**Project 36：WiFi Smart Home**

1. **项目介绍：**

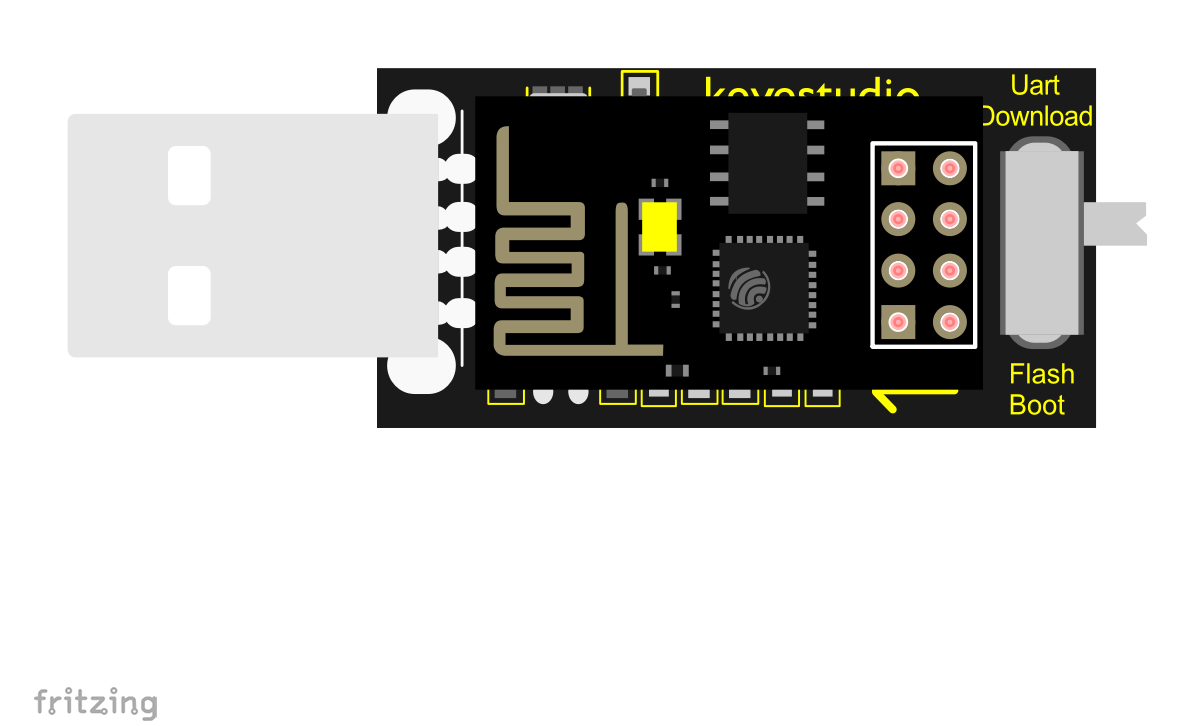
在前面的Project 35中，我们已经知道ESP8266串口WIFI ESP-01模块通过WiFi测试代码得到相关的WiFi信息。那么在这个实验中，我们将使用ESP8266串口WiFi ESP-01模块通过APP和WiFi来控制多个传感器/模块工作，实现WiFi智能家居的效果。

1. **项目元件：**

|  |  |  |
| --- | --- | --- |
| 17a6d7f241a04d4e932cb06b758197c4 | _DSC2552 | KS6033-2 |
| Raspberry Pi Pico\*1 | Raspberry Pi Pico扩展板\*1 | 温湿度传感器\*1 |
|  | KS0388 Keyestudio USB转ESP-01S WIFI模块串口测试扩展板_0002_图层 1 | KS6038 130电机驱动模块 |
| ESP8266串口WIFI ESP-01\*1 | USB转ESP-01S WIFI模块串口测试扩展板\*1 | 130直流电机模块\*1 |
| 8 | 12 | 杜邦线-3 |
| 5V继电器模块\*1 | 智能手机/平板电脑\*1 | 公对母杜邦线若干 |
|  | hc-04 |  |
| 舵机\*1 | 超声波传感器\*1 | USB 线\*1 |
|  | 杜邦线-1 |  |
| 面包板\*1 | 母对母杜邦线若干 | 跳线若干 |

**3. 将WIFI模块串口测试扩展板插入电脑的USB口：**

将ESP8266串口WIFI ESP-01模块正确方向插入USB转ESP-01S WIFI模块串口测试扩展板上。



先将USB转ESP-01S WIFI模块串口测试扩展板上的拨码开关拨到UartDownload端，再将USB转ESP-01S WIFI模块串口测试扩展板插入电脑的USB口。

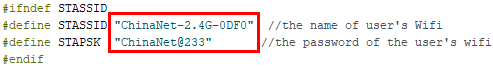
****

**3.ESP8266 代码：**

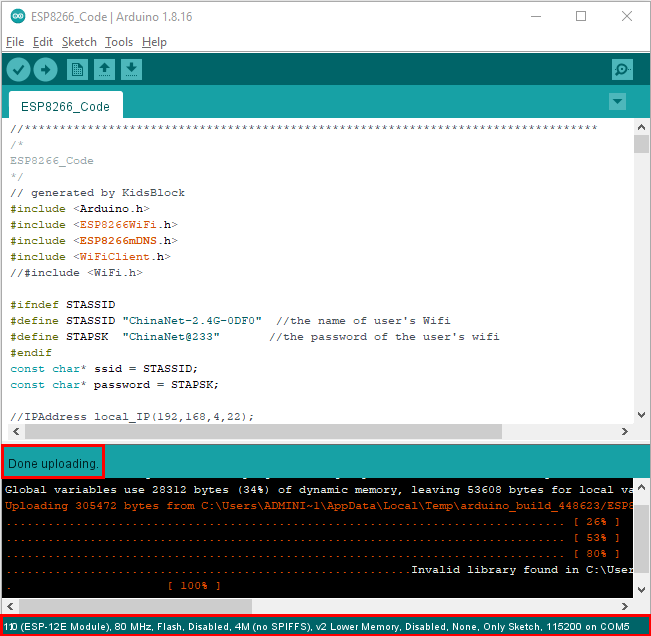
注意：打开Arduino IDE后，一定要先设置好ESP8266板型和COM口。手机和设备需要连接在同一个WiFi上，如果家里没有WiFi需要打开手机热点共享WiFi，打开手机热点共享WiFi是最好的方法。

|  |
| --- |
| //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  /\*  ESP8266\_Code  \*/  // generated by KidsBlock  #include <Arduino.h>  #include <ESP8266WiFi.h>  #include <ESP8266mDNS.h>  #include <WiFiClient.h>  //#include <WiFi.h>  #ifndef STASSID  #define STASSID "ChinaNet-2.4G-0DF0" //the name of user's Wifi  #define STAPSK "ChinaNet@233" //the password of the user's wifi  #endif  const char\* ssid = STASSID;  const char\* password = STAPSK;  //IPAddress local\_IP(192,168,4,22);  //IPAddress gateway(192,168,4,22);  //IPAddress subnet(255,255,255,0);  //  //const char \*ssid = "ESP8266\_AP\_TEST";  //const char \*password = "12345678";  WiFiServer server(80);  String unoData = "";  int ip\_flag = 0;  int ultra\_state = 1;  String ip\_str;  void setup() {  Serial.begin(9600);  // WiFi.mode(WIFI\_AP); //Set to work in AP mode  //  // WiFi.softAPConfig(local\_IP, gateway, subnet); //Setting an AP Address  // while(!WiFi.softAP(ssid, password)){}; //Start AP  // Serial.println("AP [starting](javascript:;) [success](javascript:;)");  //  // Serial.print("IP address: ");  // Serial.println(WiFi.softAPIP()); // Printing the IP Address  //  // WiFi.softAPsetHostname("myHostName"); //Set host name  // Serial.print("HostName: ");  // Serial.println(WiFi.softAPgetHostname()); //print host name  //  // Serial.print("mac Address: ");  // Serial.println(WiFi.softAPmacAddress()); //prnt mac add  WiFi.mode(WIFI\_STA);  WiFi.begin(ssid, password);  while (WiFi.status() != WL\_CONNECTED) {  delay(500);  Serial.print(".");  }  Serial.print("IP ADDRESS: ");  Serial.println(WiFi.localIP());  if (!MDNS.begin("esp8266")) {  //Serial.println("Error setting up MDNS responder!");  while (1) {  delay(1000);  }  }  // Serial.println("mDNS responder started");  server.begin();  //Serial.println("TCP server started");  MDNS.addService("http", "tcp", 80);  ip\_flag = 1;  }  void loop() {  //Serial.println(WiFi.softAPgetStationNum()); //Prints the number of client connections  if(ip\_flag == 1)  {  for(int i=3; i>0; i--)  {  Serial.print("IP: ");  Serial.print(WiFi.localIP());  Serial.println('#');  delay(500);  }  ip\_flag = 0;    }  MDNS.update();  WiFiClient client = server.available();  if (!client) {  return;  }  //Serial.println("");  while (client.connected() && !client.available()) {  delay(1);  }  String req = client.readStringUntil('\r');  int addr\_start = req.indexOf(' ');  int addr\_end = req.indexOf(' ', addr\_start + 1);  if (addr\_start == -1 || addr\_end == -1) {  //Serial.print("Invalid request: ");  //Serial.println(req);  return;  }  req = req.substring(addr\_start + 1, addr\_end);  int len\_val = String(req).length();  String M\_req = String(req).substring(0,6);  //Serial.println(M\_req);  if(M\_req == "/")  {  String s\_M\_req = String(req).substring(5,len\_val);  Serial.print(s\_M\_req);  Serial.print("#");  }  if(M\_req == "/btn/v")  {  String s\_M\_req = String(req).substring(5,len\_val);  Serial.print(s\_M\_req);  Serial.print("#");  }  client.flush();  String s;  if (req == "/") {  IPAddress ip = WiFi.localIP();  String ipStr = String(ip[0]) + '.' + String(ip[1]) + '.' + String(ip[2]) + '.' + String(ip[3]);  s = "HTTP/1.1 200 OK\r\nContent-Type: text/html\r\n\r\n<!DOCTYPE HTML>\r\n<html>Hello from ESP8266 at ";  s += ipStr;  s += "</html>\r\n\r\n";  //Serial.println("Sending 200");  Serial.println(WiFi.localIP());  Serial.write('\*');  client.println(WiFi.localIP());  ip\_flag = 0;  }  else if(req == "/btn/0")  {  Serial.write('a');  client.println("turn on the relay");  }  else if(req == "/btn/1")  {  Serial.write('b');  client.println("turn off the relay");  }  else if(req == "/btn/2")  {  Serial.write('c');  client.println("Bring the steering gear over 180 degrees");  }  else if(req == "/btn/3")  {  Serial.write('d');  client.println("Bring the steering gear over 0 degrees");  }  else if(req == "/btn/4")  {  Serial.write('e');  client.println("esp8266 already turn on the fans");  }  else if(req == "/btn/5")  {  Serial.write('f');  client.println("esp8266 already turn off the fans");  }  else if(req == "/btn/6")  {  Serial.write('g');  while(Serial.available() > 0)  {  unoData = Serial.readStringUntil('#');  client.println(unoData);  }  }  else if(req == "/btn/7")  {  Serial.write('h');  client.println("turn off the ultrasonic");  }  else if(req == "/btn/8")  {  Serial.write('i');  while(Serial.available() > 0)  {  unoData = Serial.readStringUntil('#');  client.println(unoData);  //client.flush();  }  }  else if(req == "/btn/9")  {  Serial.write('j');  client.println("turn off the temperature");  }  else if(req == "/btn/10")  {  Serial.write('k');  while(Serial.available() > 0)  {  unoData = Serial.readStringUntil('#');  client.println(unoData);  //client.flush();  }  }  else if(req == "/btn/11")  {  Serial.write('l');  client.println("turn off the humidity");  }  else if(req == "/btn/12")  {  Serial.write('m');  client.println(F("m"));  }  else if(req == "/btn/13")  {  Serial.write('n');  client.println(F("n"));  }  else if(req == "/btn/14")  {  Serial.write('o');  client.println(F("o"));  }  else if(req == "/btn/15")  {  Serial.write('p');  client.println(F("p"));  }  else if(req == "/btn/16")  {  Serial.write('q');  client.println(F("q"));  }  else if(req == "/btn/17")  {  Serial.write('r');  client.println(F("r"));  }  else if(req == "/btn/18")  {  Serial.write('s');  client.println(F("s"));  }  else if(req == "/btn/19")  {  Serial.write('t');  client.println(F("t"));  }  else if(req == "/btn/20")  {  Serial.write('u');  client.println(F("u"));  }  else if(req == "/btn/21")  {  Serial.write('v');  client.println(F("v"));  }  else if(req == "/btn/22")  {  Serial.write('w');  client.println(F("w"));  }  else if(req == "/btn/23")  {  Serial.write('x');  client.println(F("x"));  }  else {  //s = "HTTP/1.1 404 Not Found\r\n\r\n";  //Serial.println("Sending 404");  }  client.print(F("IP : "));  client.println(WiFi.localIP());  }  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |

特别注意：需要先将项目代码

中的用户Wifi名称和用户Wifi密码改成你们自己的Wifi名称和Wifi密码。

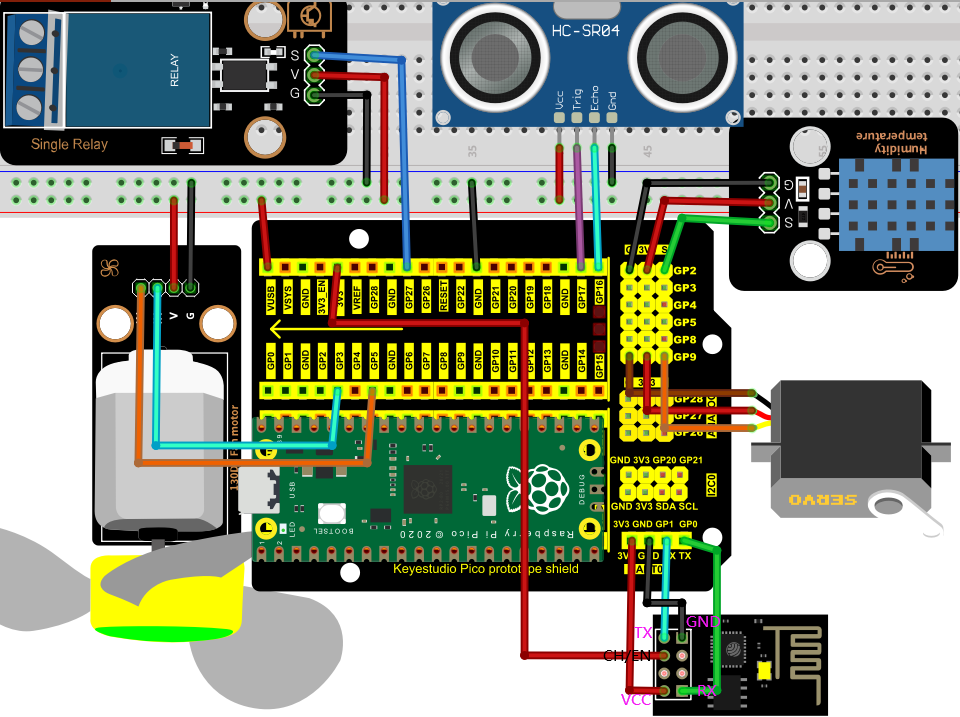
Wifi名称和Wifi密码修改后，确保USB转ESP-01S WiFi模块串口测试扩展板上的拨码开关已经拨到Uart Download 端，并且也确定USB转ESP-01S WIFI模块串口测试扩展板已经插入电脑的USB口。然后按照Project 35中的方法设置ESP8266板型和COM口，IDE右下角会显示对应的ESP8266板型和COM口，再点击将ESP8266 代码上传到ESP8266串口WIFI ESP-01模块上，上传成功。（注意：如果上传失败，在板型和COM口没问题情况下，将USB转ESP-01S WIFI模块串口测试扩展板从电脑的USB口拔下来再次插到电脑的USB口）



ESP8266 代码上传成功后，先将USB转ESP-01S WiFi模块串口测试扩展板从电脑的USB口拔下来，再将ESP8266串口WIFI ESP-01模块从USB转ESP-01S WiFi模块串口测试扩展板上拔下来。

1. **项目接线：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **继电器** | Raspberry Pi Pico扩展板 |  | 温湿度传感器 | Raspberry Pi Pico扩展板 |
| G | G | G | G |
| V | 5V | V | 3V3 |
| S | GP27 | S | GP2(S) |
|  | | | | |
| 超声波传感器 | Raspberry Pi Pico扩展板 |  | 130 风扇模块 | Raspberry Pi Pico扩展板 |
| Vcc | 5V | G | G |
| Trig | GP17 | V | 5V |
| Echo | GP16 | IN+ | GP3 |
| Gnd | G | IN- | GP5 |
|  | | | | |
| Wifi 模块 | Raspberry Pi Pico扩展板 |  | 舵机 | Raspberry Pi Pico扩展板 |
| 3V3 | 3V3 | Red line | 3V3 |
| EN/CP | 3V3 | Brown line | G |
| TX | RX(GP1) | Orange line | GP9(S) |
| RX | TX(GP0) |  |  |
| GND | GND |  |  |  |



**6.WiFi Smart Home项目代码：**

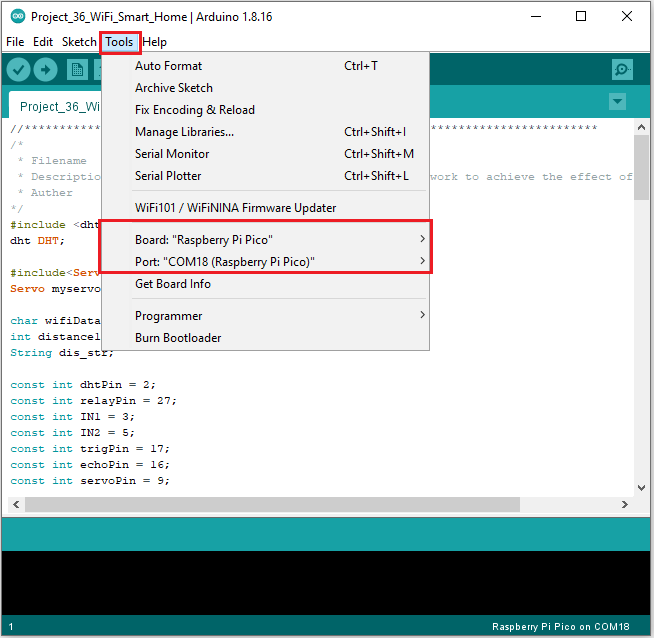
注意：打开Arduino IDE后，一定要先设置好Raspberry Pi Pico板型和COM口。如果家里没有WiFi需要打开手机热点共享WiFi.

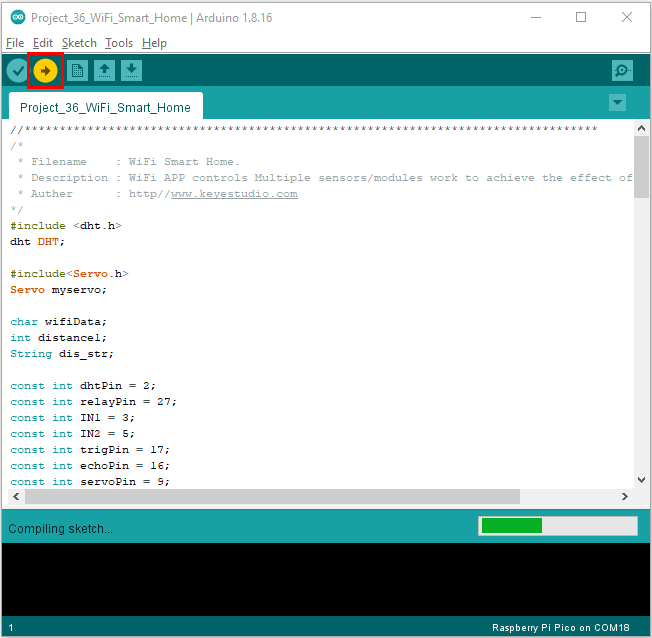
|  |
| --- |
| //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  /\*  \* Filename : WiFi Smart Home.  \* Description : WiFi APP controls Multiple sensors/modules work to achieve the effect of WiFi smart home.  \* Auther : http//www.keyestudio.com  \*/  #include <dht.h>  dht DHT;  #include<Servo.h>  Servo myservo;  char wifiData;  int distance1;  String dis\_str;  const int dhtPin = 2;  const int relayPin = 27;  const int IN1 = 3;  const int IN2 = 5;  const int trigPin = 17;  const int echoPin = 16;  const int servoPin = 9;  int ip\_flag = 1;  int ultra\_state = 1;  int temp\_state = 1;  int humidity\_state = 1;  void setup() {  Serial1.begin(9600);  pinMode(dhtPin, INPUT);  pinMode(relayPin, OUTPUT);  pinMode(servoPin, OUTPUT);  pinMode(IN1, OUTPUT);  pinMode(IN2, OUTPUT);  pinMode(trigPin, OUTPUT);  pinMode(echoPin, INPUT);  //turn off the fan  digitalWrite(IN1, LOW);  digitalWrite(IN2, LOW);  digitalWrite(relayPin, LOW); //turn off the relay module  myservo.attach(9);  //dht.begin();  }  void loop() {  int chk = DHT.read11(dhtPin);  if(Serial1.available() > 0)  {  wifiData = Serial1.read();  Serial.print(wifiData);  if(wifiData == '#')  {  ip\_flag = 0;  }    if(ip\_flag == 1)  {  //String ip\_addr = Serial.readStringUntil('#');  Serial.print(wifiData);  if(wifiData == '#')  {  Serial.println("");  }  delay(100);  }  }  switch(wifiData)  {  case 'a': digitalWrite(relayPin, HIGH); break;  case 'b': digitalWrite(relayPin, LOW); break;  case 'c': myservo.write(180); delay(200); break;  case 'd': myservo.write(0); delay(200); break;  case 'e': digitalWrite(IN1, HIGH); digitalWrite(IN2, LOW); break;  case 'f': digitalWrite(IN1, LOW); digitalWrite(IN2, LOW); break;  case 'g': while(ultra\_state>0)  {  Serial.print("Distance = ");  Serial.print(checkdistance());  Serial.println("#");  Serial1.print("Distance = ");  Serial1.print(checkdistance());  Serial1.println("#");  ultra\_state = 0;  }  break;  case 'h': ultra\_state = 1; break;  case 'i': while(temp\_state>0)  {  Serial.print("Temperature = ");  Serial.print(DHT.temperature,1);  Serial.println("#");  Serial1.print("Temperature = ");  Serial1.print(DHT.temperature,1);  Serial1.println("#");  temp\_state = 0;  }  break;  case 'j': temp\_state = 1; break;  case 'k': while(humidity\_state > 0)  {  Serial.print("Humidity = ");  Serial.print(DHT.humidity,1);  Serial.println("#");  Serial1.print("Humidity = ");  Serial1.print(DHT.humidity,1);  Serial1.println("#");  humidity\_state = 0;  }  break;  case 'l': humidity\_state = 1; break;  }    }  int checkdistance() {  digitalWrite(17, LOW);  delayMicroseconds(2);  digitalWrite(17, HIGH);  delayMicroseconds(10);  digitalWrite(17, LOW);  int distance = pulseIn(16, HIGH) / 58;    delay(10);  return distance;  }  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |

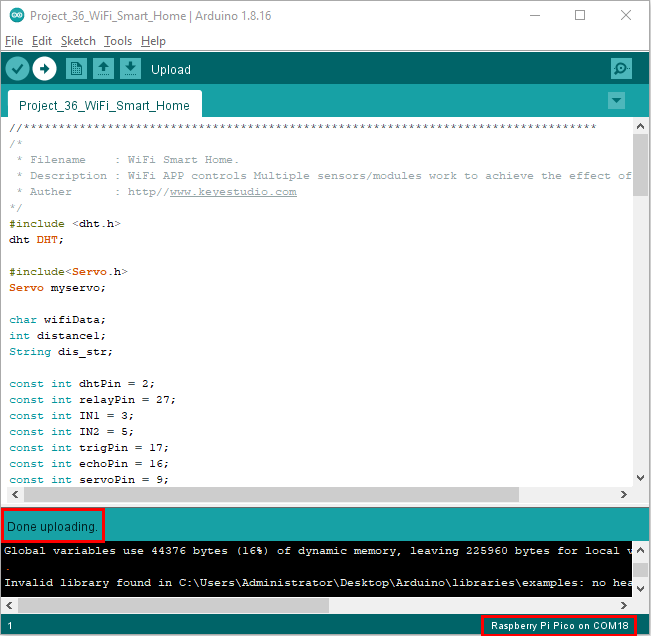
**6. 实现现象：**

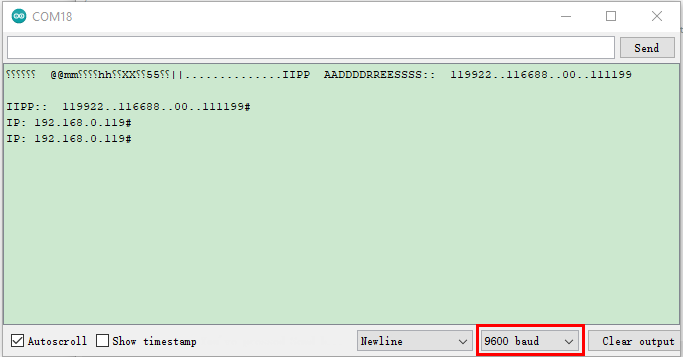
特别注意：上传项目代码前，需要先将连接到Raspberry Pi Pico扩展板上的TX和RX的杜邦线先拔下来，要不然代码上传可能不成功。

然后点击Arduino IDE菜单栏的"Tools" → "Board："，选择“Raspberry Pi Pico”，选择正确的COM端口，最后将WiFi Smart Home项目代码上传至Raspberry Pi Pico主板。上传代码成功后，再将接在ESP8266串口WIFI ESP-01模块上的TX杜邦线另一端接到Raspberry Pi Pico扩展板上的RX(GP1)引脚，RX的杜邦线另一端接到Raspberry Pi Pico扩展板上的TX(GP0)引脚。点击打开串口监视器窗口，将波特率设置为9600。这样，串口监视器就显示此时你们WiFi的IP地址。（WiFi的IP地址有时候会改变，如果原来的IP地址不行，需要重新检测WiFi的IP地址）



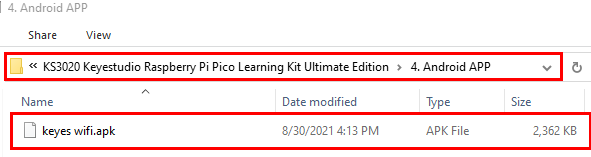




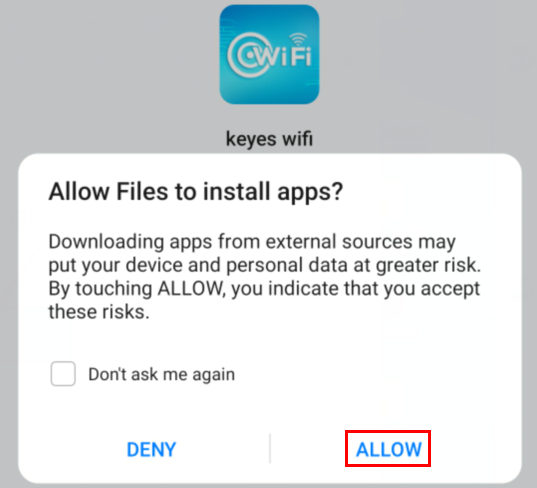


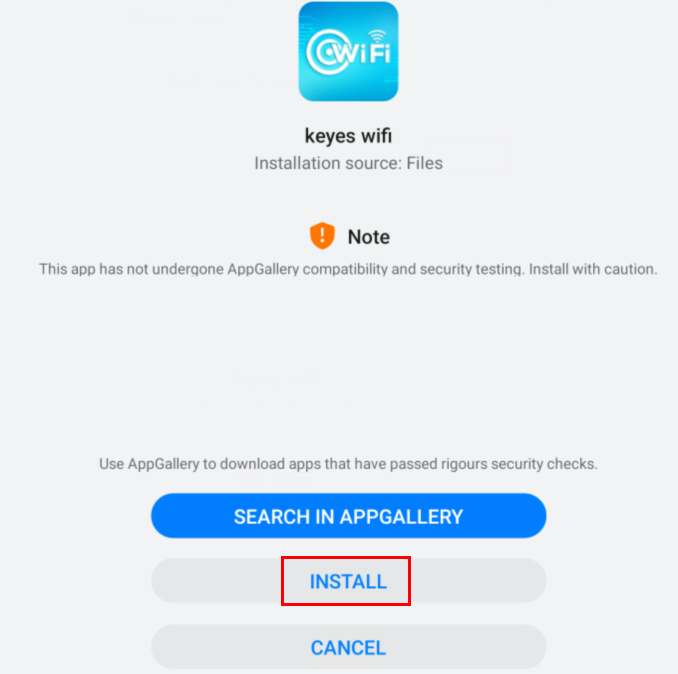
**安卓系统设备（手机/iPad）APP：**

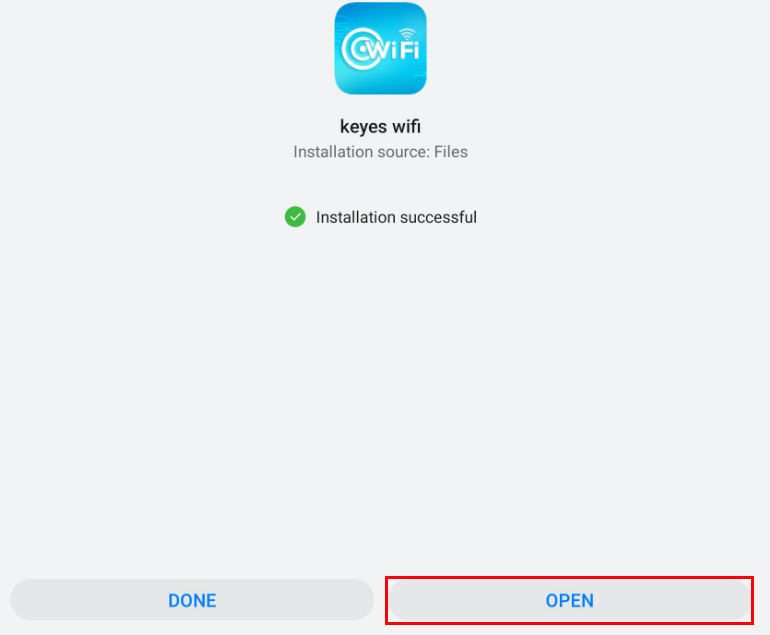
现将文件夹中的keyes wifi.apk文件转移到安卓系统手机或平板电脑上，点击keyes wifi.apk文件进入安装页面，点击“ALLOW”按钮，然后点击“INSTALL”按钮，过一会儿，安装完成后点击“OPEN”按钮就可以进入APP界面。

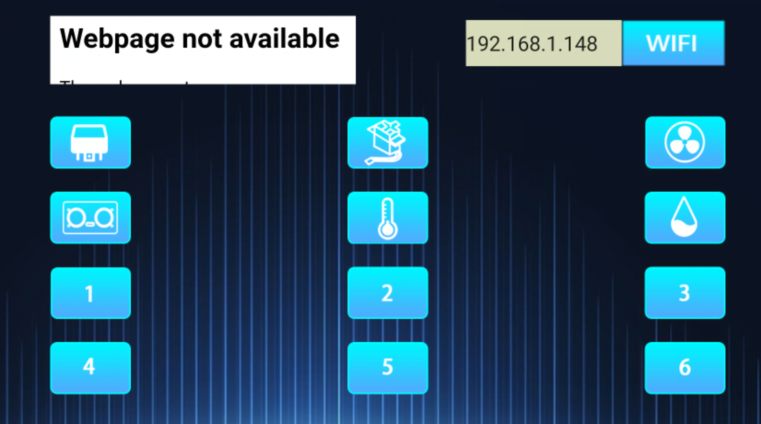


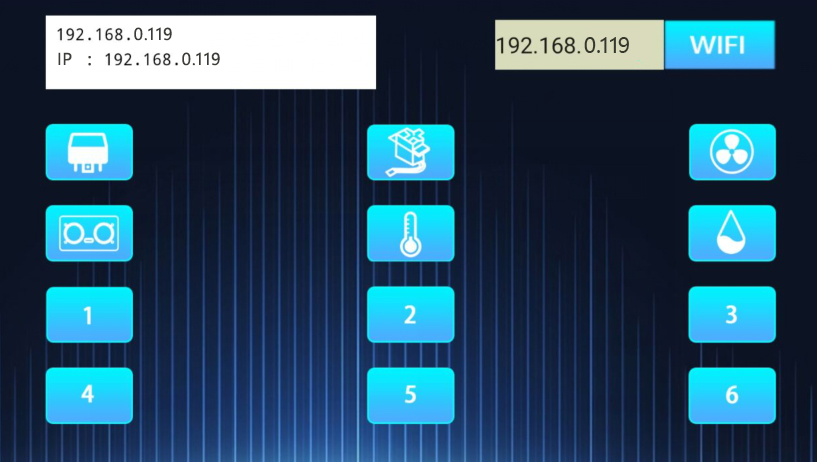








在WiFi按钮前面的文本框中输入检测到的WIFI IP地址（例如，上面串口监视器检测到的IP地址：192.168.0.119），再点击WIFI按钮，“403 Forbidden”或“Webpage not available”就会变成“192.168.0.119”。这样，就说明APP已经连接上了WiFi。



**IOS系统设备（手机/iPad）APP**

a.打开App Store。



b.在搜索框输入keyes link，点击搜索，出现下载界面，点击“”，就可以下载安装keyes link的APP。接下来的操作和安卓系统类似的，可以参考上面安卓系统的步骤进行操作。

注意：点击APP上的按钮，ESP8266串口WIFI ESP-01模块上的蓝色指示灯会闪烁，出现指示灯闪烁最亮的时候说明APP已经连接上WIFI。

APP已经连接上了WIFI后，开始进行如下操作：

1. 点击按钮，继电器打开，APP上显示，模块上的指示灯点亮；再次点击按钮，继电器关闭，APP上显示，模块上的指示灯不亮。
2. 点击按钮，舵机转动180°，APP上显示；再次点击按钮，APP上显示，舵机转动0°。
3. 点击按钮，电机（带小风扇叶）转动，APP上显示；再次点击按钮，关闭电机，APP上显示；
4. 点击按钮，超声波传感器测距，在超声波传感器前放一个物体，APP上显示（不同的距离显示不同的数字），说明此时物体离超声波传感器的距离为14cm；再次点击按钮，关闭超声波，APP上显示。
5. 点击按钮，温湿度传感器测量环境中的温度，APP上显示，说明此时环境中的温度为28℃；再次点击按钮，关闭温湿度传感器，APP上显示。
6. 点击按钮，温湿度传感器测量环境中的湿度，APP上显示，说明此时环境中的湿度为52%；再次点击按钮，关闭温湿度传感器，APP上显示。