pin = 17

lcd_rs =7
lcd_en =8
lcd_du =25
lcd_d5 =24
lcd_d6 =23
lcd_d7 =18
lcd_backlight=0
lcd_columns=16
lcd_rows=2

lcd = LCD.Adafruit_charLCD(lcd_rs, lcd_en, lcd_d4,
      lcd_64, lcd 65, Lcd_d6, lcd_d7, lcd_columns
lcd_rows_ lcd_backlight)
lcd.message ('DHT11 with Pi \n - CircuitDigest')
time.sleep(2)
while 1:
    humidity, temperature = Adufruit_DHT.read_retry
     (sensor_name, sensor_pin)
    lcd_clear()
    lcd.message ('Temp= %1.f C' % temperature)
    lcd.message ('\n Hum =%1f %%' % humidity)
    time.sleep(2)
```

- We have to import the LCD library and DHT11 library into our program to use the functions related to it.
- Next we have to specify to which pins the sensor is connected to and what type of temperature sensor is used.
- The variable sensor_name is assigned to Adafruit.DHT.DHT11 since we are using the DH711 sensor here.
- The output pin of the sensor is connected to GPIO 17 of Raspberry Pi
- Similar we also have to define which GPIO pins the LCD is connected to
- Now we have to declared the LCD pins and the number of Rows and columns of LCD.
- Finally inside our while loop we should read the value of temperature and humidity for sensor.

Q.3 Explain features and Application of Contiki OS.
→ Contiki is an operating system for networked, memory-constrained systems with a focus on low-power wireless internet of Things (IOT devices)

Features of Contiki OS
① Multitasking kernel
② Optional per-application preemptive multithreading
③ Protothreads.
④ Internet Protocol Suite (TCP/IP) networking, include IPv6
⑤ Personal web serverdam Dunkels
⑥ Simple telnet client
⑦ It is supported by popular SSL / TLS libraries.

Application of Contiki OS
① Street lighting
② Sound monitoring for smart cities
③ radiation monitoring and alarms.