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

CoAP Development Guide

Revision 1.0
Date 2024-08-14

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

This document is provided on an “as-is” basis only. Senscomm reserves the right to make corrections, improvements and other changes to it or any specification contained herein without further notice.
All liability, including liability for infringement of any proprietary rights, relating to use of information in this document is disclaimed. No licenses express or implied, by estoppel or otherwise, to any intellectual property rights are granted herein.
All third party’s information in this document is provided as is with NO warranties to its authenticity and accuracy.
All trade names, trademarks and registered trademarks mentioned in this document are property of their respective owners and are hereby acknowledged.

Version History

Version	Date	Description
1.0	2024-08-14	Update Demo
0.1	2024-05-28	Initial draft

Table of Contents

1 Introduction

This guide provides detailed instructions for implementing applications using the Constrained Application Protocol (CoAP) on the SCM1612 platform.

1.1 Overview

The SCM1612 SDK integrates the libcoap library to facilitate CoAP-based communication.
The libcoap resources are organized as follows:

- API Location: `lib/net/coap`
- Demo Location: `api/examples/protocols/coap`

The CoAP module in SCM1612 can function as both a CoAP server and a CoAP client. For comprehensive information on using the libcoap APIs, please refer to the libcoap documentation.

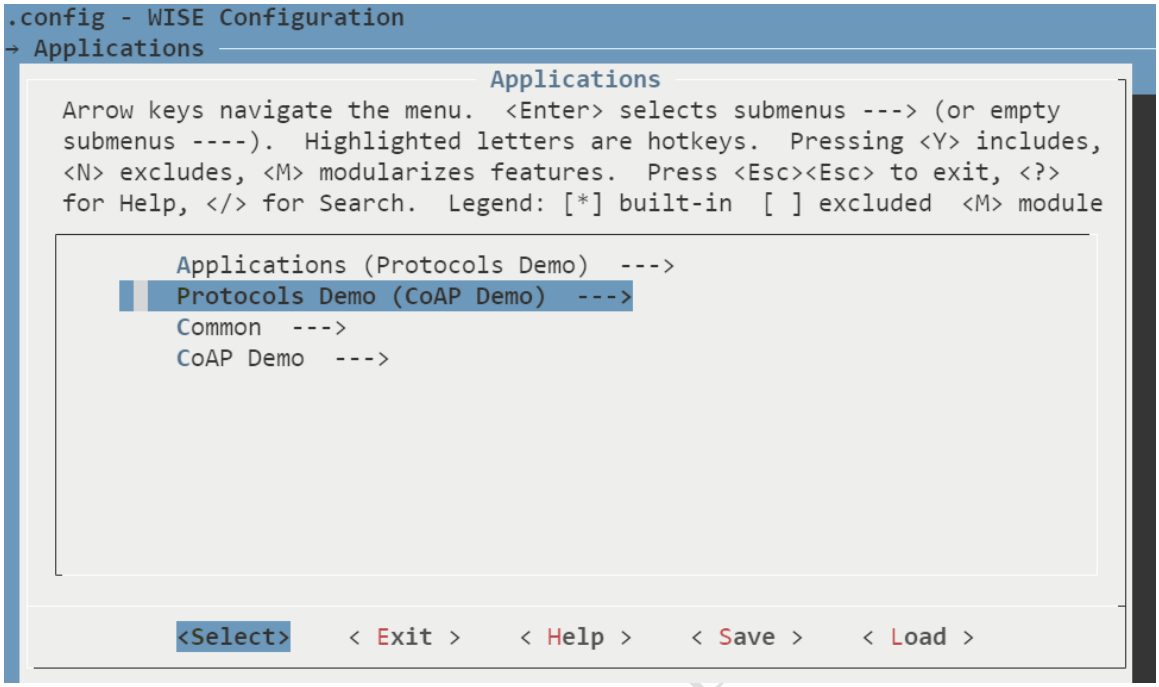
1.2 Build Instructions

To build and run the CoAP demo with CLI support, follow these steps:

1. **Enable Necessary Features**: Start by enabling the required features in the build configuration:
   $ make scm1612s_4m_defconfig
   $ make menuconfig

2. **Configure Build Options**: Navigate through the configuration menu to select the CoAP demo options:

- `Applications -> Applications -> Protocols Demo`
- `Applications -> Protocols Demo -> CoAP Demo`
- `Applications -> CoAP Demo to configure additional options`



3. **Save Configuration**: Exit the configuration menu and save your changes.
4. **Build the Firmware**: Build the `wise-mcuboot.bin` file:

   $ make
5. **Deploy the Firmware**: Refer to the SDK_Getting_Started_Guide for instructions on downloading the image and running it on the SCM1612 EVK.
6. **Verify CLI Commands**: After deployment, confirm the availability of relevant CLI commands.

2 CoAP Server Demo

2.1 Connecting to an Access Point (AP)

To connect the station interface (wlan0) to an Access Point (AP), use the Wi-Fi Station (STA) Command Line Interface (CLI) commands. For comprehensive instructions on using these commands, please refer to the **SCM1612 Wi-Fi Software Development Guide**. This guide provides a detailed mapping between Wi-Fi API functions and their corresponding CLI commands.

```
$
$ wifi help
wifi sta_start
  or:  wifi sta_stop
  or:  wifi sta_cfg <ssid> <auth> <key> <bssid> <pairwise> <hidden ap>
  or:  wifi sta_connect
  or:  wifi sta_disconnect
  or:  wifi sta_fast_connect <ssid> <auth> <bssid> <pairwise> <psk> <channel> <hidden ap>
  or:  wifi sta_get_connect
  or:  wifi sta_get_rssi
  or:  wifi sta_get_psk
  or:  wifi sta_get_country_code
  or:  wifi sta_set_reconnect <enable> <timeout> <period> <count>
  or:  wifi sta_set_ps <mode> [interval]
  or:  wifi sta_set_keepalive <enable> <interval>
  or:  wifi sta_set_country_code <code> [force]
  or:  wifi wc_set_keepalive <mode> [interval]
  or:  wifi wc_set_bcn_chk <enable>
  or:  wifi wc_set_port_filter <enable>
  or:  wifi sta_scan
  or:  wifi sta_advance_scan <scan_type> <channel>|<ssid>|<bssid>
  or:  wifi sta_scan_results <max_ap_num>
  or:  wifi sap_start
  or:  wifi sap_stop
  or:  wifi sap_cfg <ssid> <key> <ch> <hidden> <auth> <pairwise>
  or:  wifi sap_beacon <interval>
  or:  wifi sap_dtim <period>
  or:  wifi sap_deauth <sta_mac>
  or:  wifi sap_show
  or:  wifi sap_showsta
  or:  wifi ip_set <ifn> <ip> [nm] [gw]
  or:  wifi ip_reset <ifn>
  or:  wifi dhcp_start
  or:  wifi dhcp_stop
  or:  wifi dhcps_start
  or:  wifi dhcps_stop
  or:  wifi reg_evt_cb

I (6275) SCM_CLI: help    OK (0)
$ wifi sta_start
I (23043) SCM_CLI: ifname: wlan0
I (23044) SCM_CLI: sta_start    OK (0)
$
$ wifi sta_cfg Xiaohu_ASUS 0 0 00:00:00:00:00:00 1 0
I (23078) SCM_CLI: sta_cfg    OK (0)
$
$ wifi sta_connect
I (23873) SCM_CLI: sta_connect    OK (0)
$
WIFI CONNECTED
I (32161) SCM_API: AP SSID: Xiaohu_ASUS
I (32161) SCM_API: AP BSSID: 50:eb:f8:19:88:a0
I (32162) SCM_API: AP CH: 11
I (32163) SCM_API: AP RSSI: -32
I (32164) SCM_API: AP Country : AA
I (32165) SCM_API: Status: CONNECTED

WIFI GOT IP
```

Note: It is recommended to avoid using the wifi reg_evt_cb command, as it can interfere with the demo application by blocking the reception of Wi-Fi event notifications.

2.2 Running a CoAP server

This section explains how to use Command Line Interface (CLI) commands to start and stop a CoAP server.

2.2.1 Starting the CoAP Server

To start a CoAP server, the basic command is:
coap_server start

However, this command starts the CoAP server with the default configuration and without any security features. To customize the server settings, the coap_server start command supports several options:

coap_server start [**-d** max] [**-g** group] [**-p** port] [**-A** address] [**-N**] [[**-k** key] [**-h** hint]] [[**-c** certfile] [**-m**] [**-C** cafile] [**-E** oscore_conf] [**-j** keyfile]

Each of these options is described in detail below.

2.2.2 General Options

- **`-d max`:** Enables the creation of dynamic resources via the PUT method, up to a specified limit (max). If the limit is reached, a 4.06 error code is returned until one of the dynamic resources is deleted.
- **`-g group`:** *Joins the specified multicast group upon startup.*
- **`-p port`:** Specifies the port on the given address for listening to incoming connections. If (D)TLS is supported, the server will also listen on port + 1 for (D)TLS connections. The default port is 5683 if not specified.
- **`-A addr`:** Sets the local address of the interface on which the server will listen.
- **`-N`:** Sends NON-confirmable messages for "observe" responses. If this option is not specified, a confirmable response will be sent. Even when set, every fifth response will still be confirmable as required by RFC 7641.

2.2.3 Pre-Shared Key (PSK) Options

- **`-h hint`:** Specifies the pre-shared key hint to use for inbound connections. The default value is "CoAP". This field cannot be empty if defined.

- **`-k key`:** Defines the pre-shared key for inbound connections. This field cannot be empty if defined. Note: If the -c cafile option is defined, you must also define -k key to enable the server to support both PSK and PKI.

2.2.4 Public Key Infrastructure (PKI) Options

- **`-c certfile`:** Uses the specified PEM file containing the CERTIFICATE and PRIVATE KEY information. Note: If -k key is defined, you must also define -C cafile to enable the server to support both PSK and PKI.

- **`-m`:** Instructs the server to buffer the certificate files.

- **`-C cafile`:** Specifies a PEM file containing the CA Certificate that was used to sign the certfile defined with -c certfile. If defined, this CA Certificate is provided to the client during TLS setup, triggering client certificate validation. If certfile is self-signed, you must use the same filename for both certfile and cafile to trigger validation (e.g., -c certfile -C certfile).

- **`-j keyfile`:** Defines the key file used by the PKI.

2.2.5 OSCORE Options

- **`-E oscore_conf`:** Specifies the OSCORE configuration file.

Example Command
The following example starts a CoAP server with support for PKI and OSCORE, using the specified certificate and key files:

```
coap_server start -j /coap/certs/coap_server.key -C /coap/certs/coap_ca.pem -c /coap/certs/coap_server.crt -m -E /coap/oscore/coap_server_oscore.conf
```

Note: Ensure that all certification files are uploaded as described in section Uploading Certificate Files before starting the server.

2.2.6 Stopping the CoAP Server

To stop the CoAP server and release the allocated resources, use the following command:

```
coap_server stop
```

2.3 Enabling the mDNS Responder (Optional)

```
WIFI GOT IP

$
$ mdns init
$ mdns wlan0 start
$
$
```

Enabling the mDNS (Multicast DNS) responder is an optional step. If you choose not to enable it, you can proceed by directly using the device's IP address in the subsequent steps.

3 CoAP Client Demo

3.1 Running a CoAP Client

The CoAP client demo supports various Command Line Interface (CLI) commands, including the following:

- **GET/PUT/POST/DELETE:** Perform a GET, PUT, POST, or DELETE request.
- **GET/PUT/POST/DELETE with Security:** Execute a GET, PUT, POST, or DELETE request with security enabled.
- **Security Modes:** Support for PSK (Pre-Shared Key) and PKI (Public Key Infrastructure) security.
- **OSCORE:** Object Security for Constrained RESTful Environments (OSCORE).
- **Subscription/Observation:** Subscribe to or observe a resource.

Note: Using the coap:// scheme will disable security, whereas the coaps:// scheme will enable security based on the configured settings.

```
$ coap_client
coap_client v4.3.4 -- a small CoAP implementation
Copyright (C) 2010-2023 Olaf Bergmann <bergmann@tzi.org> and others

Build: libcoap-posix4.3.4
TLS Library: Mbed TLS - runtime 2.16.2, libcoap built for 2.16.2
(DTLS and no TLS support; PSK, PKI, no PKCS11, and no RPK support)
(No OSCORE)
(No WebSockets)

Usage: coap_client [-a addr] [-b [num,]size] [-e text] [-l loss]
                   [-m method] [-o file] [-p port] [-r] [-s duration] [-t type]
                   [-v num] [-w] [-A type] [-B seconds]
                   [-E oscore_conf_file] [-G count] [-H hoplimit]
                   [-K interval] [-N] [-O num,text] [-P scheme://address[:port]
                   [-T token] [-U]  [-V num] [-X size]
                   [[-h match_hint_file] [-k key] [-u user]]
                   [[-c certfile] [-j keyfile] [-n] [-C cafile]
                   [-J pkcs11_pin] [-R trust_casfile]
                   [-S match_pki_sni_file]] URI
           URI can be an absolute URI or a URI prefixed with scheme and host
```

Connect to an AP: Follow the instructions in section 2.2 to connect to an AP.

```
$ wifi sta_start
I (220275) SCM_CLI: ifname: wlan0
I (220275) SCM_CLI: sta_start    OK (0)
$ wifi sta_cfg HUAWEI-Test 0 0 00:00:00:00:00:00 1 0
I (220312) SCM_CLI: sta_cfg    OK (0)
$ wifi sta_connect
I (220334) SCM_CLI: sta_connect    OK (0)
$ wifi dhcp_start
I (221087) SCM_CLI: dhcp_start    OK (0)
$
WIFI CONNECTED
I (221977) SCM_API: AP SSID: HUAWEI-Test
I (221977) SCM_API: AP BSSID: c0:b4:7d:33:d0:00
I (221978) SCM_API: AP CH: 6
I (221979) SCM_API: AP RSSI: -47
I (221981) SCM_API: AP Country :
I (221981) SCM_API: Status: CONNECTED

WIFI GOT IP
```

Note: Before running the CoAP client demo, ensure that the PKI and OSCORE files are loaded into the system.

3.2 CoAP Client Commands

3.2.1 Basic GET Request

coap_client -m get coap://californium.eclipseprojects.io

```
$ coap_client -m get coap://californium.eclipseprojects.io
$ I (452477) CoAP_client: DNS lookup succeeded. IP=20.47.97.44
*****************************************************************
CoAP RFC 7252                                   Cf 3.13.0-SNAPSHOT
*****************************************************************
This server is using the Eclipse Californium (Cf) CoAP framework
published under EPL+EDL: http://www.eclipse.org/californium/

Note: the data sent to this server is public visible to other
      users! Don't send data, which requires data privacy.

(c) 2014-2023 Institute for Pervasive Computing, ETH Zurich
              and others
mail: cf-dev@eclipse.org
*****************************************************************
$
```

Using Non-Confirmable Messages for Broadcasting

coap_client -m get coap://255.255.255.255 -N

```
$ coap_client -m get coap://255.255.255.255 -N
$ I (73010) CoAP_client: DNS lookup succeeded. IP=255.255.255.255
This is a test server made with libcoap (see https://libcoap.net)
Copyright (C) 2010--2023 Olaf Bergmann <bergmann@tzi.org> and others
```

3.2.2 GET Request with PSK Security

coap_client -m get -u password -k sesame coaps://californium.eclipseprojects.io

```
$ coap_client  -m get -u password -k sesame coaps://californium.eclipseprojects.io
$ I (3462215) CoAP_client: DNS lookup succeeded. IP=20.47.97.44
****************************************************************
CoAP RFC 7252                                   Cf 3.13.0-SNAPSHOT
****************************************************************
This server is using the Eclipse Californium (Cf) CoAP framework
published under EPL+EDL: http://www.eclipse.org/californium/

Note: the data sent to this server is public visible to other
      users! Don't send data, which requires data privacy.

(c) 2014-2023 Institute for Pervasive Computing, ETH Zurich
              and others
mail: cf-dev@eclipse.org
****************************************************************
```

3.2.3 GET Request with PKI Security

coap_client -m get -C /coap/certs/coap_ca.pem -c /coap/certs/coap_client.crt -j /coap/certs/coap_client.key coaps://californium.eclipseprojects.io

```
$
$ coap_client -m get -C /coap/certs/coap_ca.pem -c /coap/certs/coap_client.crt  -j /coap/certs/coap_client.key coaps:
//californium.eclipseprojects.io
$ I (468202) CoAP_client: DNS lookup succeeded. IP=20.47.97.44
****************************************************************
CoAP RFC 7252                                   Cf 3.13.0-SNAPSHOT
****************************************************************
This server is using the Eclipse Californium (Cf) CoAP framework
published under EPL+EDL: http://www.eclipse.org/californium/

Note: the data sent to this server is public visible to other
      users! Don't send data, which requires data privacy.

(c) 2014-2023 Institute for Pervasive Computing, ETH Zurich
              and others
mail: cf-dev@eclipse.org
****************************************************************
```

Note: For this case, please make sure the CoAP server has supported PKI security correctly.

3.2.4 GET/PUT/DELETE Operations

```
coap_client -m get coap://[192.168.3.18]/Senscomm

coap_client -m put coap://[192.168.3.18]/Senscomm -e "ABC"

coap_client -m get coap://[192.168.3.18]/Senscomm

coap_client -m delete coap://[192.168.3.18]/Senscomm
```

```
$ coap_client -m get coap://[192.168.3.18]/Senscomm
$ I (3714208) CoAP_client: DNS lookup succeeded. IP=192.168.3.18
4.04 Not Found

$ coap_client -m put coap://[192.168.3.18]/Senscomm -e "ABC"
$ I (3717474) CoAP_client: DNS lookup succeeded. IP=192.168.3.18

$ coap_client -m get coap://[192.168.3.18]/Senscomm
$ I (3720786) CoAP_client: DNS lookup succeeded. IP=192.168.3.18
ABC
$ coap_client -m delete coap://[192.168.3.18]/Senscomm
$ I (3725967) CoAP_client: DNS lookup succeeded. IP=192.168.3.18

$
$ coap_client -m get coap://[192.168.3.18]/Senscomm
$ I (3732762) CoAP_client: DNS lookup succeeded. IP=192.168.3.18
4.04 Not Found
```

Note: For this case, please make sure the CoAP Server has the correct URI already. If using SCM1612 CoAP Server demo, please use `-d max` option when starting CoAP Server, please refer to General Options. For example, start CoAP Server with command coap_server start -d 1.

3.2.5 Subscribe to/Observe a Resource

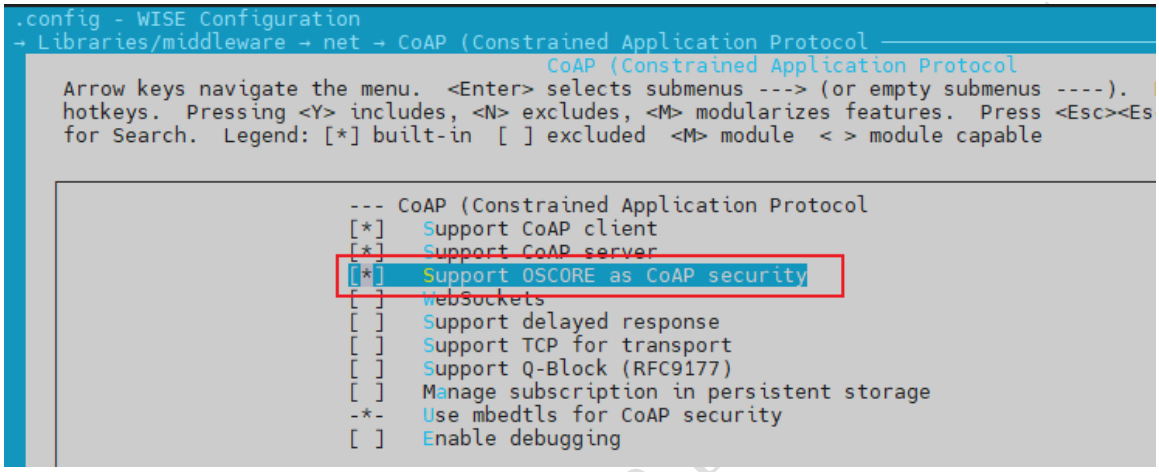
`-s duration`: Subscribe to/Observe a Resource for a Given Duration (in seconds):

```
coap_client -s 60 -m get coap://[192.168.3.18]/time
```

```
$ coap_client -s 60 -m get coap://[192.168.3.18]/time
$ I (3844619) CoAP_client: DNS lookup succeeded. IP=192.168.3.18
Jan 01 01:09:05Jan 01 01:09:06Jan 01 01:09:07Jan 01 01:09:08Jan 01 01:09:09Jan 01 01:09:10Jan 01 01:09:11Jan 01 01:09
:12Jan 01 01:09:13Jan 01 01:09:14Jan 01 01:09:15Jan 01 01:09:16Jan 01 01:09:17Jan 01 01:09:18Jan 01 01:09:19Jan 01 01
:09:20Jan 01 01:09:21Jan 01 01:09:22Jan 01 01:09:23Jan 01 01:09:24Jan 01 01:09:25Jan 01 01:09:26Jan 01 01:09:27Jan 01
 01:09:28Jan 01 01:09:29Jan 01 01:09:30Jan 01 01:09:31Jan 01 01:09:32Jan 01 01:09:33Jan 01 01:09:34Jan 01 01:09:35Jan
 01 01:09:36Jan 01 01:09:37Jan 01 01:09:38Jan 01 01:09:39Jan 01 01:09:40Jan 01 01:09:41Jan 01 01:09:42Jan 01 01:09:43
Jan 01 01:09:44Jan 01 01:09:45Jan 01 01:09:46Jan 01 01:09:47Jan 01 01:09:48Jan 01 01:09:49Jan 01 01:09:50Jan 01 01:09
:51Jan 01 01:09:52Jan 01 01:09:53Jan 01 01:09:54Jan 01 01:09:55Jan 01 01:09:56Jan 01 01:09:57Jan 01 01:09:58Jan 01 01
:09:59Jan 01 01:10:00Jan 01 01:10:01Jan 01 01:10:02Jan 01 01:10:03Jan 01 01:10:04Jan 01 01:10:05
```

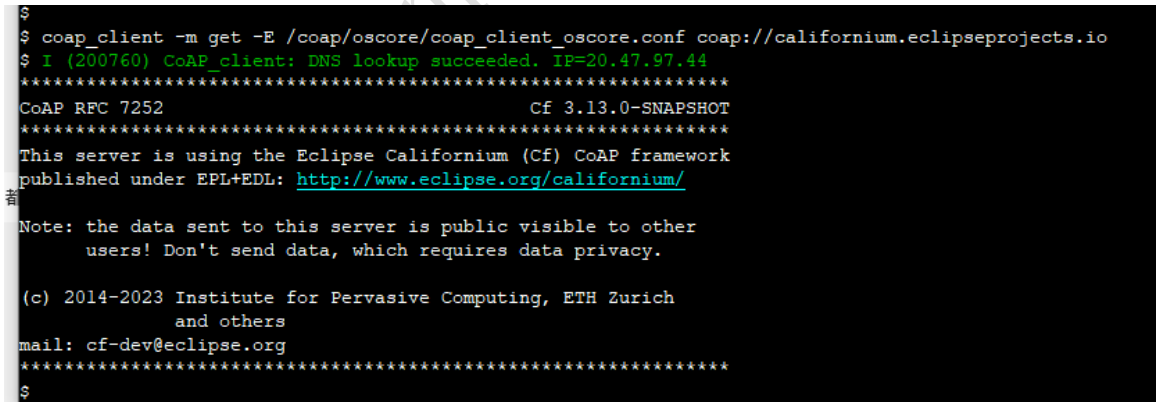
3.3 Running with OSCORE Security

To use OSCORE security, ensure that "Support OSCORE as CoAP security" is enabled in the configuration menu.



3.3.1 GET Request with OSCORE Security

coap_client -m get -E /coap/oscore/coap_client_oscore.conf coap://californium.eclipseprojects.io



3.3.2 GET Request with PKI + OSCORE Security

coap_client -m get -E /coap/oscore/coap_client_oscore.conf -C /coap/certs/coap_ca.pem -c /coap/certs/coap_client.crt -j /coap/certs/coap_client.key coaps://californium.eclipseprojects.io

```
$ coap_client -m get -E /coap/oscore/coap_client_oscore.conf -C /coap/certs/coap_ca.pem -c /coap/certs/coap_client.cr
t  -j /coap/certs/coap_client.key coaps://californium.eclipseprojects.io
$ I (231877) CoAP_client: DNS lookup succeeded. IP=20.47.97.44
****************************************************************
CoAP RFC 7252                                  Cf 3.13.0-SNAPSHOT
****************************************************************
This server is using the Eclipse Californium (Cf) CoAP framework
published under EPL+EDL: http://www.eclipse.org/californium/

Note: the data sent to this server is public visible to other
      users! Don't send data, which requires data privacy.

(c) 2014-2023 Institute for Pervasive Computing, ETH Zurich
              and others
mail: cf-dev@eclipse.org
****************************************************************
```

3.3.3 GET Request with PSK + OSCORE Security

coap_client -m get -E /coap/oscore/coap_client_oscore.conf -u password -k sesame coaps://californium.eclipseprojects.io

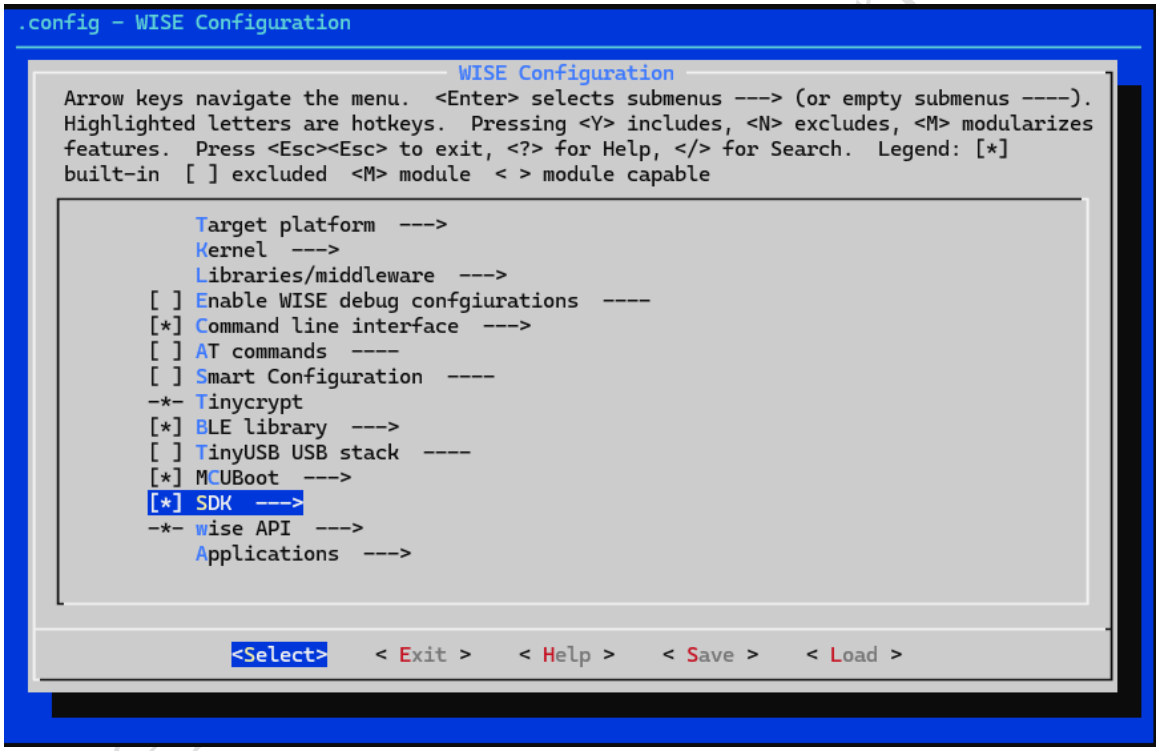
```
$
$ coap_client -m get -E /coap/oscore/coap_client_oscore.conf -u password -k sesame coaps://californium.eclipseproject
s.io
$ I (271778) CoAP_client: DNS lookup succeeded. IP=20.47.97.44
****************************************************************
CoAP RFC 7252                                  Cf 3.13.0-SNAPSHOT
****************************************************************
This server is using the Eclipse Californium (Cf) CoAP framework
published under EPL+EDL: http://www.eclipse.org/californium/

Note: the data sent to this server is public visible to other
      users! Don't send data, which requires data privacy.

(c) 2014-2023 Institute for Pervasive Computing, ETH Zurich
              and others
mail: cf-dev@eclipse.org
****************************************************************
$
```

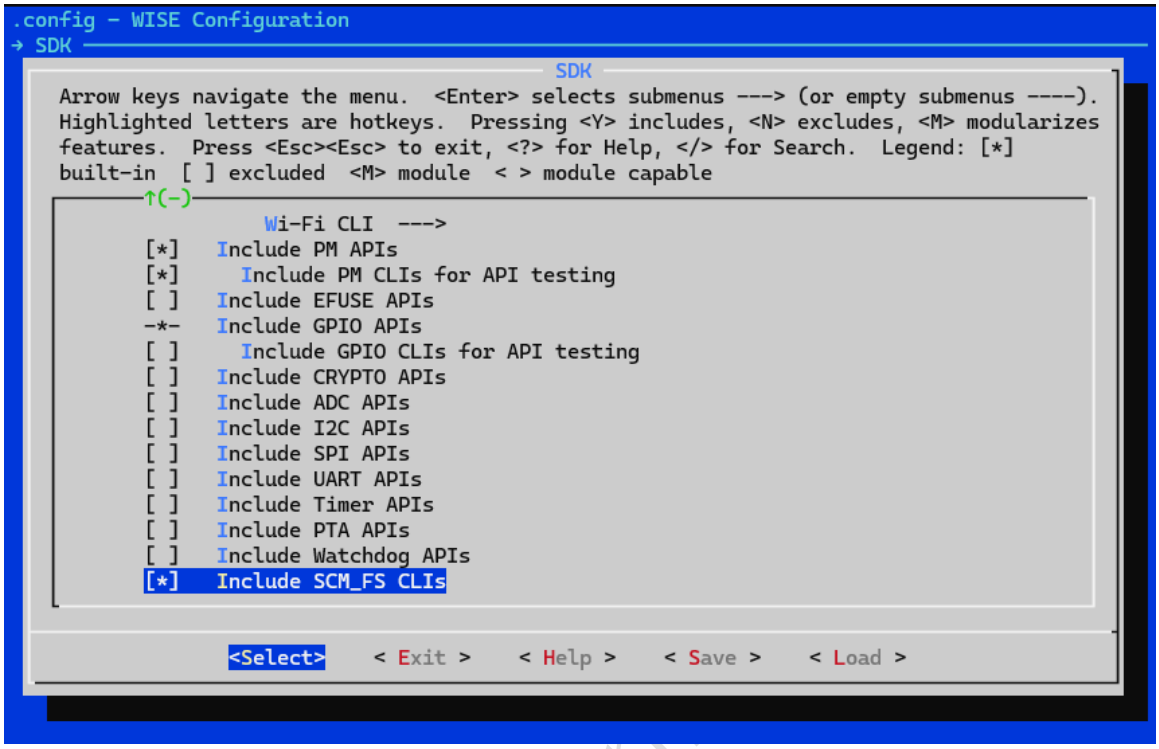
4 Uploading Certificate Files

4.1 Enabling SCM_FS CLI Commands

To configure the SCM_FS CLI for file upload:
Access Configuration Menu:

- Navigate to the configuration menu.
- Choose SDK.
- Select Include SCM_FS CLIs.





The SCM_FS CLI supports several commands, including:

- **`fs load`:** Upload a file from the local PC via YMODEM.
- **`fs read`:** Read the content of a file.
- **`fs write`:** Write data to a file.
- **`fs rm`:** Remove a file.
- **`fs size`:** Query the size of a file.

```
$ help fs
Usage: fs load <filename>
  or:  fs read <filename>
  or:  fs write <filename> <content>
  or:  fs rm <filename>
  or:  fs size <filename>
CLI for scm_fs operations
$ 
```

4.2 Uploading Certificate Files

For the CoAP demo, the `fs load` command is used to upload certificate files. Follow the steps below to upload a file into WISE for demo purposes:

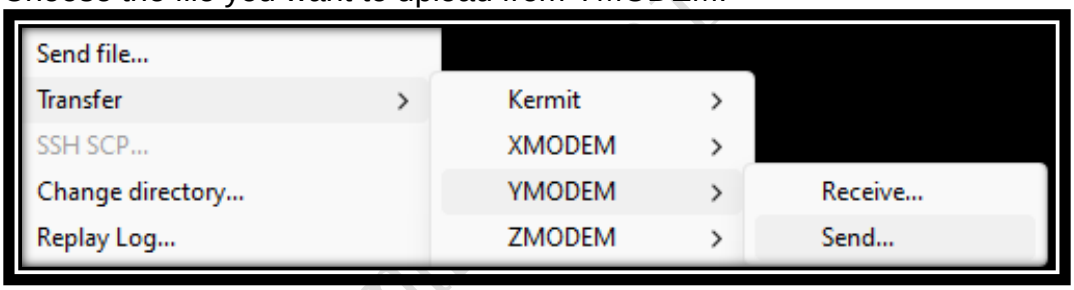
4.2.1 Upload the File:

Use the fs load command and specify the file name under which you want to save the uploaded file.

```
$ fs load /coap/certs/coap_ca.pem
load local file to /coap/certs/coap_ca.pem
C
```

4.2.2 Select the File:

Choose the file you want to upload from YMODEM.



coap_ca.pem	8/2/2024 2:20 PM	PEM File	2 KB
coap_client.crt	8/2/2024 2:20 PM	Security Certificate	1 KB
coap_client.key	8/2/2024 2:20 PM	KEY File	1 KB

4.2.3 Read the Uploaded File:

Use the `fs read` command along with the file name to read the content of the specific file.

```
$ fs read /coap/certs/coap_ca.pem
read /coap/certs/coap_ca.pem
size: 1538
-----BEGIN CERTIFICATE-----
MIICDDCCAbKgAwIBAgIIPKO8L7vZoqAwCgYIKoZIzj0EAwIwXDEQMA4GA1UEAxMH
Y2Ytcm9vdDEUMBIGA1UECxMLQ2FsaWZvcm5pdW0xFDASBgNVBAoTC0VjbGlwc2Ug
SW9UMQ8wDQYDVQQHEwZPdHRhd2ExCzAJBgNVBAYTAkNBMB4XDTIzMTAyNjA4MDgx
```

4.2.4 View Files in the Directory:

Use the `ls` command to list the files in the current directory.

```
$ ls
f   1538 /coap/certs/coap_ca.pem
$
```

Note: The `ls` command must be enabled in the configuration menu and included in the build as described above.