

## Bài 6: Lập trình với Wi-Fi client - server

Bài toán: Chương trình giao tiếp 2 Wi-Fi.

Wi-Fi server lắng nghe tín hiệu từ Client. Khi nhận được tín hiệu “get” sẽ trả về tín hiệu “Server gửi lại get”;

Wi-Fi Client kết nối đến server và gửi tín hiệu “get”.

### 1. Bước 1: Mô tả thiết bị.

Thiết đặt thông số cho Wi-Fi: thiết đặt thông số cho Wi-Fi như 1 server.



Property	Value
Baud	baud_9600
Connected Chann	Multiple
Constraints	521,16,50,50
Id	esp
Id Connection	id_0
IP	192.168.4.22
Mode	Access_Point
Name	Wifi ESP8266
Password Access P	
Password ST	1234567890
Pin CHPD	CHPD,Power
Pin GND	GND,Power
Pin RX	RX,IO,3<->RX
Pin TX	TX,IO,2<->TX
Pin Vcc	Vcc,Power
Port	80
Protocol	TCP
SSID Access Point	
SSID ST	ESPLAP


Hình 1: Cài đặt thông số cho Wi-Fi

Tên thuộc tính	Ý nghĩa
Baud	Baud của Wi-Fi
Connected Channel	Số lượng truy cập vào Wi-Fi
Mode	Chế độ truy cập Wi-Fi
Password Access Point	Mật khẩu mạng Wi-Fi muốn kết nối
Password ST	Mật khẩu của Wi-Fi
SSID Access Point	SSID của Wi-Fi muốn kết nối
SSID ST	SSID của Wi-Fi
IP	Địa chỉ IP của Wi-Fi muốn kết nối (dùng cho cấu hình Client)  Địa chỉ IP của server để các thiết bị khác giao tiếp với server.(dùng khi cấu hình esp là server)
Port	Chế độ Server: Port của Server.  Chế độ Client: kết nối đến Port của Server.
Protocol	Phương thức Protocol.

Bảng 1: các thuộc tính của Wi-Fi ESP 8266

Lưu ý: Mỗi ESP8266 có giá trị baud khác nhau.

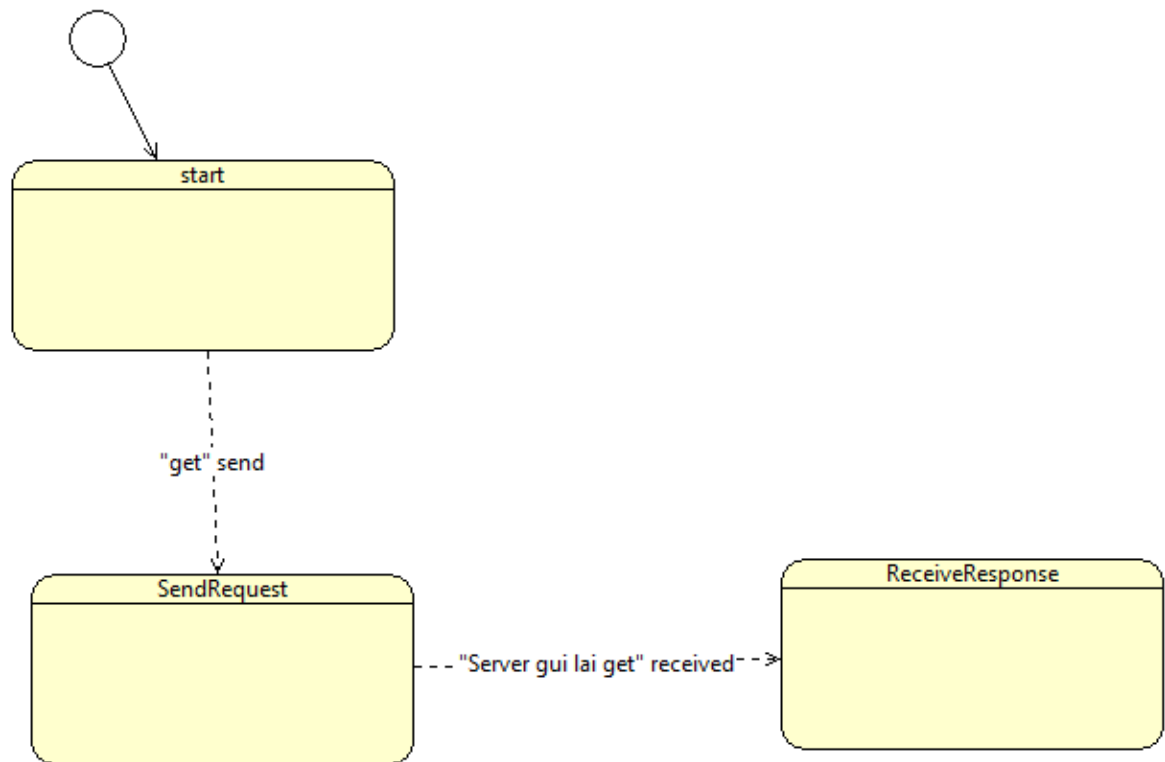
Thiết đặt thông số cho Wi-Fi client



Property	Value
Baud	baud_115200
Connected Chann	Multiple
Constraints	544,46,50,50
Id	wifi
Id Connection	id_0
IP	192.168.4.22
Mode	Station
Name	Wifi ESP8266
Password Access P	1234567890
Password ST	
Pin CHPD	CHPD,Power
Pin GND	GND,Power
Pin RX	RX,IO,3<->RX
Pin TX	TX,IO,2<->TX
Pin Vcc	Vcc,Power
Port	80
Protocol	TCP
SSID Access Point	ESPLAP
SSID ST	

Hình 2: Cài đặt thông số đèn led.

## 2. Bước 2: mô tả trạng thái chương trình client.



Hình 3: Lược đồ trạng thái cho chương trình client.

### 3. Kết quả mã nguồn client

```
1  /*
2  Source code for
3  */
4  #include <SoftwareSerial.h>
5  /*
6  Each state in the application corresponds to one the integer.
7  Numbered starting at 0
8  stateCurrent is a variable that stores the current state of the application.
9  start: 0
10 SendRequest: 1
11 ReceiveResponse: 2
12 */
13 int currentState = 0;
14 int nextState = -1;
15 /*-----Define-----*/
16 /*Define Wifi ESP8266 - wifi */
17 #define TIMEOUT 3000
18 #define RX 2 // pin TX of ESP8266 connect to pin 2 of Arduino
19 #define TX 3 //pin RX of ESP8266 connect to pin 3 of Arduino
20 SoftwareSerial wifi(RX,TX);
21 /*-----END Define-----*/
22 /*-----Prototype-----*/
23 //Read response from esp. Arduino is a receiver
24 String readResponse();
25 //Arduino send command to the esp
26 String sendCommand(String command);
27 //Close connection with a link id
28 void closeConnect();
29 void sendResponse(String content);
30 void HandleRequest(String res);
```

Hình 4: Mã nguồn sau khi phát sinh

```

31 void statestart();
32 void stateSendRequest();
33 void stateReceiveResponse();
34 /*-----END Prototype-----*/
35 void setup()
36 {
37     Serial.begin(9600); //Change baud rate according to your ESP
38     wifi.begin(115200);
39     Serial.println(sendCommand("AT+RST"));
40     Serial.println(sendCommand("AT+CWMODE=1"));
41     Serial.println(sendCommand("AT+CWJAP=\"ESPLAP\", \"1234567890\""));
42     Serial.println(sendCommand("AT+CIFSR"));
43     if(currentState == 0){
44         statestart();
45     }
46 }
47 void loop()
48 {
49     //Listening the request from server
50     String res = readResponse();
51     if(res != ""){
52         HandleRequest(res);
53     }
54     delay(300);
55     switch(currentState){
56         case 0:
57             //Event: "get" send
58             //send to client response
59             sendResponse("get");
60             stateSendRequest();

```

Hình 5: Mã nguồn sau khi phát sinh

```

61         //<case0>
62         break;
63         case 1:
64         //<case1>
65         break;
66         case 2:
67         //<case2>
68         break;
69         default:
70         break;
71     }
72 }
73 /*-----Implement-----*/
74 String readResponse()
75 {
76     String res = "";
77     long timeout = millis() + TIMEOUT;
78     while (millis() < timeout)
79     {
80         if (wifi.available())
81         {
82             char ch = wifi.read();
83             res += ch;
84         }
85     }
86     return res;
87 }
88 //Send command to esp.
89 String sendCommand(String command)
90 {

```

Hình 6: Mã nguồn sau khi phát sinh

```

91     wifi.println(command);
92     return readResponse();
93 }
94 //Close connection with a link id
95 void closeConnect()
96 {
97     Serial.println(sendCommand("AT+CIPCLOSE=5"));
98 }
99 void sendResponse(String content)
100 {
101     closeConnect();
102     Serial.println(sendCommand("AT+CIPSTART=\\"TCP\\",\\"192.168.4.22\\",80));
103     String cmd = "AT+CIPSEND=";
104     cmd += content.length();
105     Serial.println(sendCommand(cmd));
106     Serial.println(sendCommand(content));
107 }
108 void HandleRequest(String res){
109     if(res.indexOf("+IPD") >= 0)
110     {
111         //Event: "Server gui lai get" received
112         if(res.indexOf("Server gui lai get") != -1)
113         {
114             // thêm lệnh in kết quả ra màn hình
115             Serial.println("Server gui lai get");
116             stateReceiveResponse();
117         }
118         //<requestWifi>
119     }
120     delay(300);

```

Hình 7: Mã nguồn sau khi phát sinh.

```

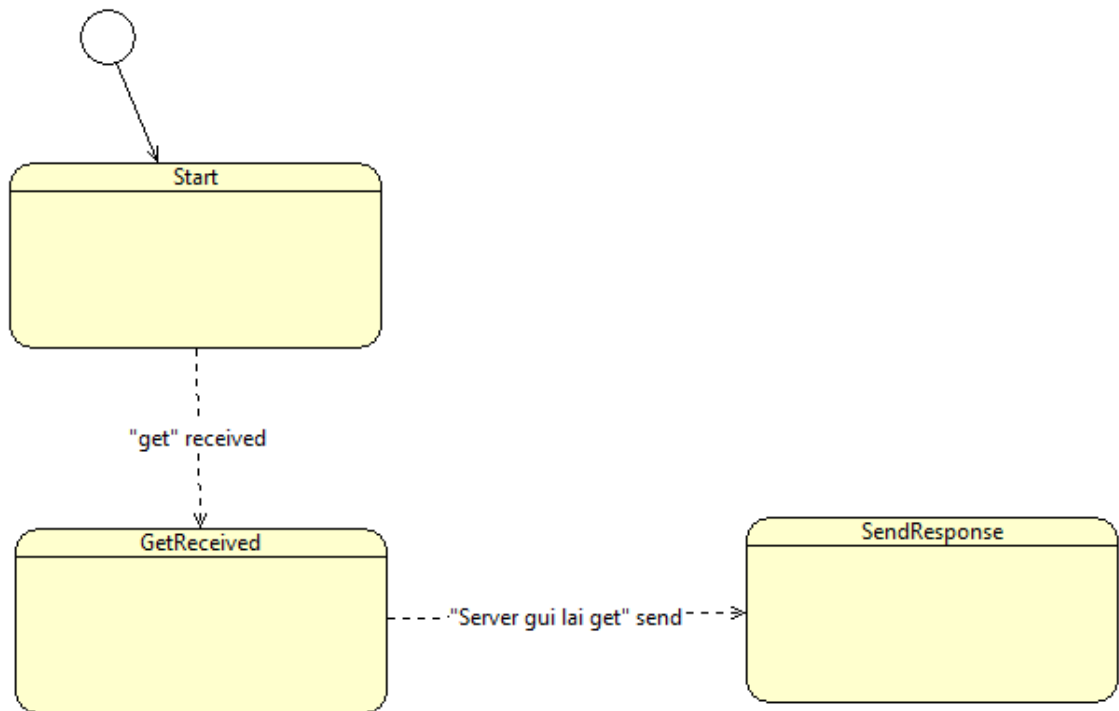
121 }
122 void statestart(){
123     currentState = 0;
124     delay(200);
125 }
126 void stateSendRequest(){
127     currentState = 1;
128     delay(200);
129 }
130 void stateReceiveResponse(){
131     currentState = 2;
132     delay(200);
133 }
134 /*-----END Implement-----*/

```

Hình 8: Mã nguồn phát sinh



#### 4. Bước 2: mô tả trạng thái chương trình server.



Hình 9: lược đồ xử lý phía server.

## 5. Mã nguồn phát sinh phía Server

```
1  /*
2  Source code for
3  */
4  #include <SoftwareSerial.h>
5  /*
6  Each state in the application corresponds to one the integer.
7  Numbered starting at 0
8  stateCurrent is a variable that stores the current state of the application.
9  Start: 0
10 GetReceived: 1
11 SendResponse: 2
12 */
13 int currentState = 0;
14 int nextState = -1;
15 /*-----Define-----*/
16 /*Define Wifi ESP8266 - esp */
17 #define TIMEOUT      3000
18 #define RX 2 // pin TX of ESP8266 connect to pin 2 of Arduino
19 #define TX 3 //pin RX of ESP8266 connect to pin 3 of Arduino
20 SoftwareSerial esp(RX,TX);
21 /*-----END Define-----*/
22 /*-----Prototype-----*/
23 //Read response from esp. Arduino is a receiver
24 String readResponse();
25 //Arduino send command to the esp
26 String sendCommand(String command);
27 //Close connection with a link id
28 void closeConnect();
29 void sendResponse(String content);
30 void HandleRequest(String res);
```

Hình 10: Mã nguồn server

```
31 void stateStart();
32 void stateGetReceived();
33 void stateSendResponse();
34 /*-----END Prototype-----*/
35 void setup()
36 {
37     Serial.begin(9600); //Change baud rate according to your ESP
38     esp.begin(9600);
39     Serial.println(sendCommand("AT+RST"));
40     Serial.println(sendCommand("AT+CWMODE=2"));
41     Serial.println(sendCommand("AT+CIFSR"));
42     Serial.println(sendCommand("AT+CWSAP=\"ESPLAP\", \"1234567890\", 5, 3));
43     Serial.println(sendCommand("AT+CIPAP=\"192.168.4.22\", \"192.168.4.22\", \"255.255.255.0\""));
44     Serial.println(sendCommand("AT+CIPMUX=1"));
45     Serial.println(sendCommand("AT+CIPSERVER=1,80"));
46     Serial.println(sendCommand("AT+CIFSR"));
47     Serial.println("Server is ready.");
48     if(currentState == 0){
49         stateStart();
50     }
51 }
52 void loop()
53 {
54     //Listening the request from server
55     String res = readResponse();
56     if(res != ""){
57         HandleRequest(res);
58     }
59     delay(300);
60     switch(currentState){
```

Hình 11: mã nguồn server

```

61         case 0:
62             //<case0>
63             break;
64         case 1:
65             //Event: "Server gui lai get" send
66             //send to client response
67             sendResponse("Server gui lai get");
68             stateSendResponse();
69             //<case1>
70             break;
71         case 2:
72             //<case2>
73             break;
74         default:
75             break;
76     }
77 }
78 /*-----Implement-----*/
79 String readResponse()
80 {
81     String res = "";
82     long timeout = millis() + TIMEOUT;
83     while (millis() < timeout)
84     {
85         if (esp.available())
86         {
87             char ch = esp.read();
88             res += ch;
89         }
90     }

```

Hình 12: mã nguồn server.

```

91         return res;
92     }
93     //Send command to esp.
94     String sendCommand(String command)
95     {
96         esp.println(command);
97         return readResponse();
98     }
99     //Close connection with a link id
100    void closeConnect()
101    {
102        Serial.println(sendCommand("AT+CIPCLOSE=5"));
103    }
104    void sendResponse(String content)
105    {
106        String cmd = "AT+CIPSEND=0,";
107        cmd += content.length();
108        Serial.println(sendCommand(cmd));
109        Serial.println(sendCommand(content));
110        closeConnect();
111    }
112    void HandleRequest(String res) {
113        if (res.indexOf("GET / HTTP/1.1") >= 0)
114        {
115            String content = "Well come to ESP LAP";
116            String cmd = "AT+CIPSEND=0,";
117            cmd += content.length();
118            Serial.println(sendCommand(cmd));
119            Serial.println(sendCommand(content));
120            closeConnect();

```

Hình 13: mã nguồn server

```

120         closeConnect();
121     }
122     else if (res.indexOf("CONNECT") >= 0)
123     {
124         Serial.println("receive...");
125     }
126     else if (res.indexOf("+IPD") >= 0)
127     {
128         //Event: "get" received
129         if(res.indexOf("get") != -1)
130         {
131             stateGetReceived();
132         }
133         //<requestWifi>
134     }
135     delay(300);
136 }
137 void stateStart() {
138     currentState = 0;
139     delay(200);
140 }
141 void stateGetReceived() {
142     currentState = 1;
143     delay(200);
144 }
145 void stateSendResponse() {
146     currentState = 2;
147     delay(200);
148 }
149 /*-----END Implement-----*/

```

Hình 14: mã nguồn server

## 6. Lắp đặt thiết bị.

## 7. Bài tập.

1. Hãy cấu hình giao tiếp giữa 2 Wi-Fi esp8266v1. Client gửi tín hiệu yêu cầu mở đèn bên server. Server thực thi lệnh bật đèn và gửi lại tín hiệu thông báo đã bật đèn.

Cách cấu hình Wi-Fi Client để kết nối đến Wi-Fi khác.



Property	Value
Baud	baud_115200
Connected Chann	Multiple
Constraints	541,77,50,50
Id	wifi
Id Connection	id_0
IP	192.168.4.22
Mode	Station
Name	Wifi ESP8266
Password Access P	1234567890
Password ST	
Pin CHPD	CHPD,Power
Pin GND	GND,Power
Pin RX	RX,IO,3<->RX
Pin TX	TX,IO,2<->TX
Pin Vcc	Vcc,Power
Port	80
Protocol	TCP
SSID Access Point	ESPLAP
SSID ST	

Hình 15: Cấu hình Wi-Fi Client kết nối đến 1 Wi-Fi Server khác.

Mode: Station

Thiết đặt Wi-Fi dạng Station.

SSID Access Point, Password Access Point: kết nối đến Wi-Fi server hay Wi-Fi trung gian.

Protocol: thiết lập phương thức protocol là UDP/ TCP tùy loại.

IP: Địa chỉ IP của Wi-Fi server trong mạng

Port: Port giao tiếp với Server.