IOT- FIRMWARE DEVELOPMENT

R MADESHWAR
(CEG-Anna university)

3.Write an Esp32-based code using an esp32 microcontroller that has the following tasks.
a. To perform OTA through Wi-Fi/LTE by using HTTP/MQTT/FTP protocol for ESP32 microcontroller.
b. Transfer the firmware via UART protocol to update the firmware of the Arduino microcontroller.

```
#include <WiFi.h>
#include <ESPmDNS.h>
#include <WiFiUdp.h>
#include <ArduinoOTA.h>

const char* ssid = "your-SSID";
const char* password = "your-PASSWORD";

void setupWiFi() {
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi...");
  }
  Serial.println("Connected to WiFi");
}

void setupOTA() {
  ArduinoOTA.onStart([]() {
    Serial.println("OTA Update Started");
  });

  ArduinoOTA.onEnd([]() {
    Serial.println("\nOTA Update Finished");
  });

  ArduinoOTA.onProgress([](unsigned int progress, unsigned int total) {
    Serial.printf("Progress: %u%%\r", (progress / (total / 100)));
  });

  ArduinoOTA.onError([](ota_error_t error) {
    Serial.printf("Error[%u]: ", error);
    if (error == OTA_AUTH_ERROR) Serial.println("Auth Failed");
    else if (error == OTA_BEGIN_ERROR) Serial.println("Begin Failed");
    else if (error == OTA_CONNECT_ERROR) Serial.println("Connect
Failed");
    else if (error == OTA_RECEIVE_ERROR) Serial.println("Receive
Failed");
    else if (error == OTA_END_ERROR) Serial.println("End Failed");
  });

  ArduinoOTA.begin();
}

void setup() {
  Serial.begin(115200);
  setupWiFi();
  setupOTA();
}

void loop() {
  ArduinoOTA.handle();
}
```

Firmware transfer via UART for Arduino:

```
#include <SoftwareSerial.h>

SoftwareSerial mySerial(2, 3); // RX, TX

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

void loop() {
  if (mySerial.available()) {
    // Read data from UART and update firmware
  }
}
```