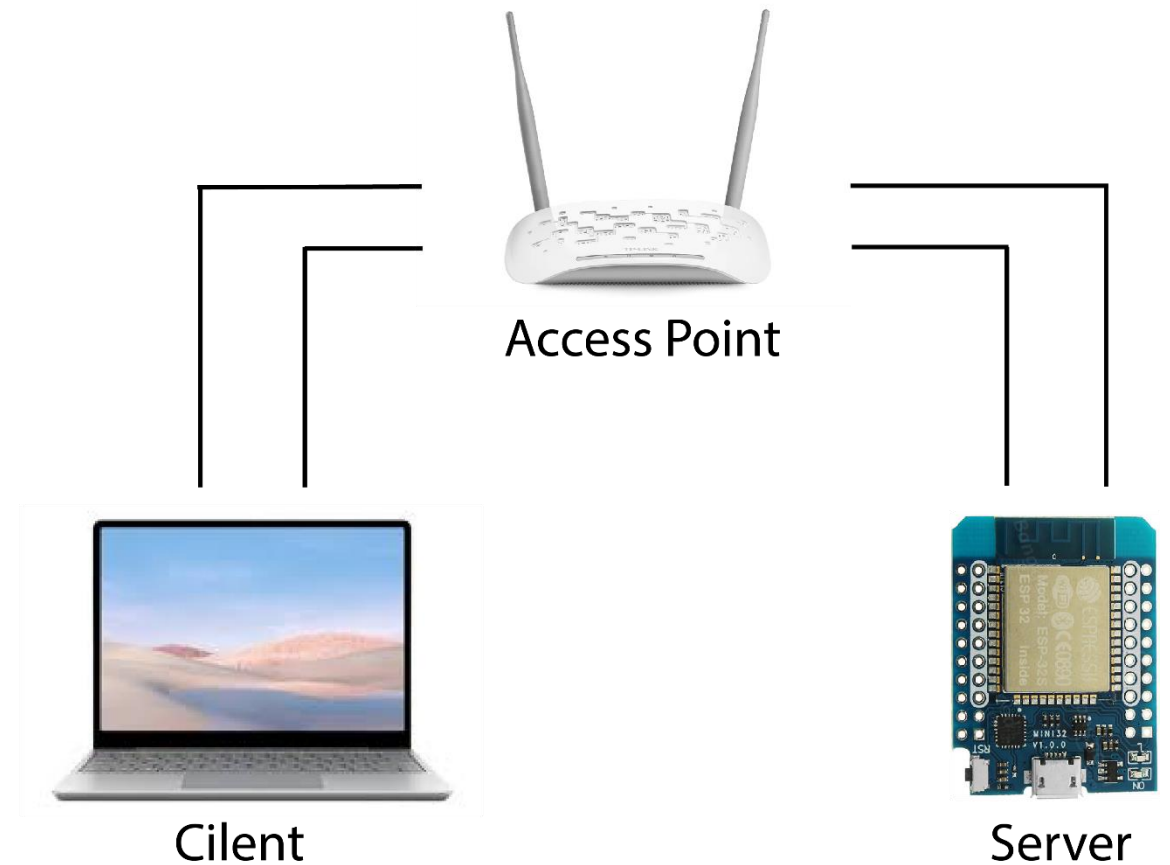
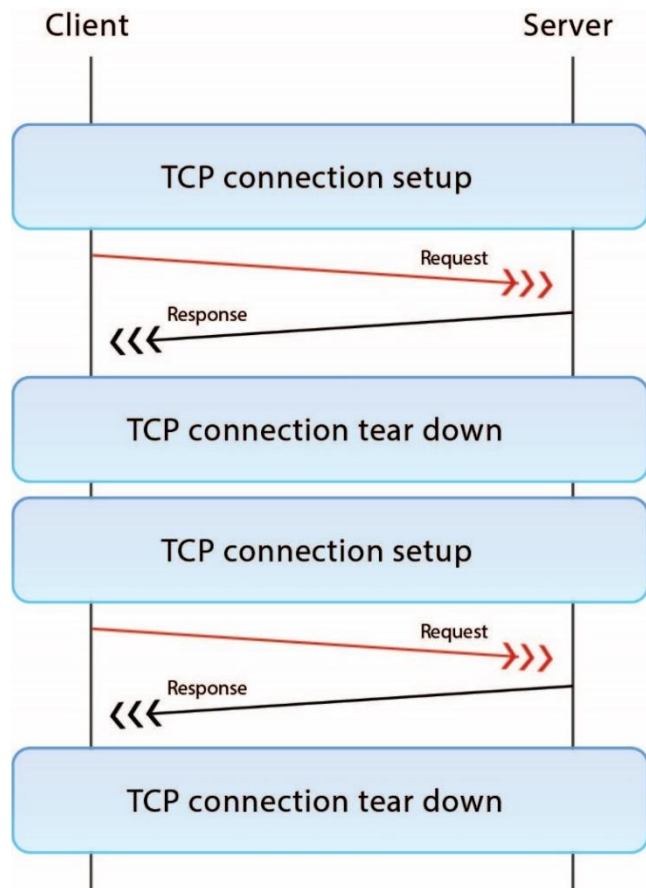


# Socket - Web Sever

# เว็บเซิร์ฟเวอร์: องค์ประกอบ

- 2 ส่วนหลัก ได้แก่
  - ส่วนของฮาร์ดแวร์ เช่น คอมพิวเตอร์ หรือ ในที่นี้ คือ โมดูล ESP32
  - ส่วนของซอฟต์แวร์ ได้แก่ HTTP เซิร์ฟเวอร์ สามารถติดต่อผ่าน URL โดยใช้ดีพอลท์พอร์ต 80 สำหรับ HTTP และ 443 สำหรับ HTTPS

# การสื่อสารกายใต้เว็บ เซิร์ฟเวอร์



# การสร้างการเชื่อมต่อผ่าน Socket

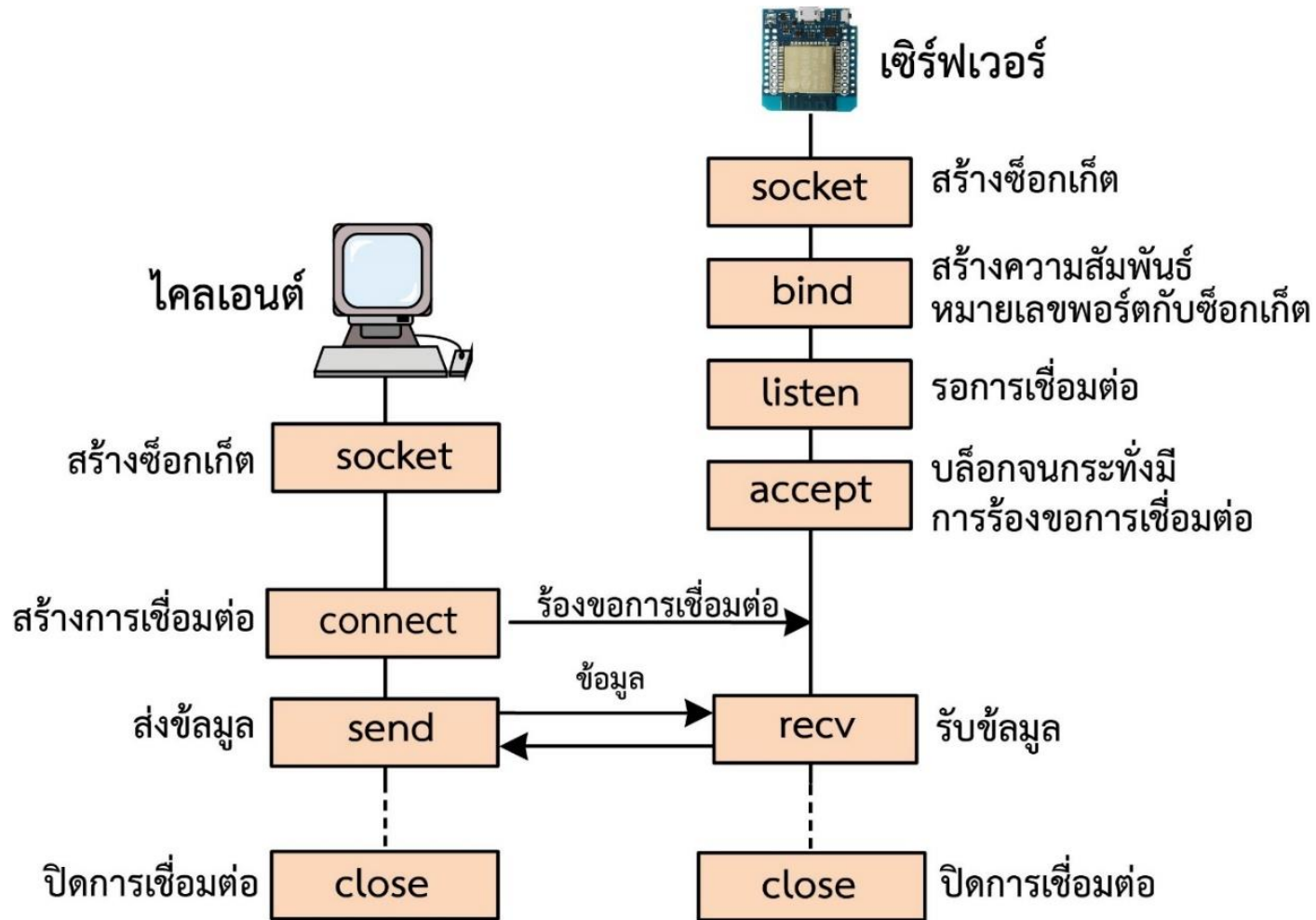
โปรโตคอล HTTP เป็นแอปพลิเคชันเลเยอร์โปรโตคอล

การเชื่อมต่อแบบ TCP/IP ผ่านซ็อกเก็ต ประกอบด้วยตัวแปร 3 ตัว ได้แก่

- Protocol family เช่น
  - AF\_INET สำหรับ IPv4
  - AF\_INET6 สำหรับ IPv6
- Type of socket :
  - SOCK\_STREAM ใช้เพื่อการใช้งาน connection-oriented
  - SOCK\_DGRAM เพื่อใช้งานแบบ datagram
- Protocol type
  - IPPROTO\_TCP สำหรับ TCP
  - IPPROTO\_UDP สำหรับ UDP

คำสั่ง

- **result = socket(protocol family, type of socket, protocol type)**



## รูปแบบการสื่อสาร

- ฟังก์ชันเซิร์ฟเวอร์
- Bind กำหนดพอร์ตแอดเดรสและ IP address เพื่อการสื่อสาร
- Listen รอการเชื่อมต่อ
- Accept รอการเชื่อมต่อจากอุปกรณ์
- ฟังก์ชันไคลเอนต์
- `connect(socket, dest address, address length)`

# TCP Server

```
1 import network
2 import socket
3 import time
4
5 SSID="dfrobotYanfa"
6 PASSWORD="hidfrobot"
7 port=10000
8 wlan=None
9 listenSocket=None
10
11 def connectWifi(ssid,passwd):
12     global wlan
13     wlan=network.WLAN(network.STA_IF)
14     wlan.active(True)
15     wlan.disconnect()
16     wlan.connect(ssid,passwd)
17     while(wlan.ifconfig()[0]!='0.0.0.0'):
18         time.sleep(1)
19     return True
```

```
21 #Catch exceptions,stop program if interrupted accidentally in the 'try'
22 try:
23     connectWifi(SSID,PASSWORD)
24     ip=wlan.ifconfig()[0] #get ip addr
25     listenSocket = socket.socket() #create socket
26     listenSocket.bind((ip,port)) #bind ip and port
27     listenSocket.listen(1) #listen message
28     listenSocket.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
29     print ('tcp waiting...')
30
31 while True:
32     print("accepting....")
33     conn,addr = listenSocket.accept() #Accept a connection,conn is
34     print(addr,"connected")
35
36 while True:
37     data = conn.recv(1024) #Receive 1024 byte of data f
38     if(len(data) == 0):
39         print("close socket")
40         conn.close() #if there is no data,close
41         break
42     print(data)
43     ret = conn.send(data) #send data
44 except:
45     if(listenSocket):
46         listenSocket.close()
47     wlan.disconnect()
48     wlan.active(False)
```

# TCP Client

```
1 import network
2 import socket
3 import time
4
5 SSID="yourSSID"
6 PASSWORD="yourPASSWD"
7 host="192.168.3.147"
8 port=1000
9 wlan=None
0 s=None
1
2 def connectWifi(ssid,passwd):
3     global wlan
4     wlan=network.WLAN(network.STA_IF)
5     wlan.active(True)
6     wlan.disconnect()
7     wlan.connect(ssid,passwd)
8     while(wlan.ifconfig()[0]!='0.0.0.0'):
9         time.sleep(1)
0     return True
```

```
22 #Catch exceptions,stop program if interrupted accidentally i
23 try:
24     connectWifi(SSID,PASSWORD)
25     ip=wlan.ifconfig()[0] #get
26     s = socket.socket() #cre
27     s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)#Se
28     s.connect((host,port)) #ser
29     s.send("hello DFRobot,I am TCP Client") #ser
30 while True:
31     data = s.recv(1024) #Recei
32     if(len(data) == 0): #if th
33         print("close socket")
34         s.close()
35         break
36     print(data)
37     ret = s.send(data)
38 except:
39     if (s):
40         s.close()
41     wlan.disconnect()
42     wlan.active(False)
```

# Socket Connection – with Browser

```
1  import network
2  import socket
3  import time
4
5  wlan=None
6  soc=None
7
8
9  def connectWifi(ssid,passwd):
10     global wlan
11     wlan=network.WLAN(network.STA_IF)
12     wlan.active(True)
13     wlan.disconnect()
14     wlan.connect(ssid,passwd)
15     while(wlan.ifconfig()[0]!='0.0.0.0'):
16         time.sleep(1)
17     print(wlan.ifconfig())
18     return True
```

```
21 def Hello_farm():
22     global soc
23     soc = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
24     soc.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
25     soc.bind((wlan.ifconfig()[0], 80))
26     soc.listen(1)
27     while True:
28         conn, addr = soc.accept()
29         print("Connection from %s" % str(addr))
30         request = conn.recv(1024)
31         conn.sendall('HTTP/1.1 200 OK\nContent-Type: text/html\nConnection: close\n\n')
32         with open('Hello_smart_farm.html', 'r') as html:
33             conn.sendall(html.read())
34             conn.sendall('\r\n')
35             conn.close()
36         print("Connection from %s closed" % str(addr))
37
38 connectWifi("username", "password")
39 Hello_farm()
```



# ไฟล์ HTML

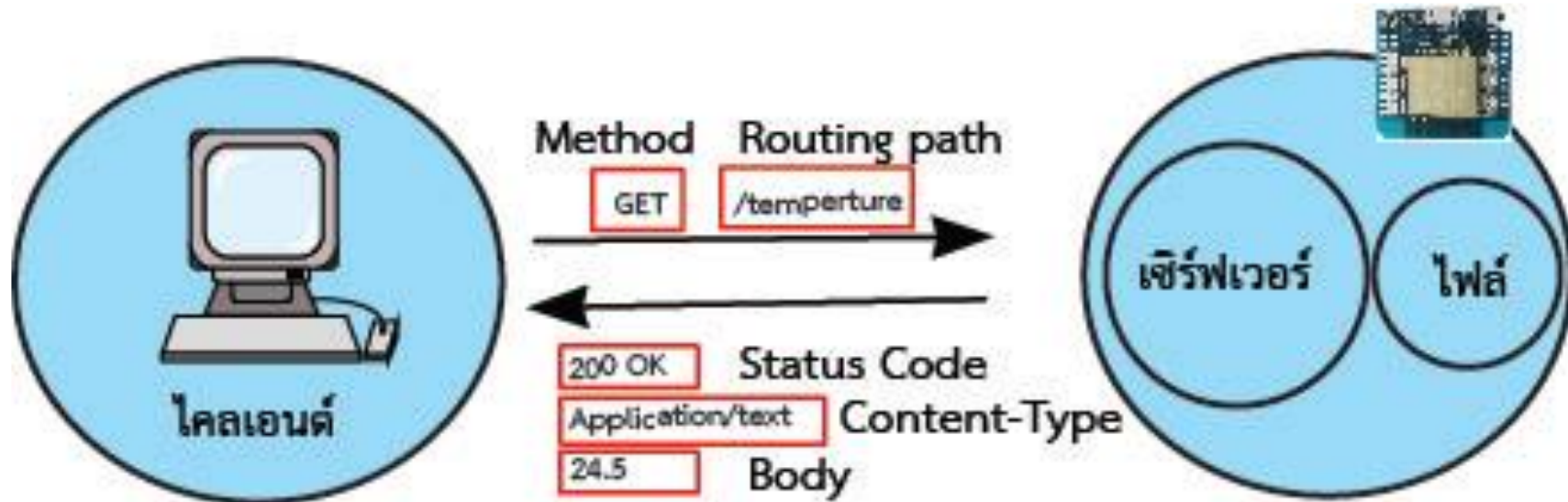
```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  |...<title> my first page </title>
5  </head>
6  <body>
7  |...<h1>Hello world</h1>
8  |...<p>Introduction to Smartfarm </p>
9  </body>
10 </html>
```

# แบบฝึกหัด

- สร้าง Socket สื่อสารระหว่างโมดูลและคอมพิวเตอร์

เว็บเซิร์ฟเวอร์และเว็บช็อกเก็ต

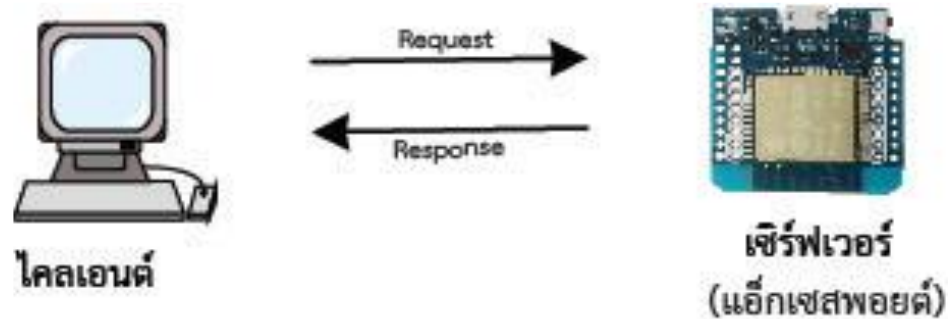
# การสื่อสารระหว่างไคลเอนต์กับเว็บเซิร์ฟเวอร์



# โมดูล MicroWebSrv เวอร์ชัน 1

- การติดต่อกับผู้ใช้ผ่านคำสั่ง GET, POST
- รองรับการทำงานแบบ REST API ในรูปแบบ JSON
- รองรับการทำงานแบบ AJAX
- สามารถสื่อสารผ่านเว็บซ็อกเก็ต
- สามารถพัฒนาเว็บเพจบนภาษา HTML (.html) และ ภาษาไพทอน (.pyhtml)
- <https://github.com/jczic/MicroWebSrv>

# Simple Web Server



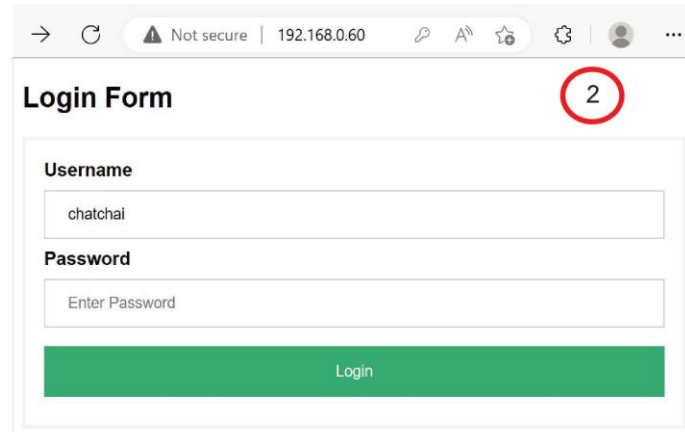
- การเรียกใช้งานโมดูล

```
from microWebSrv import MicroWebSrv
```

- สร้างไดเรกทอรีสำหรับจัดเก็บไฟล์ html ที่ www ภายในโมดูล
- การเรียกใช้งานสามารถทำได้โดยการสร้างอินสแตนส์ของเว็บและสั่งให้เซิร์ฟเวอร์เริ่มทำงาน ทั้งนี้ผู้ใช้สามารถกำหนดให้มีการสร้างเซิร์ฟเวอร์ใหม่ได้ด้วยการกำหนดให้เป็นมีค่าเป็น True

```
mws = MicroWebSrv(webPath='www/')  
mws.Start(threaded=True)
```

# สิ่งที่เราจะดำเนินการ



1. เริ่มรันโปรแกรม Simple\_web\_server.py เพื่อทำหน้าที่เป็นเว็บเซิร์ฟเวอร์
2. เปิดเบราว์เซอร์และกรอกไอพีแอดเดรสตามหมายเลขที่ปรากฏบน OLED ทำให้เปิดหน้าเว็บจากไฟล์ index.html
3. เมื่อผู้ใช้กดจาก index.html จะเห็นว่าเป็นการส่งไปยัง action="/test" method="post"
4. เมื่อเซิร์ฟเวอร์ได้รับ method="POST" จะส่งกลับด้วยไฟล์ hello\_micropython.html

# การใช้งาน พัฒนา web server

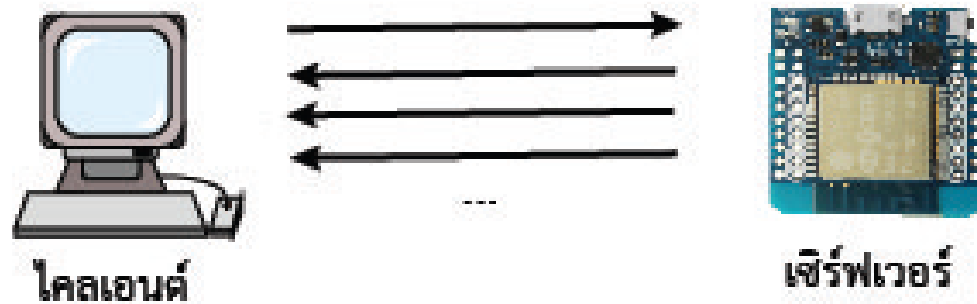
1. ไฟล์โมดูล MicroWebSrv และการเชื่อมต่อไวไฟ (connetwifi.py)
2. หน้าเว็บของโปรแกรม (index.html, hello\_micropython.html)
3. ไมโครไพทอนโค้ดทำหน้าที่เว็บเซิร์ฟเวอร์ (Simple\_web\_server.py)

ไฟล์ที่ควรมีภายในโมดูล ESP32

```
>>> uos.listdir()
[ 'connetwifi.py', 'Simple_web_server.py', 'microWebSocket.py', '
  microWebSrv.py', 'microWebTemplate.py', 'ssd1306.py', 'www']
>>> uos.listdir('www')
['hello_micropython.html', 'index.html']
```



# การส่งข้อมูลแบบต่อเนื่องบนโปรโตคอล HTTP



- การทำงานแบบนี้เรียกว่า Server-Sent Event (SSE)

```
>>> import uos
>>> uos.listdir()
['connetwifi.py', 'microWebSocket.py', 'microWebSrv.py', '
  microWebTemplate.py', 'sse.py', 'www']
>>> uos.listdir('www')
['index.html', 'result.html']
```

Not secure | 192.168.0.75

## Login Form

**Username**

**Password**

Login

Real-time Sensor Data

Not secure | 192.168.0.75/test

# Welcome

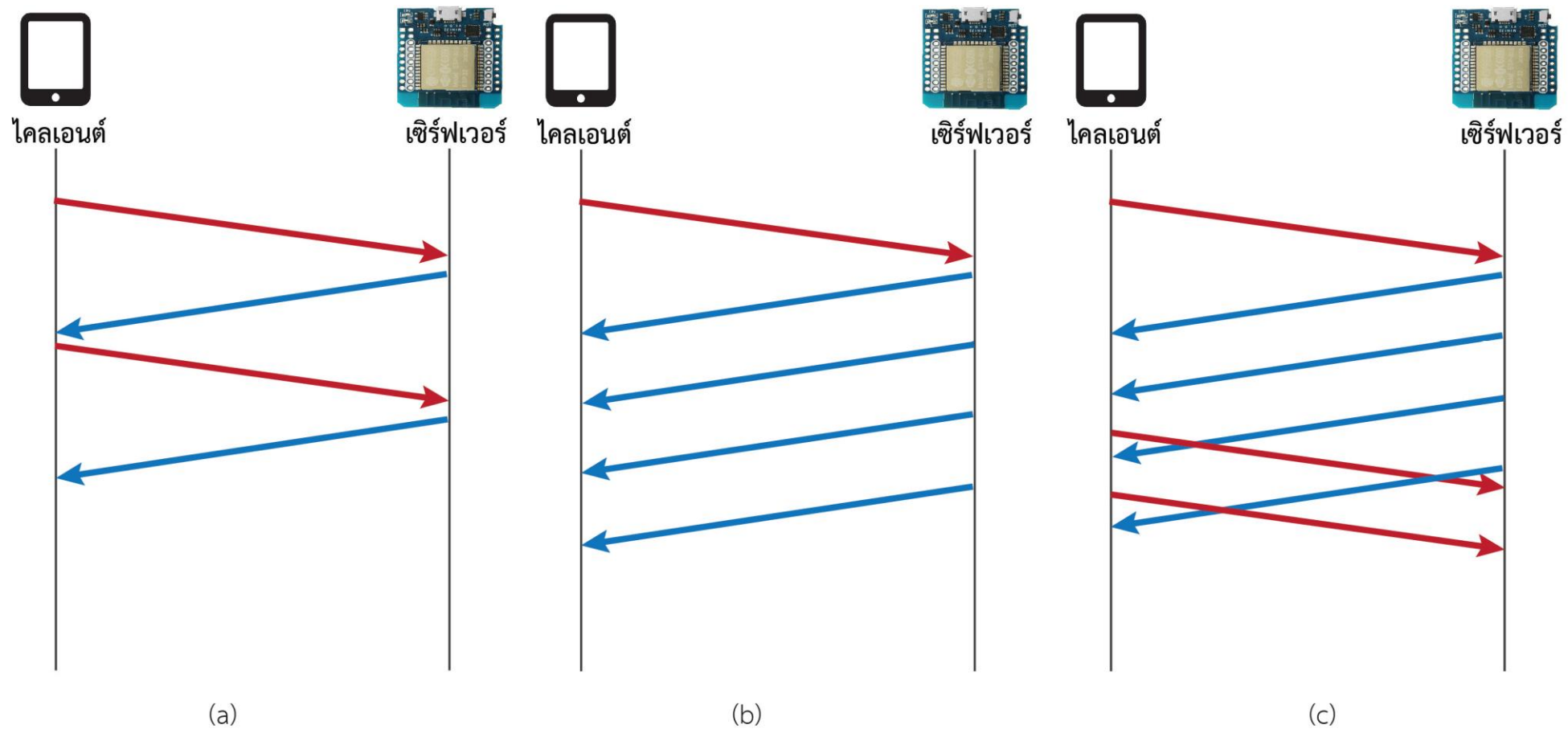
Firstname: chatchai

## Real-time Sensor Data

Current Sensor Data:

9: Temp: 53.89 C; Hall: 74.00

# Web Socket



# ไฟล์สำหรับรัน

1. ไฟล์โมดูล MicroWebSrv และการเชื่อมต่อไวไฟ (connectwifi.py)
2. หน้าเว็บของโปรแกรม (index.html, style.css)
3. ไมโครไพทอนโค้ดทำหน้าที่รองรับเว็บซ็อกเก็ต (web\_sc\_sensor.py)

```
>>> import uos
>>> uos.listdir()
['microWebSocket.py', 'microWebSrv.py', 'microWebTemplate.py', '
    web_sc_sensor.py', 'connectwifi.py', 'www']
>>> uos.listdir('www')
['index.html', 'style.css']
```

# ผลที่เกิดขึ้น

