```
#include <WiFi.h>
#include <OneWire.h>
#include <DallasTemperature.h>   //温度
#include <DNSServer.h>
#include "ESPAsyncWebServer.h"

DNSServer dnsserver;
AsyncWebServer server(80);

#define ONE_WIRE_BUS (5) // DQ 接 ESP8266 引脚 D4(GPIO2)

//设置一个 oneWire 实例来与任何 oneWire 设备通信(不仅仅是 Maxim/Dallas 温度 ic)
OneWire oneWire(ONE_WIRE_BUS);

// 通过 oneWire 实例化对象 Dallas Temperature.
DallasTemperature sensors(&oneWire);


// 发现的温度设备数量
int numberOfDevices;

// We'll use this variable to store a found device address
DeviceAddress tempDeviceAddress;

//连接 WIFI
void connect_wifi(){
  const byte DNS_PORT = 53;                          //DNS 端口
  const String url = "luo.com";                     //域名
  IPAddress APIp(1,2,3,4);                           //AP IP
  IPAddress APGateway(1,2,3,4);                      //AP 网关
  IPAddress APSubnetMask(255,255,255,0);             //AP 子网掩码
  const char* APSsid = "yang";                      //AP SSID
  const char* APPassword = "";                 //AP wifi 密码

  const char* wifi_ssid = "P20";       //不要与上面的混淆，这是真实的 WIFI 账号
  const char* wifi_password = "12345678";     //不要与上面的混淆，这是真实的 WIFI 密码
  Serial.begin(9600);
  WiFi.mode(WIFI_AP_STA);                                    //打开 AP 和 STA 共存模式
  WiFi.softAPConfig(APIp, APGateway, APSubnetMask);     //设置 AP 的 IP 地址，网关和子网掩
码
  WiFi.softAP(APSsid, APPassword, 6);                        //设置 AP 模式的登陆名和密码
  dnsserver.start(DNS_PORT, url, APIp);                     //设置 DNS 的端口、网址、和 IP
  WiFi.begin(wifi_ssid, wifi_password);                     //连接 WIFI
  Serial.print("Connected");
```

```
  //循环，直到连接成功
  while(WiFi.status() != WL_CONNECTED){
    Serial.print(".");
    delay(500);
  }
  Serial.println();
  IPAddress local_IP = WiFi.localIP();
  Serial.print("WIFI is connected,The local IP address is: "); //连接成功提示
  Serial.println(local_IP);
  Serial.println(APIp);

}

void tongji(){

  // 获取单总线上的设备数量
  numberOfDevices = sensors.getDeviceCount();
  Serial.print("温度计的个数： ");
  Serial.println(numberOfDevices);

  // 定位总线上的设备并打印出每个设备被的地址
  Serial.print("Locating devices...");
  Serial.print("Found ");
  Serial.print(numberOfDevices, DEC);
  Serial.println(" devices.");

  // Loop through each device, print out address
  for(int i=0;i<numberOfDevices; i++){
      // 搜索总线上的设备地址并打印出来
      if(sensors.getAddress(tempDeviceAddress, i)){
          Serial.print("Found device ");
          Serial.print(i, DEC);
          Serial.print(" with address: ");
          //Serial.print(tempDeviceAddress);
          printAddress(tempDeviceAddress);
          Serial.println();
      }else{
            Serial.print("Found ghost device at ");
            Serial.print(i, DEC);
            Serial.print(" but could not detect address. Check power and cabling");
             }
           }

  }
```

```
// 一个储存网页的数组
const char index_html[] PROGMEM = R"rawliteral(
<!DOCTYPE HTML>
<html>
<head>
  <meta charset="utf-8">
  <style type="text/css">
html{
font-family: 'Open Sans', sans-serif;
display: block;
margin: 0px auto;
text-align: center;
color: #333333;
}
body{
margin-top: 50px;
}
h1{
margin: 50px auto 30px;
}
.wd{
margin: 50px auto 30px;
width: auto;
color: #f39c12
}
.wd1{
margin: 50px auto 30px;
width: auto;
color: #3498db
}
.side-by-side{
display: inline-block;
vertical-align: middle;
position: relative;
}
.temperature-text{
font-weight: 600;
padding-left: 15px;
font-size: 19px;
width: 160px;
text-align: left;
}
.temperature{
font-weight: 300;
```

```
font-size: 60px;
color: #f39c12;
}
.superscript{
font-size: 17px;
font-weight: 600;
position: absolute;
right: -20px;
top: 15px;
}
.data{
padding: 10px;
}
</style>
</head>
<body>
  <h1>不同材料导热性实验</h1>
  <!-- 创建一个 ID 位 dht 的盒子用于显示获取到的数据 -->
  <div id="dht">
  </div>
  <!--<button onclick="set()"> 发送数据 </button>-->
</body>
<script>
  // 按下按钮会运行这个 JS 函数
  function set() {
    var payload = "ESP32"; // 需要发送的内容
    // 通过 get 请求给 /set
    var xhr = new XMLHttpRequest();
    xhr.open("GET", "/set?value=" + payload, true);
    xhr.send();
  }
  // 设置一个定时任务, 1000ms 执行一次
  setInterval(function () {
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function () {
      if (this.readyState == 4 && this.status == 200) {
        // 此代码会搜索 ID 为 dht 的组件，然后使用返回内容替换组件内容
        document.getElementById("dht").innerHTML = this.responseText;
      }
    };
    // 使用 GET 的方式请求 /dht
    xhttp.open("GET", "/dht", true);
    xhttp.send();
  }, 1000)
```

```
</script>)rawliteral";

String Merge_Data(void)
{
   /*
   numberOfDevices = sensors.getDeviceCount();
   String dataBuffer = "<p>";
   for(int i=0;i<numberOfDevices; i++){
      int tempC = 0;// 此处应为获取温度代码
      // Search the wire for address
      if(sensors.getAddress(tempDeviceAddress, i)){
         // Output the device ID
         //Serial.print("Temperature for device: ");
         //Serial.println(i,DEC);
         // Print the data
         dataBuffer += "<b>第";
         i=i+1;
         dataBuffer +=i;
         dataBuffer +="个温度: ";
         float tempC = sensors.getTempC(tempDeviceAddress);
         dataBuffer += float(tempC);
         dataBuffer += "</b><br/>";
      }
   }
   dataBuffer += "</p>";
   return dataBuffer;
   */
   sensors.requestTemperatures(); // 发送命令获取温度
   numberOfDevices = sensors.getDeviceCount();
   //numberOfDevices 应该是 3
   String dataBuffer = "<p>";
   for(int i=0;i<numberOfDevices; i++){
         dataBuffer += "<b>第";
         int a=i+1;
         dataBuffer +=a;
         dataBuffer +="个温度: ";
         sensors.getAddress(tempDeviceAddress, i);    //这一句是必须的
         // Print the data
         float tempC = sensors.getTempC(tempDeviceAddress);
         dataBuffer += float(tempC);
         dataBuffer += "</b><br/>";
   }
   dataBuffer += "</p>";
   return dataBuffer;
```

```
  /*
  //*****************************

  int Humidity = 50;// 此处应为获取温度代码
  int Temperature = 26;// 此处应为获取湿度代码
  // 将温湿度打包为一个 HTML 显示代码
  String dataBuffer = "<p>";
  dataBuffer += "<h1>传感器数据 </h1>";
  dataBuffer += "<b>温度: </b>";
  dataBuffer += String(Temperature);
  dataBuffer += "<br/>";
  dataBuffer += "<b>湿度: </b>";
  dataBuffer += String(Humidity );
  dataBuffer += "<br /></p>";
  // 最后要将数组返回出去
  return dataBuffer;
*/
}

// 下发处理回调函数
void Config_Callback(AsyncWebServerRequest *request)
{
  if (request->hasParam("value")) // 如果有值下发
  {
    String HTTP_Payload = request->getParam("value")->value();      // 获取下发的数据
    Serial.printf("[%lu]%s\r\n", millis(), HTTP_Payload.c_str()); // 打印调试信息
  }
  request->send(200, "text/plain", "OK"); // 发送接收成功标志符
}

void setup()
{
  connect_wifi();
  sensors.begin();//如果这里不初始化，就不会成功。
  tongji();
 /*
 // 获取单总线上的设备数量
  numberOfDevices = sensors.getDeviceCount();
  //Serial.println(numberOfDevices);
  Serial.print("温度计的个数： ");
  Serial.println(numberOfDevices);

  // 定位总线上的设备并打印出每个设备被的地址
  Serial.print("Locating devices...");
```

```
  Serial.print("Found ");
  Serial.print(numberOfDevices, DEC);
  Serial.println(" devices.");

  // Loop through each device, print out address
  for(int i=0;i<numberOfDevices; i++){
      // 搜索总线上的设备地址并打印出来
      if(sensors.getAddress(tempDeviceAddress, i)){
          Serial.print("Found device ");
          Serial.print(i, DEC);
          Serial.print(" with address: ");
          //Serial.print(tempDeviceAddress);
          printAddress(tempDeviceAddress);
          Serial.println();
      }else{
            Serial.print("Found ghost device at ");
            Serial.print(i, DEC);
            Serial.print(" but could not detect address. Check power and cabling");
             }
           }
  */
  // 你需要再此处添加 WiFi 操作代码，开启热点或者连接到热点
  // 添加 HTTP 主页，当访问的时候会把网页推送给访问者
  server.on("/", HTTP_GET, [](AsyncWebServerRequest *request)
            { request->send_P(200, "text/html", index_html); });
  // 设置反馈的信息，在 HTML 请求这个 Ip/dht 这个链接时，返回打包好的传感器数据
  server.on("/dht", HTTP_GET, [](AsyncWebServerRequest *request)
            { request->send_P(200, "text/plain", Merge_Data().c_str()); });
  server.on("/set", HTTP_GET, Config_Callback);    // 绑定配置下发的处理函数
  server.begin();   // 初始化 HTTP 服务器
  //Serial.print(Merge_Data());
}

void loop() {
  dnsserver.processNextRequest();
}

// 打印设备地址
void printAddress(DeviceAddress deviceAddress) {
  for (uint8_t i = 0; i < 8; i++){
    if (deviceAddress[i] < 16) Serial.print("0");
      Serial.print(deviceAddress[i], HEX);
  }
}
```