SCM1612
Wi-Fi 6 and BLE 5 Low-Power SoC

HTTP Server Development Guide

Revision 0.1
Date 2024-3-11

Contact Information
Senscomm Semiconductor (www.senscomm.com)
Room 303, International Building, West 2 Suzhou Avenue,
SIP, Suzhou, China
For sales or technical support, please send email to
info@senscomm.com

Disclaimer and Notice

Version History

Version	Date	Description
0.1	2024-3-11	Initial draft

Table of Contents

1 Development Guide

This document serves as a guide for implementing applications that require running an HTTP server.

1.1 Overview

The SCM1612 SDK uses the esp_http_server module from ESP-IDF:

- API located in: `lib/net/esp_http_server`
- Demo located in: `api/examples/protocols/http_server`

1.2 Demo

To run the HTTP server demo, follow these steps:

1.2.1 Set up build configuration as follows.

- Select HTTP server demo as a main application.
  $ make scm1612s_defconfig
  $ make menuconfig

- Navigate to `Applications -> Protocols Demo`
- Select `Protocols Demo -> HTTP Server Demo`
- Exit & Save.

1.2.2 Set up Wi-Fi parameters as follows.

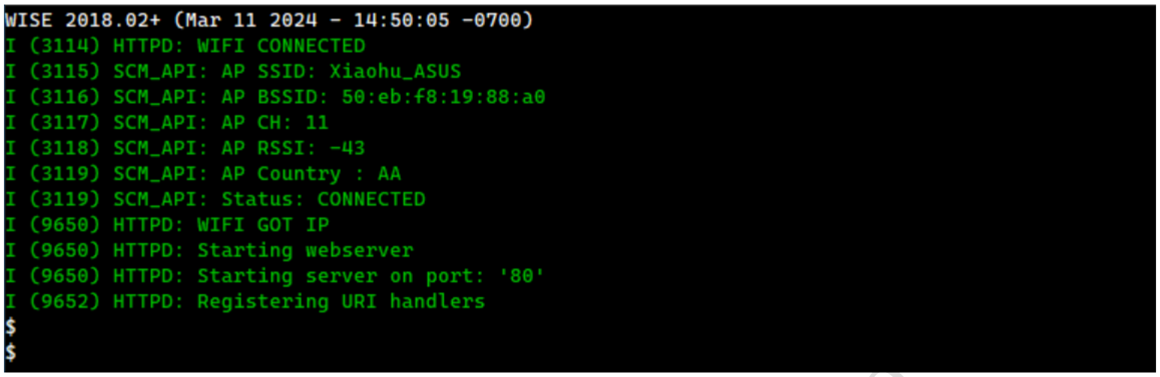
$ make menuconfig

- Navigate `Applications -> Common -> include WI-FI Configuration`
- Enter parameters in `DEMO WI-FI Configuration`*(Use Help menu for each item if needed.)*
- Exit & Save.

1.2.3 Build wise-mcuboot.bin.

$ make

- Refer to the `SDK_Getting_Started_Guide` to download the image and run it on a scm1612 EVK.
  You will see it run as follows.



```
WISE 2018.02+ (Mar 11 2024 - 14:50:05 -0700)
I (3114) HTTPD: WIFI CONNECTED
I (3115) SCM_API: AP SSID: Xiaohu_ASUS
I (3116) SCM_API: AP BSSID: 50:eb:f8:19:88:a0
I (3117) SCM_API: AP CH: 11
I (3118) SCM_API: AP RSSI: -43
I (3119) SCM_API: AP Country : AA
I (3119) SCM_API: Status: CONNECTED
I (9650) HTTPD: WIFI GOT IP
I (9650) HTTPD: Starting webserver
I (9650) HTTPD: Starting server on port: '80'
I (9652) HTTPD: Registering URI handlers
$
$
```

1.2.4 Running the demo:

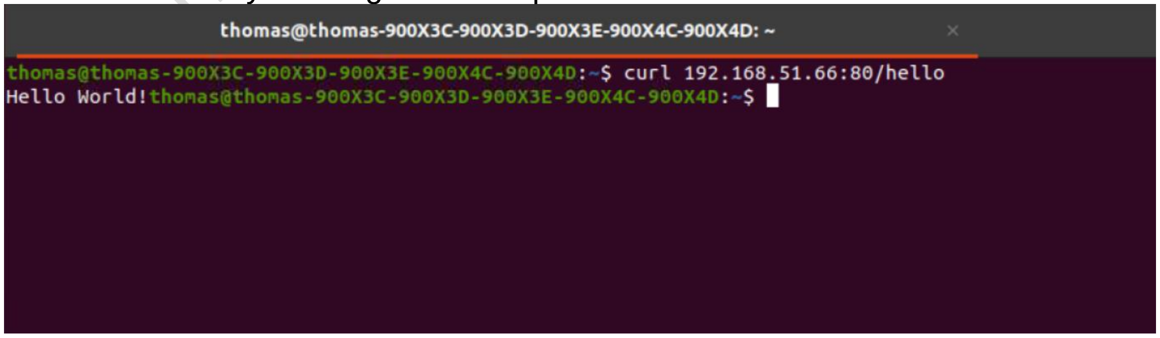
- This demo registers 3 URIs, each serving different purposes:
  - /hello: URI for responding to GET HTTP method from a client.
  - /echo: URI for responding to POST HTTP method from a client.
  - /ctrl: URI for responding to PUT HTTP method from a client.

- A HTTP client is needed for this demo and it should be running on the same network with the SCM1612 EVB hosting the HTTP server so that they can be connected to each other and exchange HTTP messages between them.

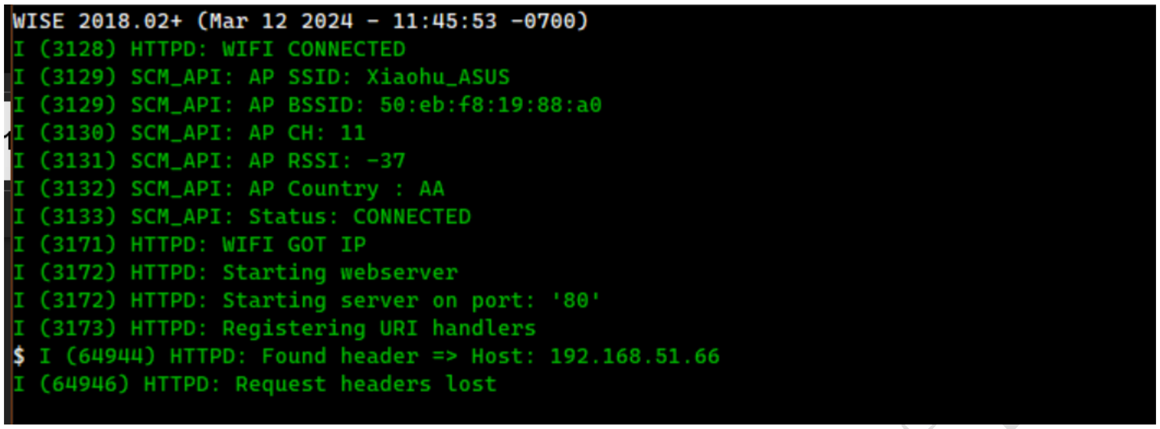
- There might be many different choices for running a HTTP client. In this example, curl command has been used on a Linux PC connected to the same network with the SCM1612 EVB.

1.2.5 Testing the URIs:

- /hello: Test by sending a GET request.

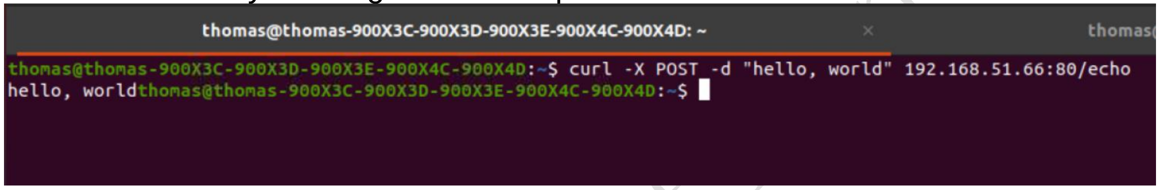


```
thomas@thomas-900X3C-900X3D-900X3E-900X4C-900X4D:~$ curl 192.168.51.66:80/hello
Hello World!thomas@thomas-900X3C-900X3D-900X3E-900X4C-900X4D:~$ 
```

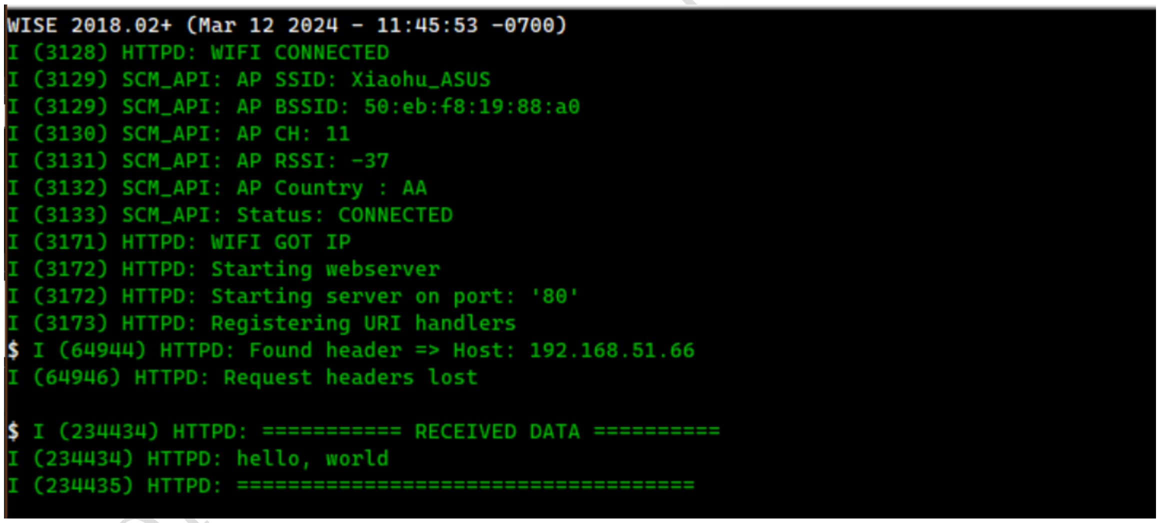


```
WISE 2018.02+ (Mar 12 2024 - 11:45:53 -0700)
I (3128) HTTPD: WIFI CONNECTED
I (3129) SCM_API: AP SSID: Xiaohu_ASUS
I (3129) SCM_API: AP BSSID: 50:eb:f8:19:88:a0
I (3130) SCM_API: AP CH: 11
I (3131) SCM_API: AP RSSI: -37
I (3132) SCM_API: AP Country : AA
I (3133) SCM_API: Status: CONNECTED
I (3171) HTTPD: WIFI GOT IP
I (3172) HTTPD: Starting webserver
I (3172) HTTPD: Starting server on port: '80'
I (3173) HTTPD: Registering URI handlers
$ I (64944) HTTPD: Found header => Host: 192.168.51.66
I (64946) HTTPD: Request headers lost
```

- /echo: Test by sending a POST request.



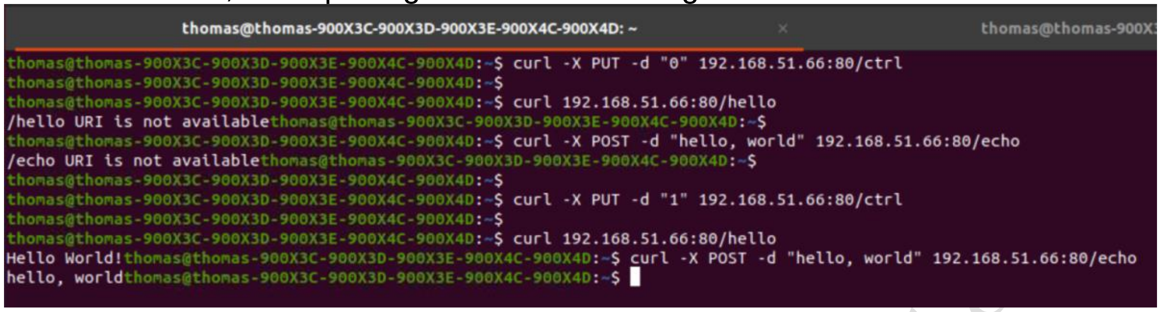
```
thomas@thomas-900X3C-900X3D-900X3E-900X4C-900X4D: ~
thomas@thomas-900X3C-900X3D-900X3E-900X4C-900X4D:~$ curl -X POST -d "hello, world" 192.168.51.66:80/echo
hello, worldthomas@thomas-900X3C-900X3D-900X3E-900X4C-900X4D:~$
```



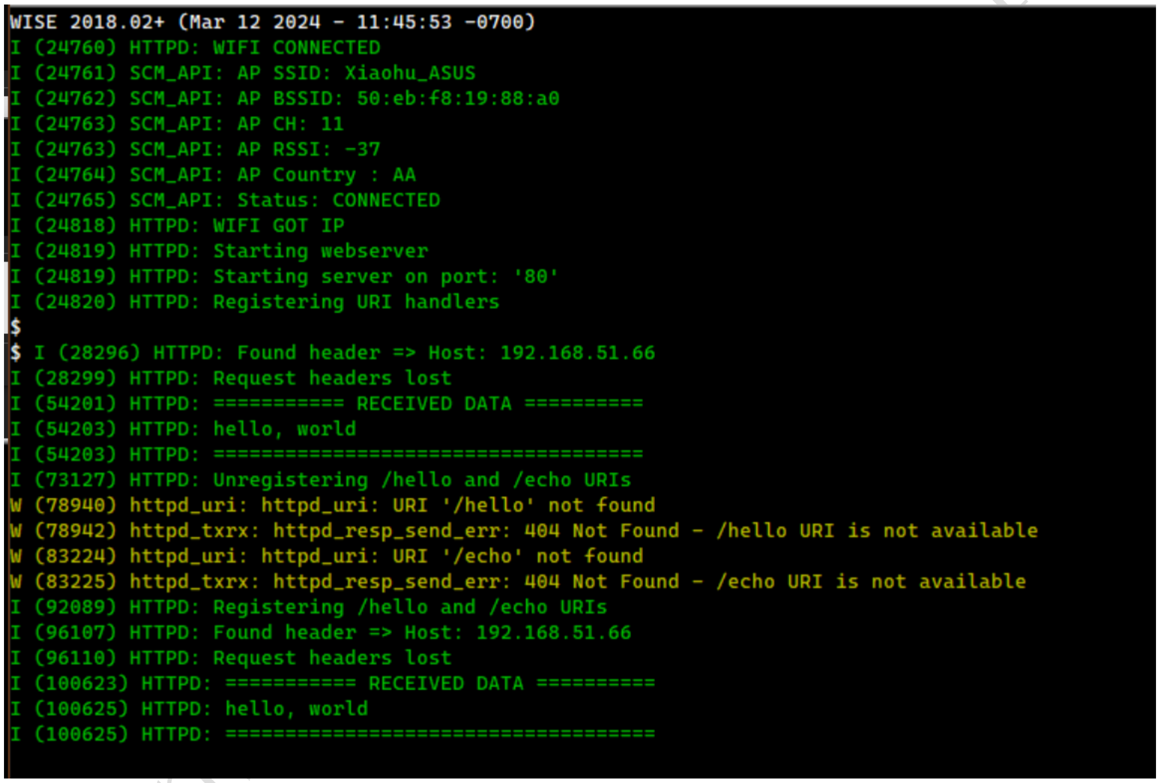
```
WISE 2018.02+ (Mar 12 2024 - 11:45:53 -0700)
I (3128) HTTPD: WIFI CONNECTED
I (3129) SCM_API: AP SSID: Xiaohu_ASUS
I (3129) SCM_API: AP BSSID: 50:eb:f8:19:88:a0
I (3130) SCM_API: AP CH: 11
I (3131) SCM_API: AP RSSI: -37
I (3132) SCM_API: AP Country : AA
I (3133) SCM_API: Status: CONNECTED
I (3171) HTTPD: WIFI GOT IP
I (3172) HTTPD: Starting webserver
I (3172) HTTPD: Starting server on port: '80'
I (3173) HTTPD: Registering URI handlers
$ I (64944) HTTPD: Found header => Host: 192.168.51.66
I (64946) HTTPD: Request headers lost

$ I (234434) HTTPD: =========== RECEIVED DATA ==========
I (234434) HTTPD: hello, world
I (234435) HTTPD: ====================================
```

- /ctrl: Test by sending a PUT request. Putting "0" into /ctrl will unregister /hello and /echo URIs, while putting "1" into /ctrl will register those URIs.



```
thomas@thomas-900X3C-900X3D-900X3E-900X4C-900X4D: ~                                        thomas@thomas-900X
thomas@thomas-900X3C-900X3D-900X3E-900X4C-900X4D:~$ curl -X PUT -d "0" 192.168.51.66:80/ctrl
thomas@thomas-900X3C-900X3D-900X3E-900X4C-900X4D:~$
thomas@thomas-900X3C-900X3D-900X3E-900X4C-900X4D:~$ curl 192.168.51.66:80/hello
/hello URI is not availablethomas@thomas-900X3C-900X3D-900X3E-900X4C-900X4D:~$
thomas@thomas-900X3C-900X3D-900X3E-900X4C-900X4D:~$ curl -X POST -d "hello, world" 192.168.51.66:80/echo
/echo URI is not availablethomas@thomas-900X3C-900X3D-900X3E-900X4C-900X4D:~$
thomas@thomas-900X3C-900X3D-900X3E-900X4C-900X4D:~$
thomas@thomas-900X3C-900X3D-900X3E-900X4C-900X4D:~$ curl -X PUT -d "1" 192.168.51.66:80/ctrl
thomas@thomas-900X3C-900X3D-900X3E-900X4C-900X4D:~$
thomas@thomas-900X3C-900X3D-900X3E-900X4C-900X4D:~$ curl 192.168.51.66:80/hello
Hello World!thomas@thomas-900X3C-900X3D-900X3E-900X4C-900X4D:~$ curl -X POST -d "hello, world" 192.168.51.66:80/echo
hello, worldthomas@thomas-900X3C-900X3D-900X3E-900X4C-900X4D:~$ 
```



```
WISE 2018.02+ (Mar 12 2024 - 11:45:53 -0700)
I (24760) HTTPD: WIFI CONNECTED
I (24761) SCM_API: AP SSID: Xiaohu_ASUS
I (24762) SCM_API: AP BSSID: 50:eb:f8:19:88:a0
I (24763) SCM_API: AP CH: 11
I (24763) SCM_API: AP RSSI: -37
I (24764) SCM_API: AP Country : AA
I (24765) SCM_API: Status: CONNECTED
I (24818) HTTPD: WIFI GOT IP
I (24819) HTTPD: Starting webserver
I (24819) HTTPD: Starting server on port: '80'
I (24820) HTTPD: Registering URI handlers
$
$ I (28296) HTTPD: Found header => Host: 192.168.51.66
I (28299) HTTPD: Request headers lost
I (54201) HTTPD: =========== RECEIVED DATA ==========
I (54203) HTTPD: hello, world
I (54203) HTTPD: ====================================
I (73127) HTTPD: Unregistering /hello and /echo URIs
W (78940) httpd_uri: httpd_uri: URI '/hello' not found
W (78942) httpd_txrx: httpd_resp_send_err: 404 Not Found - /hello URI is not available
W (83224) httpd_uri: httpd_uri: URI '/echo' not found
W (83225) httpd_txrx: httpd_resp_send_err: 404 Not Found - /echo URI is not available
I (92089) HTTPD: Registering /hello and /echo URIs
I (96107) HTTPD: Found header => Host: 192.168.51.66
I (96110) HTTPD: Request headers lost
I (100623) HTTPD: =========== RECEIVED DATA ==========
I (100625) HTTPD: hello, world
I (100625) HTTPD: ====================================
```