

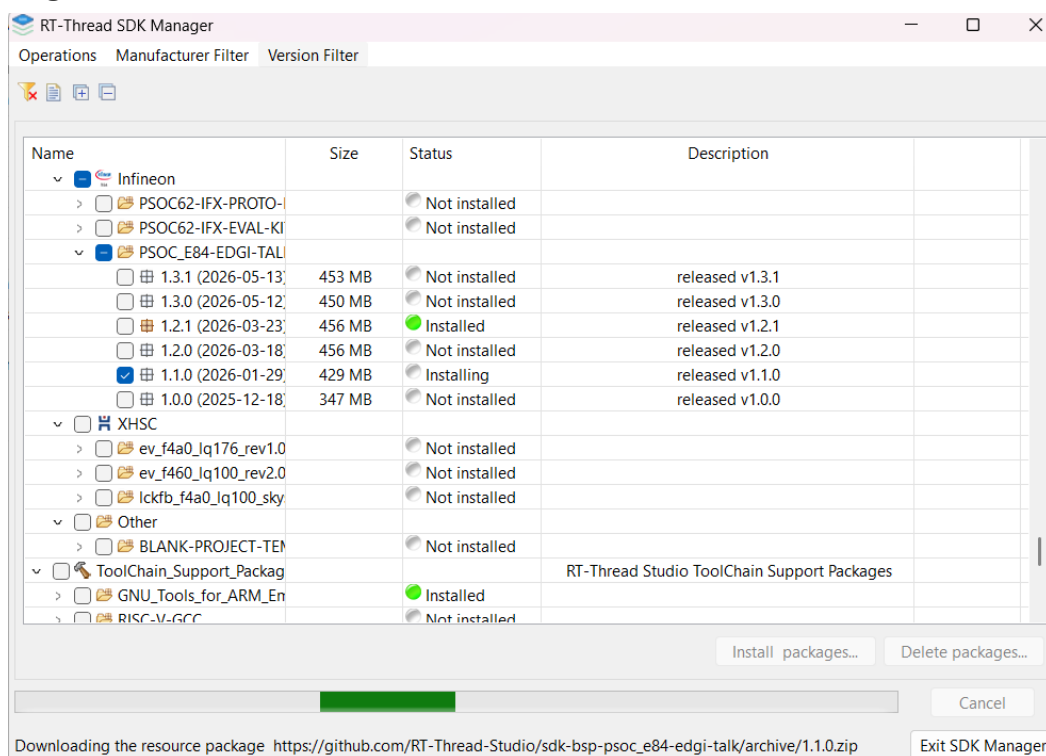
Module 5: LLM Web Interface Deployment

Learning Objectives

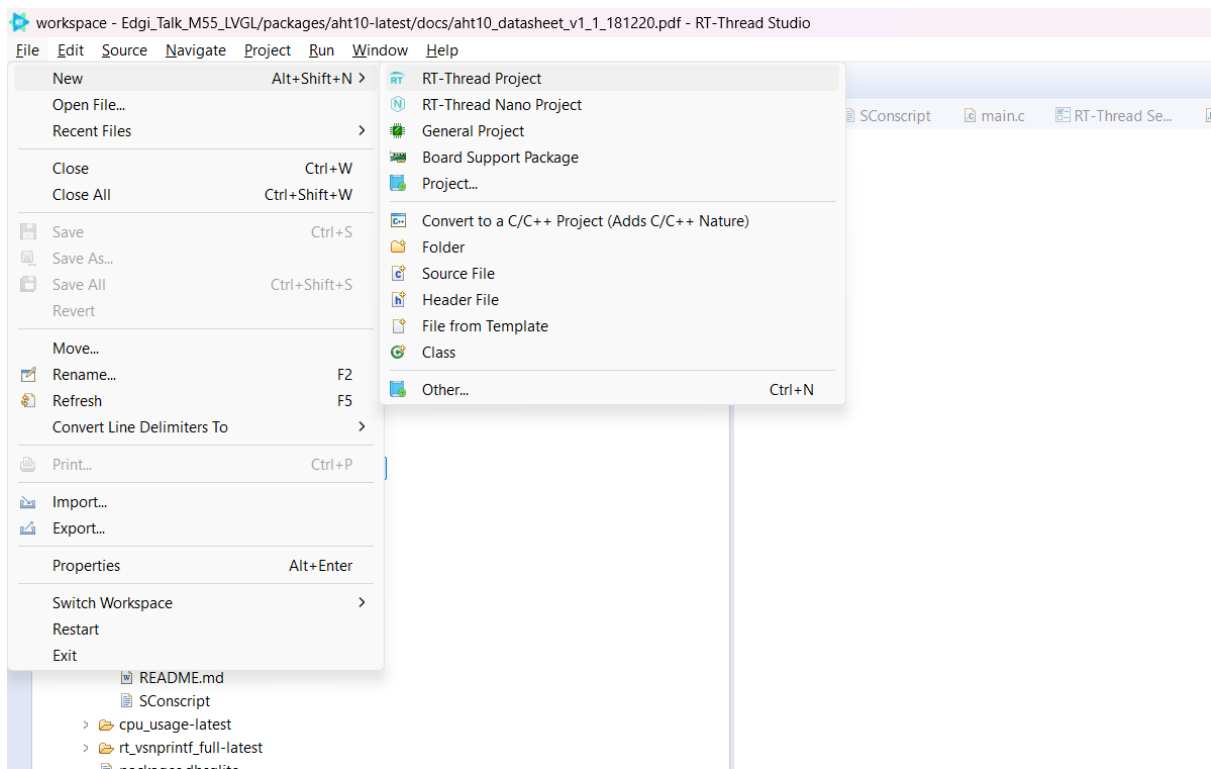
- Create WiFi-enabled project
- Configure llmchat package
- Register Alibaba Cloud / Qwen
- Configure network stack (mbedtls, webclient, webnet)
- Deploy web interface

5.1 WiFi Project Creation

- Open SDK Manager, In the dropdown, search for **PSOC_E84-EDGI-TALK**, install **Edgi-Talk 1.1.0**



- Create File, Project Name: Edgi_Talk_M55_WIFI
- Board: PSOC_E84-EDGI-TALK | BSP: 1.0.1.0
- Example: Edgi_Talk_M55_WIFI



Create RT-Thread Project

Input project name, choose RT-Thread version and select one Board and choose template or example.

Project name:

☒ Use default location

Location:

☐ Base On MCU ☒ Base On Board

Board :

BSP :

Type :

Example :

RT-Thread :

Adapter : Port :

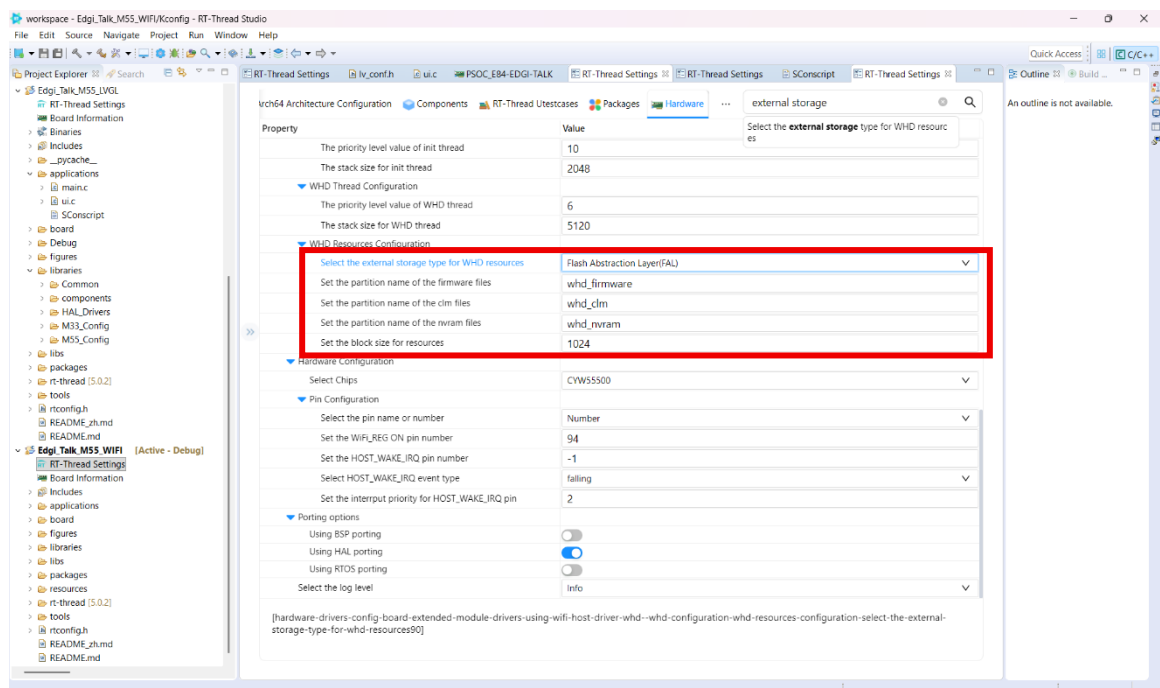
Because the WiFi project requires burning WiFi firmware and other assets, please refer to the following tutorial:

https://rt-thread-studio.github.io/sdk-bsp-psoc_e84-edgi-talk/latest/driver/Edgi_Talk_M55_WIFI/README_zh.html

Background: The WiFi chip needs 3 extra files (firmware, radio calibration table, module calibration parameters) that are separate from your compiled program — flashing your project does NOT include these, so you must upload them manually the first time.

Steps:

1. In **RT-Thread Settings (menuconfig)** for your WiFi project — confirm the option **WHD_RESOURCES_IN_EXTERNAL_STORAGE_FAL** is turned ON (should be default, just double-check).



2. **Connect XShell to the board's serial port (COM5)**, reset/reboot the board so you're at the `msh >` prompt.

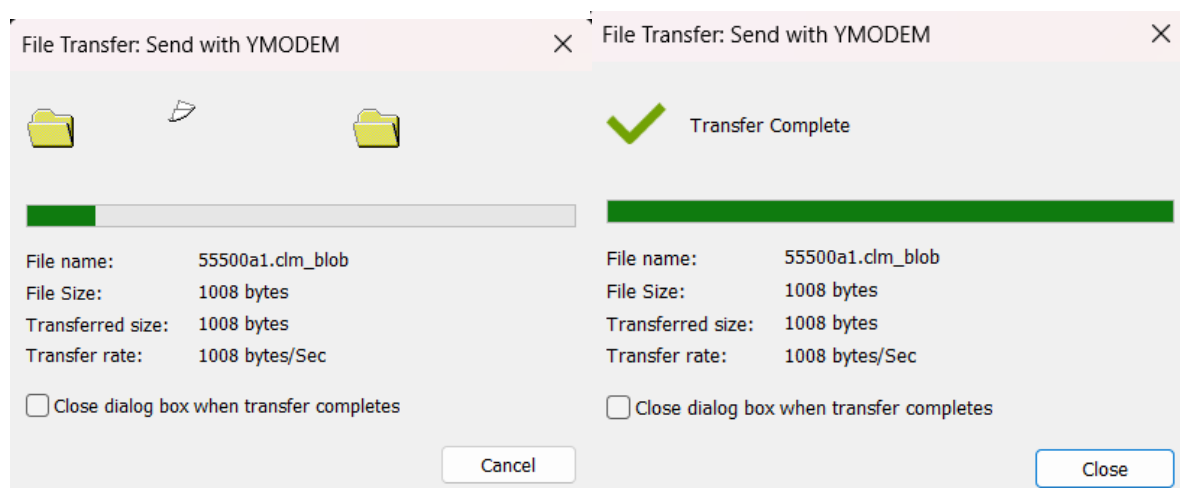
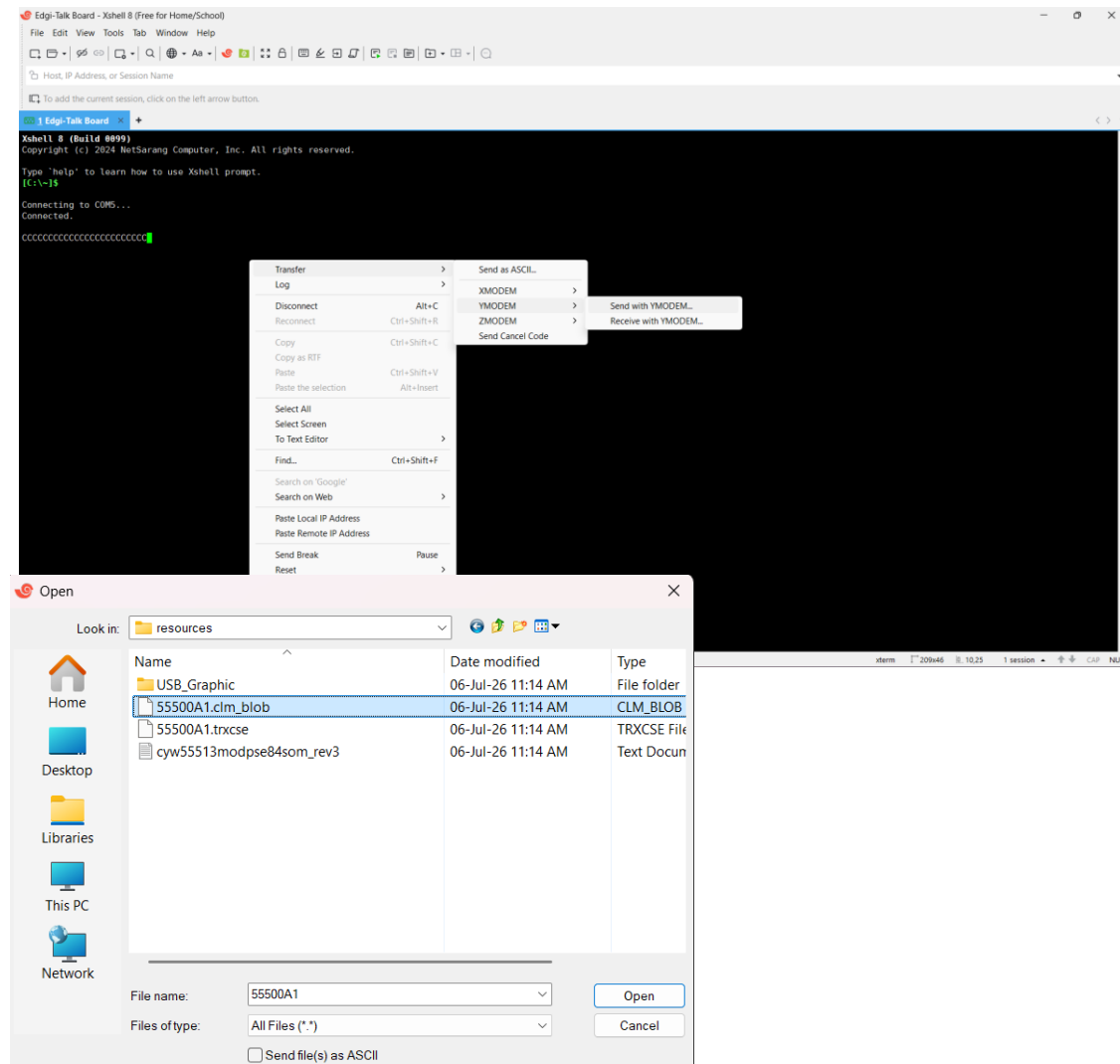
3. **Type this first command:**

```
whd_res_download whd_firmware
```

Press Enter. This puts the board into **YMODEM receive mode**, waiting for a file.

4. In **XShell**, trigger the **YMODEM file send**:

- Right-click in the terminal window → **Transfer** → **Send file (YMODEM)** → **Send with YMODEM...**
- Navigate to your project folder's **resources/** subfolder.
- Select the correct firmware file for your specific chip variant (there should be one matching your WiFi module).
- Send it — wait for confirmation.



5. **Wait for Download ... success** message in the terminal before moving to the next command.

6. **Repeat the same process** for the other two:

whd_res_download whd_clm

whd_res_download whd_nvram

(Each time: type the command → YMODEM send the matching file from resources/ → wait for success message.)

7. **Once all 3 are done, reboot the board** (reboot command) — WiFi will now use these newly loaded resources.

Command	File to send
whd_res_download whd_firmware	55500A1.trxcse
whd_res_download whd_clm	55500A1.clm_blob
whd_res_download whd_nvram	cyw55513modpse84som_rev3

```
***** PSOC Edge MCU: CM33 Secure Mode*****
PSRAM Cache is Enabled
PSRAM init successful
***** PSOC Edge MCU: CM33 Secure Mode Exit*****

\ | /
- RT -   Thread Operating System
/ | \   5.0.2 build Jul  6 2026 15:37:23
2006 - 2022 Copyright by RT-Thread team
lwIP-2.1.2 initialized!
[/sal.skt] Socket Abstraction Layer initialize success.
[/fal] RT-Thread Flash Abstraction Layer initialize success.
[/fal] The FAL MTD NOR device (filesystem) created successfully
[/app.filesystem] Block device created for filesystem
[/drv.sdio] cmd 8 arg 0x000001aa flags 0x000001f7 hw_err 0x0001
[156] RT-Thread WiFi Host Drivers (WHD)
[157] You can get the latest version on https://github.com/evlers/rt-thread_wifi-host-driver
[158] Chip supports bootloader handshake
[161] chip ID: 55500, chip rev: 1, Support ChipId Read from SDIO Core
[/app.filesystem] Filesystem mounted successfully
Hello RT-Thread
It's cortex-m55
msh />[991] WLAN FW download size: 233524 bytes
[991] WLAN FW download time: 824 ms
[1289] WLAN MAC Address : 9C:C7:D3:E1:BB:72
[1290] WLAN Firmware : wl0: Jul 25 2024 08:20:41 version 28.10.301 (aede64b) FWID 01-8cf45cc8
[1291] WLAN CLM : API: 20.0 Data: IFX.5551x Compiler: 1.49.5 ClmImport: 1.48.0 Customization: v2 24/06/28 Creation: 2024-07-02 03:05:22
[1303] WHD VERSION : 4.1.0.24264[1303] : llax v4.1.0[1303] : GCC 13.3[1303] : 2024-07-25 22:34:24 +0000
[1314] Disabled scanmac randomisation for 55500
[/WLAN.dev] wlan init success
[/WLAN.lwip] eth device init ok name:w0
[/WLAN.dev] wlan init success
[/WLAN.lwip] eth device init ok name:w1
```

After that, test WiFi with:

Command: wifi scan

```
[I/WLAN.dev] wlan init success
[I/WLAN.lwip] eth device init ok name:w1

msh />wifi scan
      SSID                MAC                security    rssi chn Mbps
-----
SepHealthBeauty_5GHz      32:f9:47:90:83:02  UNKNOWN    -82   48  300
LKL-5Ghz@unifi            f8:e5:a4:a5:50:b9  WPA2_MIXED_PSK -68  157  300
LKL-5Ghz@unifi            f8:e5:a4:82:7e:d9  WPA2_MIXED_PSK -58  157  300
Gateway_036463            c0:ba:1f:03:64:63  WPA2_MIXED_PSK -61   1   72
KCA-2.4Ghz@unifi          40:86:cb:73:d5:d2  UNKNOWN    -79   6  144
DIRECT-9E-EPS0N-680900    e2:bb:9e:68:89:00  WPA2_AES_PSK  -76  11   72
SepHealthBeauty_2.4GHz    32:f9:47:10:83:02  UNKNOWN    -58  11  300
LKL-2.4Ghz@unifi          f8:e5:a4:82:7e:d1  WPA2_MIXED_PSK -50   4  144
LKL-2.4Ghz@unifi          f8:e5:a4:a5:50:b1  WPA2_MIXED_PSK -62   4  300
[96928] Event handler not registered
TaiXiangChi_Guest         18:34:af:06:89:f3  WPA2_AES_PSK  -84   9  300
msh />
```

Command: wifi join <your_wifi_name> <your_wifi_password>

```
wifi: Command not found.
msh />wifi join LKL-2.4Ghz@unifi Lkl@13579
[I/WLAN.mgmt] wifi connect success ssid:LKL-2.4Ghz@unifi
msh />[I/WLAN.lwip] Got IP address : 192.168.101.138
```

Command: ping www.rt-thread.org

```
msh />ping www.rt-thread.org
ping: not found specified netif, using default netdev w0.
60 bytes from 155.102.32.51 icmp_seq=0 ttl=60 time=18 ms
60 bytes from 155.102.32.51 icmp_seq=1 ttl=60 time=11 ms
60 bytes from 155.102.32.51 icmp_seq=2 ttl=60 time=9 ms
60 bytes from 155.102.32.51 icmp_seq=3 ttl=60 time=10 ms
msh />
```

5.2 Add llmchat Package

Download this replacement package:

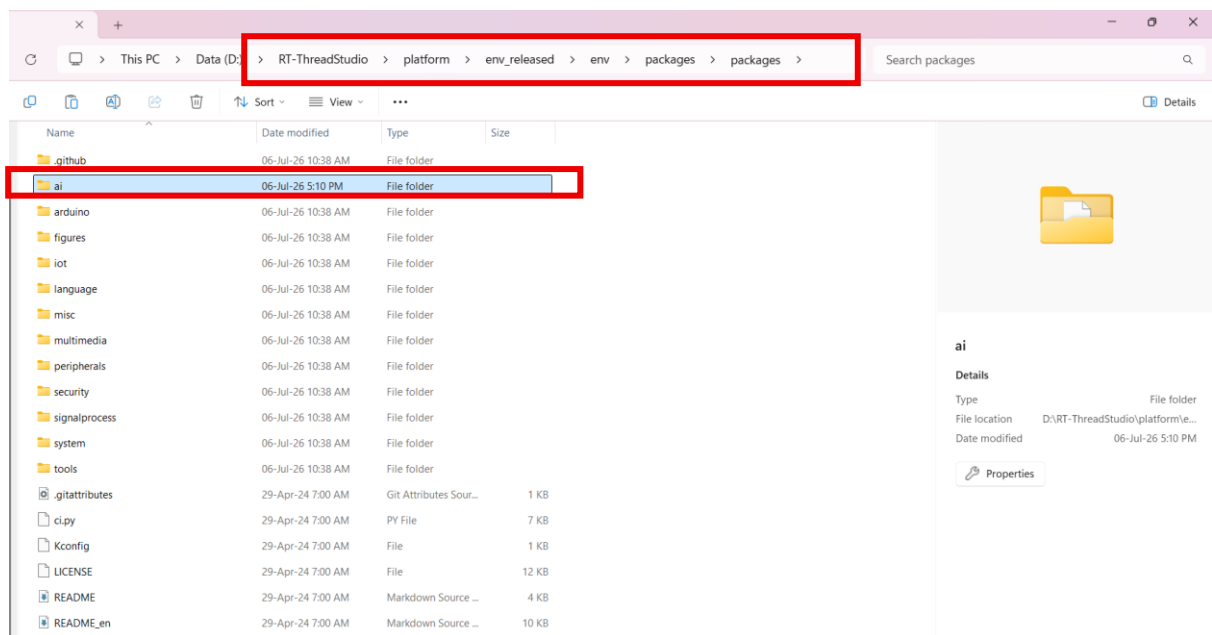
<https://github.com/RT-Thread/packages/archive/refs/heads/master.zip>

Steps:

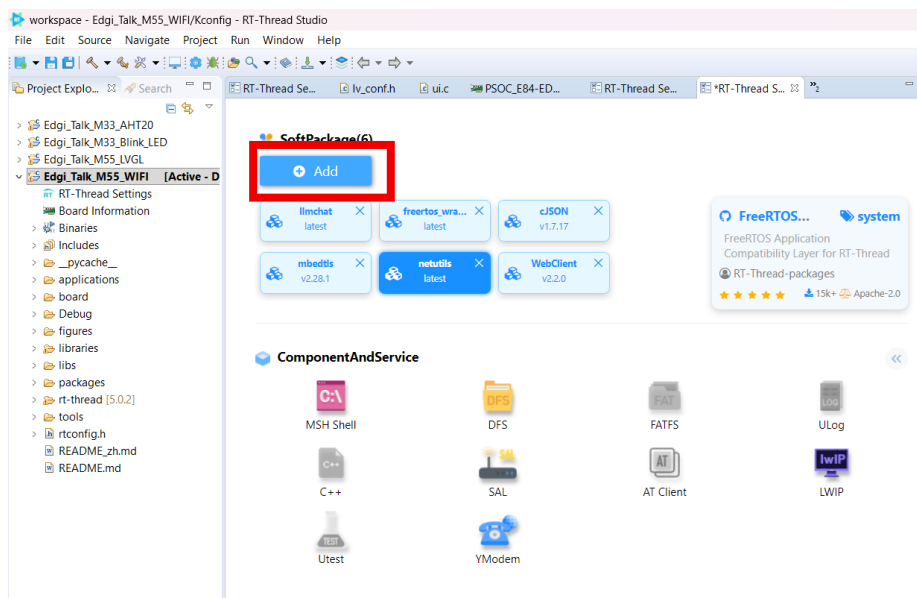
1. Extract the downloaded zip — you'll get a folder named **ai**.
2. Navigate to this location in Windows Explorer:

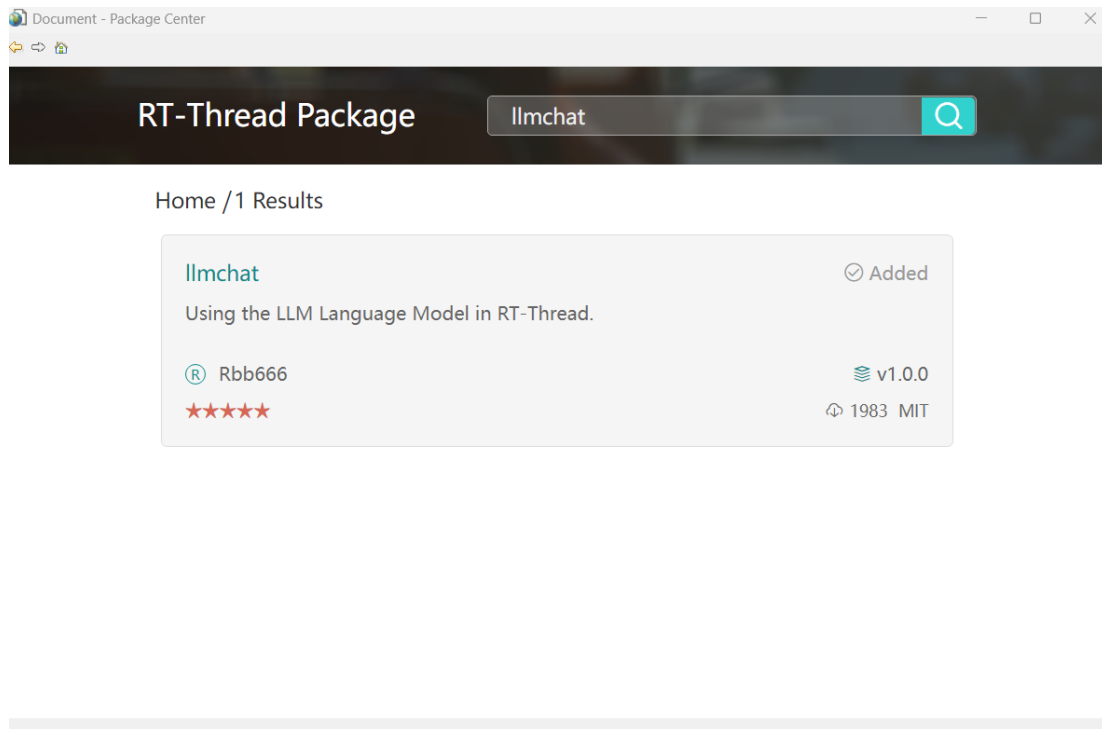
RT-ThreadStudio\platform\env_released\env\packages\packages

3. **Replace/overwrite** the existing ai folder there with the one you just extracted.

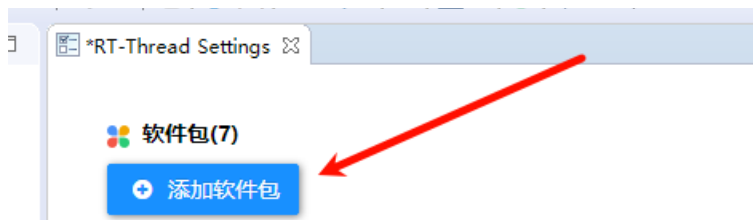
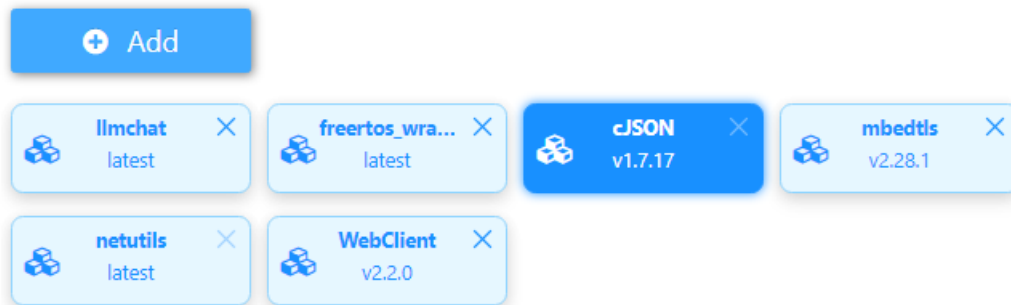


4. **Close RT-Thread Studio completely.**
5. **Reopen Studio**, open your Edgi_Talk_M55_WIFI project, go back to RT-Thread Settings.
6. Click the **Add Software Package** button, search for and add the llmchat package:





SoftPackage(6)



Once successfully added, it will look like this:

软件包(7)

添加软件包

llmchat
latest

Littlefs
v2.5.0

cJSON
v1.7.17

mbedtls
v2.28.1

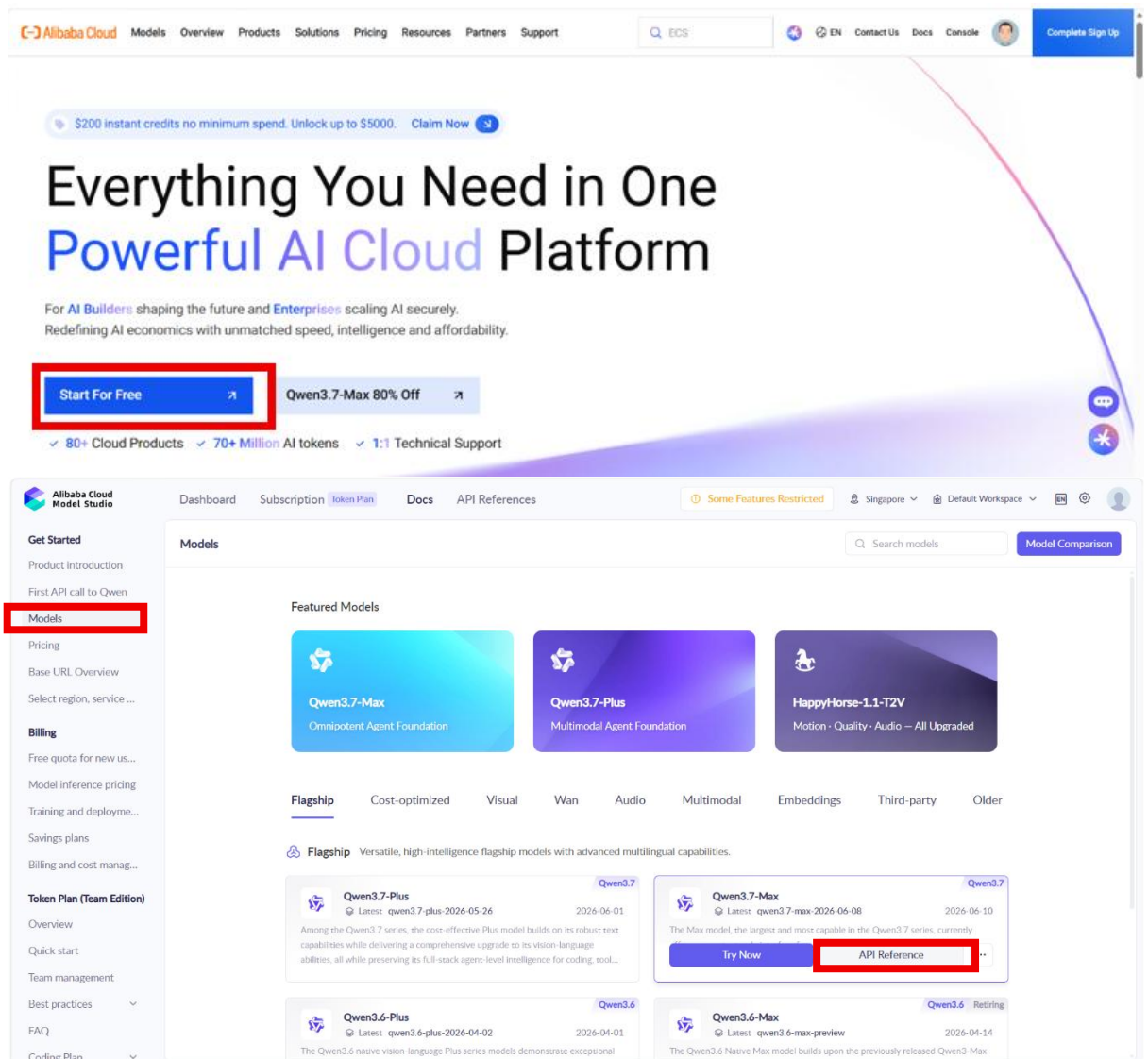
wifi_host_dr...
latest

WebNet
v2.0.3

WebClient
v2.2.0

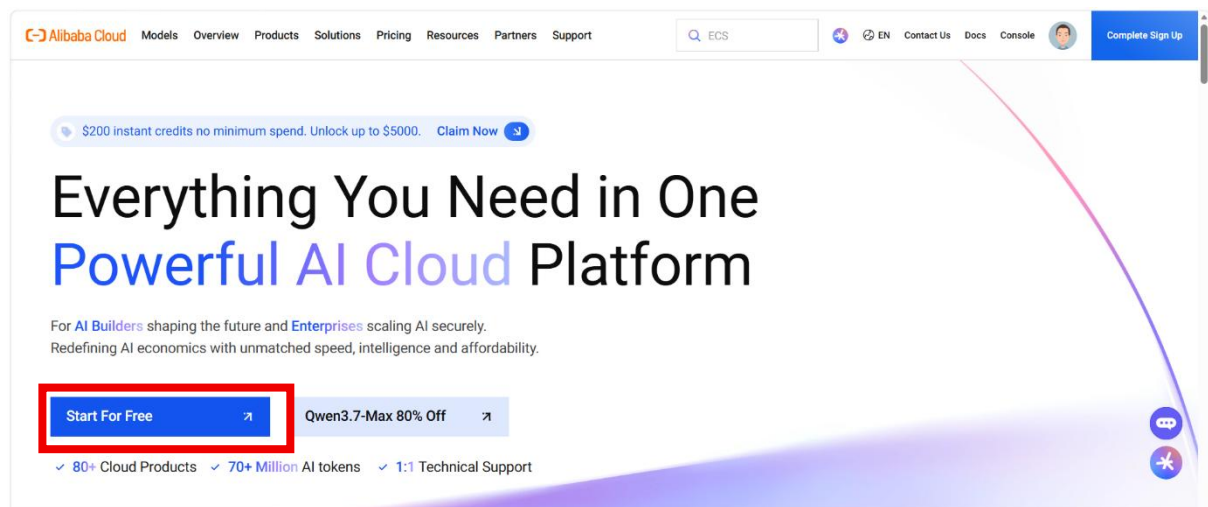
5.3 Qwen Account Setup

1. Register at Alibaba Cloud: <https://www.aliyun.com/product/bailian> and enter the **Model Studio (Bailian)** platform: Choose **Model Square** → **API Call Examples**.

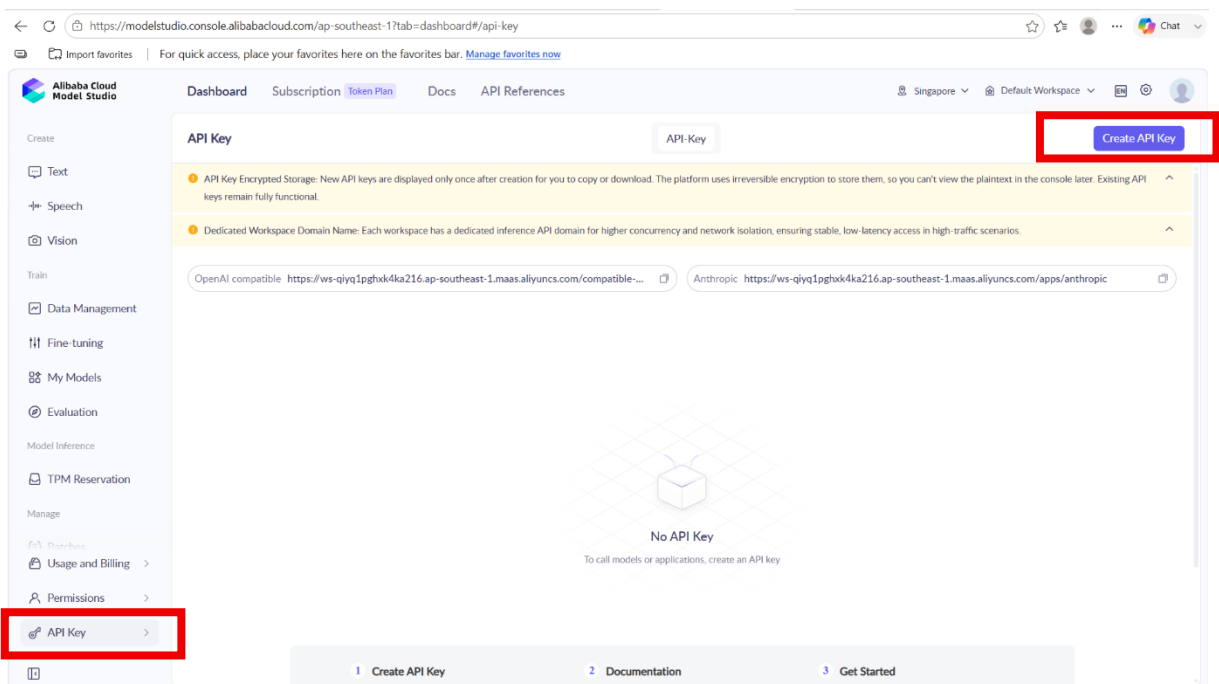


2. Top up minimum ¥1: Using the model requires topping up your account balance. Please top up as appropriate. For this demonstration, 1 RMB is enough to start using it: <https://billing-cost.console.aliyun.com/fortune/fund-management/recharge>

3. Next, enter the Qianwen Developer Console.



4. Create a new LLM API-KEY:



5. Fill in the description to create your API-KEY (SK-XXXXXXXX):

API-Key

Create API Key

Your account has no valid payment method. Bind one to avoid service disruption. [Bind Now](#)

Workspace *

Default Workspace

Description

API Model

9 / 200

Permissions *

All Custom

Cancel

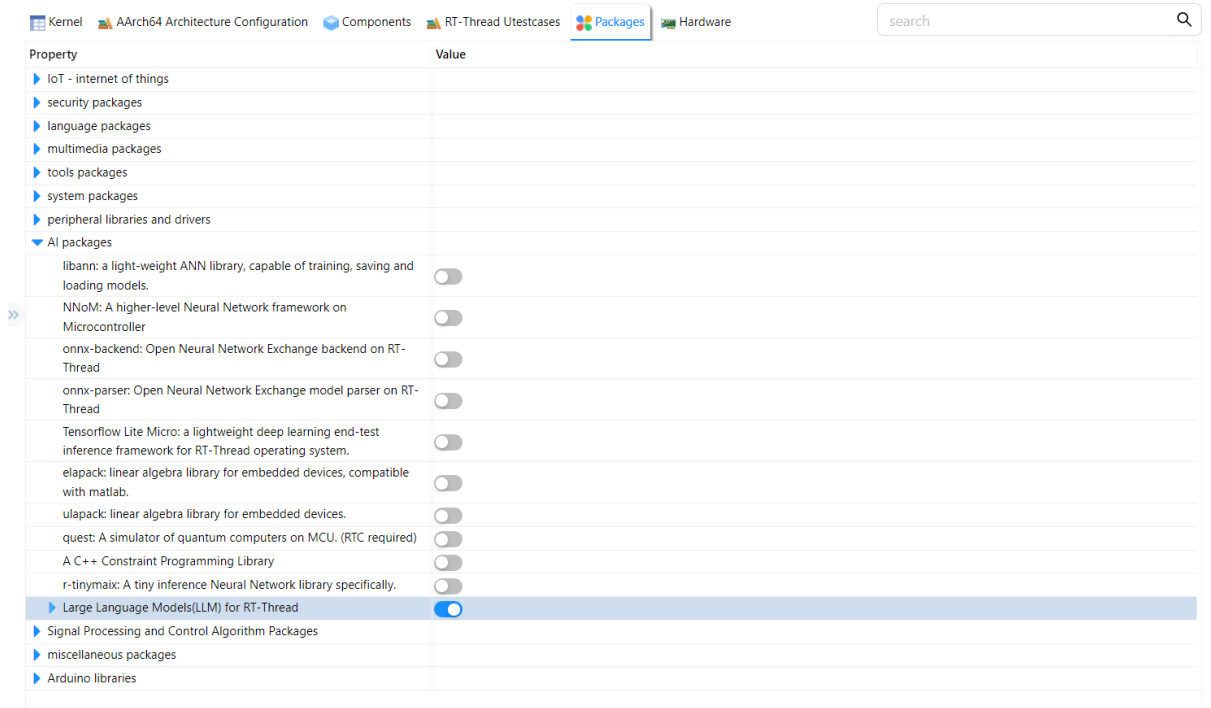
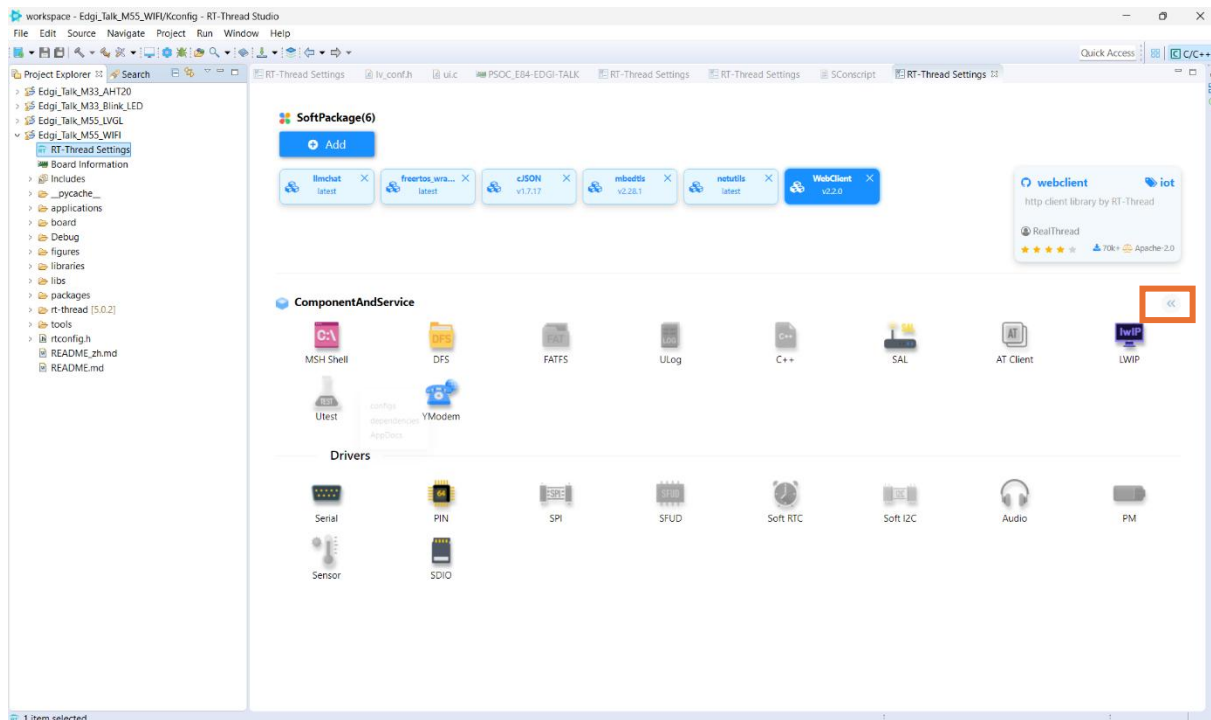
OK

我的API-KEY

ID	API KEY
1855420	sk-0****d839
1855420	sk-6****48c9

6. Open RT-Thread Settings and go to the following directory: Packages→ AI packages → Large Language Models(LLM) for RT-Thread; then configure it according to the image below:

- llm qwen user api key: your SK-xxx key
- llm qwen model name: qwen-plus



▼ Large Language Models(LLM) for RT-Thread	
select llm model	qwen llm port
llm qwen user api key	sk-ws-H.XDXPXM.DMIb.MEUCIE3I1wp5-6Ff0OQNOHcdCkSKNm
llm qwen model name	qwen-plus
llm qwen model api url	https://dashscope.aliyuncs.com/compatible-mode/v1/chat/completions

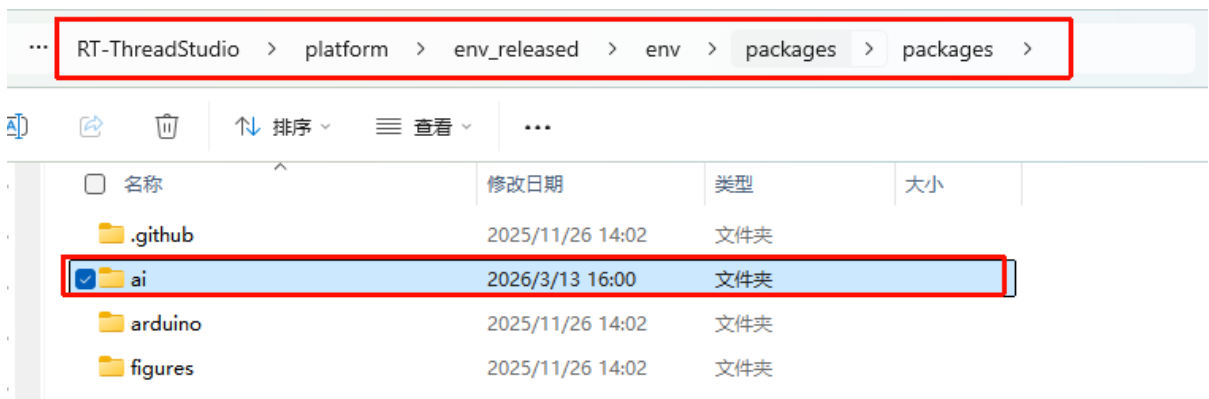
7. Modify the LLM configuration as shown below, and enable the llmchat webnet mode:

▼ Large Language Models(LLM) for RT-Thread	
select llm model	qwen llm port
llm qwen user api key	sk-ws-H.XDXPXM.DMIb.MEUCIE3I1wp5-6Ff0OQNOHcdCkSKNm
llm qwen model name	qwen-plus
llm qwen model api url	https://dashscope.aliyuncs.com/compatible-mode/v1/chat/completions
llm thread size	10240
llm cmd input buffer size	1024
websocket session buffer size	4096
Enable llmchat http stream	☒
Enable llmchat history payload	☐
Enable llmchat webnet mode	☒
Enable llmchat debug	☐
Enable llmchat example	☐
Version	latest

Note: If you run into issues where the LLM Chat package cannot be added, please solve it using the method below:

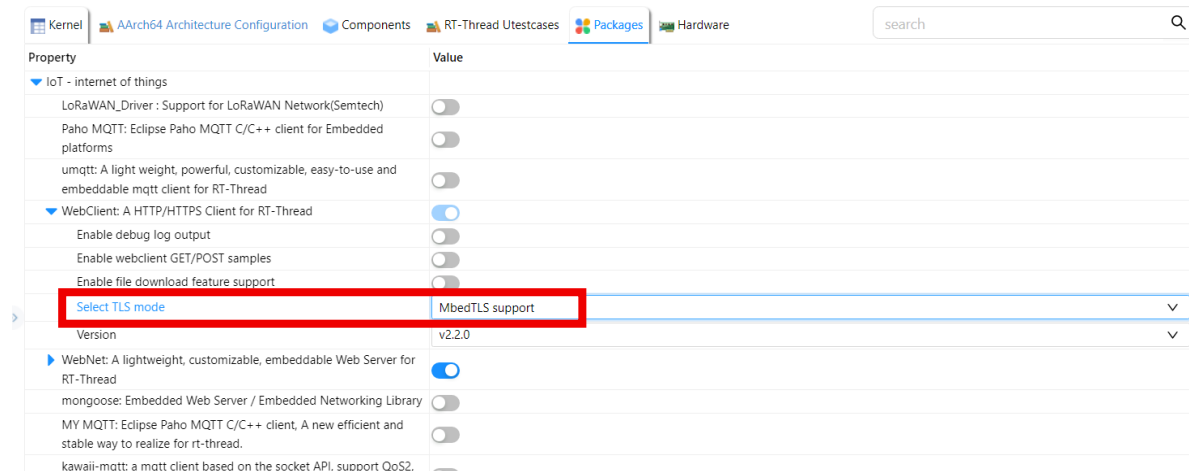
Download this zip package: <https://github.com/RT-Thread/packages/archive/refs/heads/master.zip>

Unzip and replace the ai folder under the Studio installation path shown below:

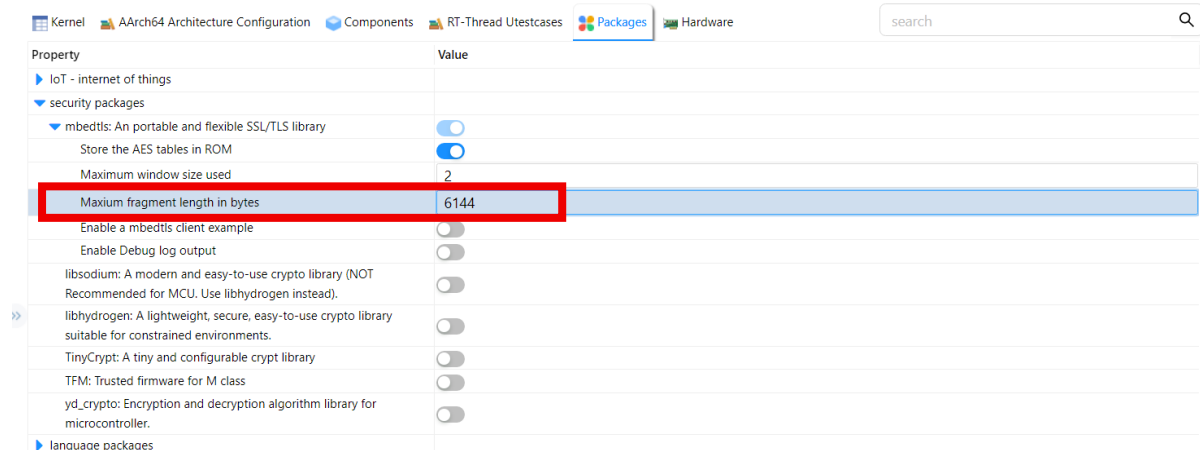


Save, restart Studio, then reopen **RT-Thread Settings** to successfully add the package.

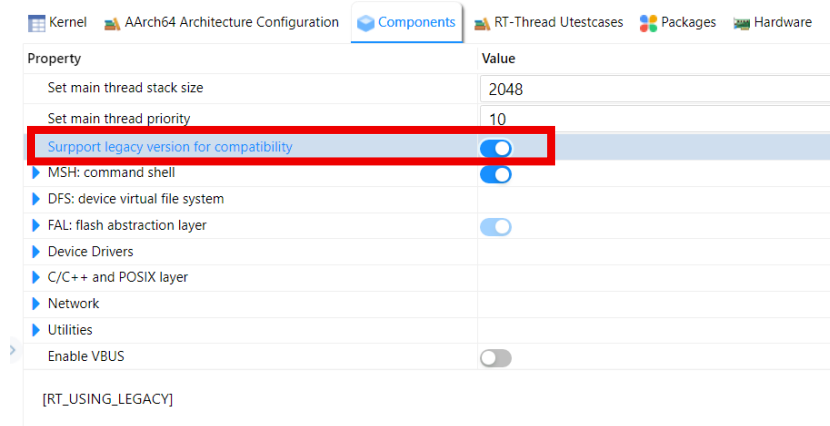
8. Enter Packages → IoT internet of things → WebClient: A HTTP/HTTPS Client for RT-Thread. Select MbedTLS support.



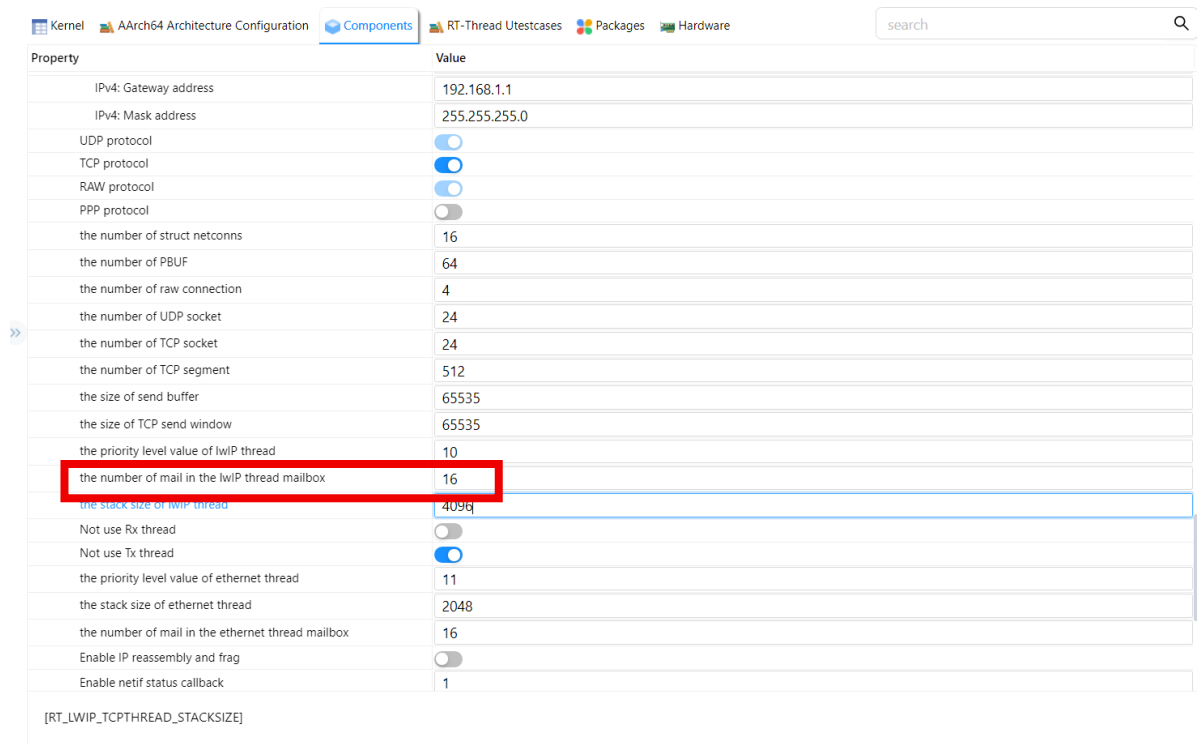
9. Enter RT-Thread online packagesPackages → security packages → mbedtls menu, change the **Maximum fragment length in bytes** field to **6144** (otherwise TLS handshake will fail).



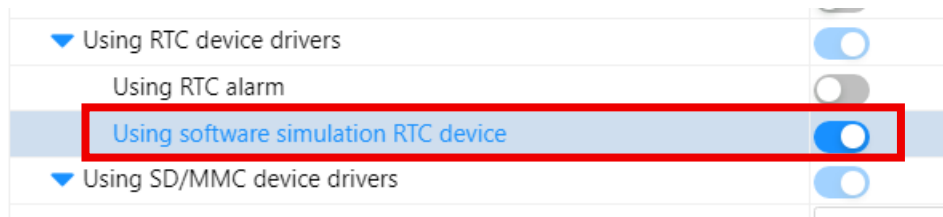
10. Enable old version compatibility support:



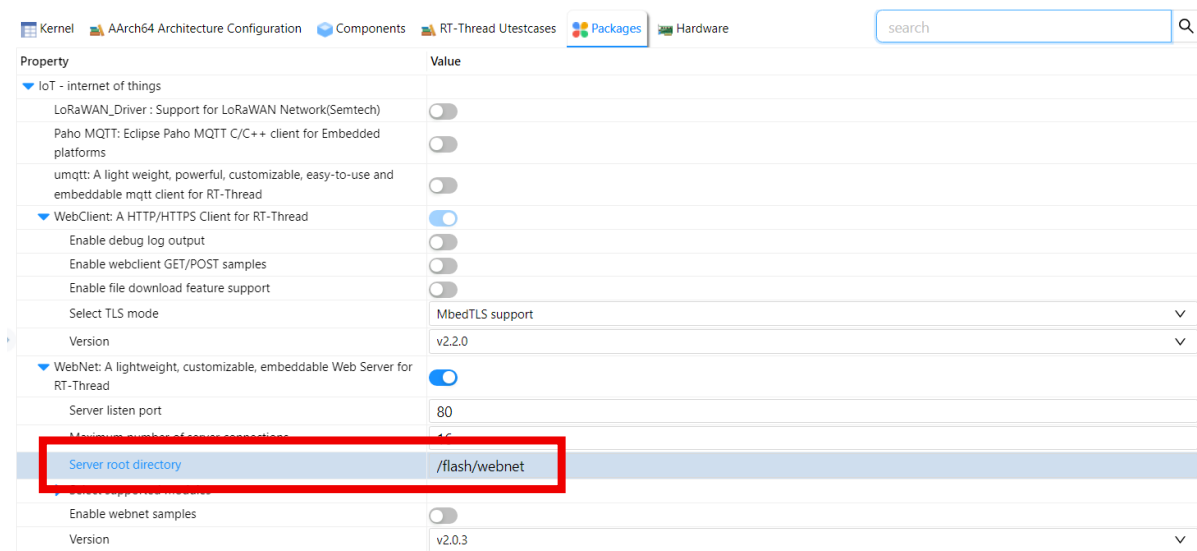
11. Modify the LWIP configuration: go to the following directory: Packages → Components → Network → Enable lwIP stack → Static IPv4 Address; then increasing the LWIP thread stack size to 4096:



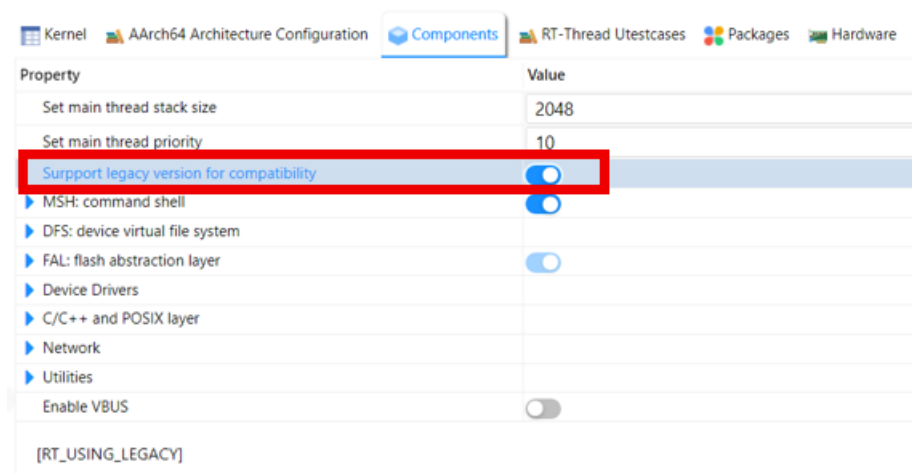
12. Enable the software-simulated RTC device under Components -> Device Drivers -> Using RTC Device Drivers -> Open Using software simulation RTC device:



13. Configure the webnet package(IoT – internet of things -> WebNet: A lightweight, customizable, embeddable Web Server for RT-Thread -> Server root directory) as shown below, changing the root path to /flash/webnet:



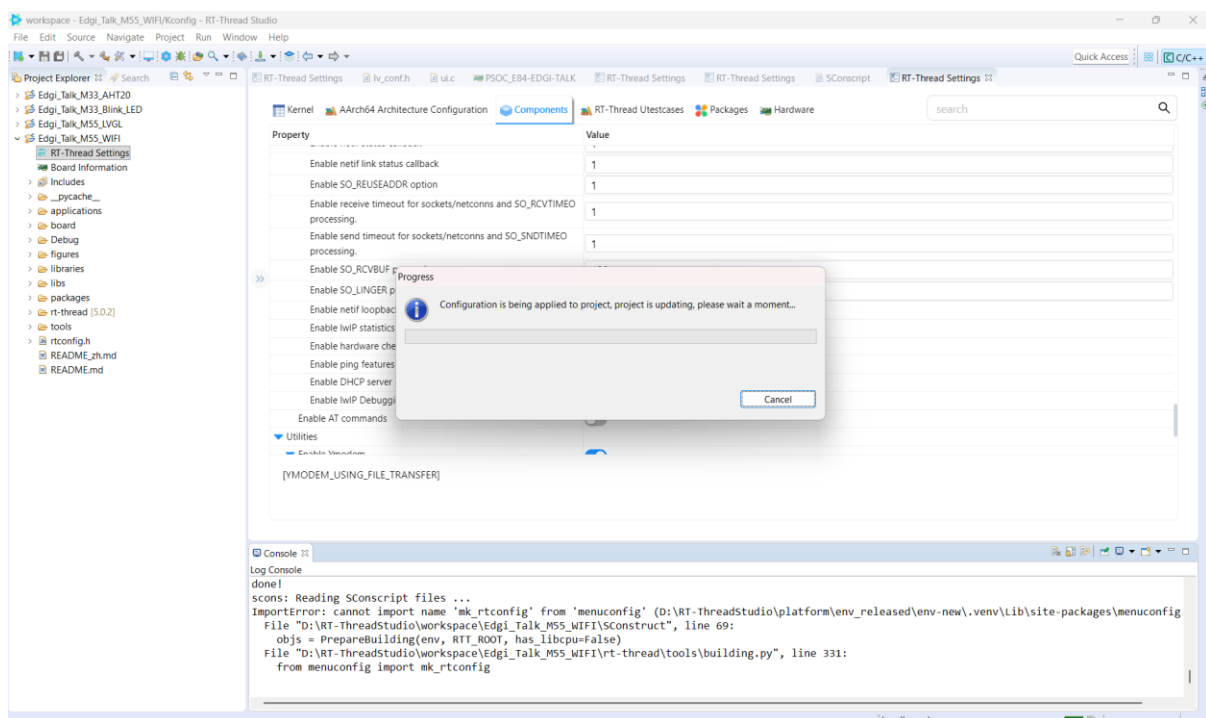
14. Enable backward compatibility support:



15. Enable the file transfer feature for ymodem(Uilities -> Enable Ymodem -> Enable file transfer feature):

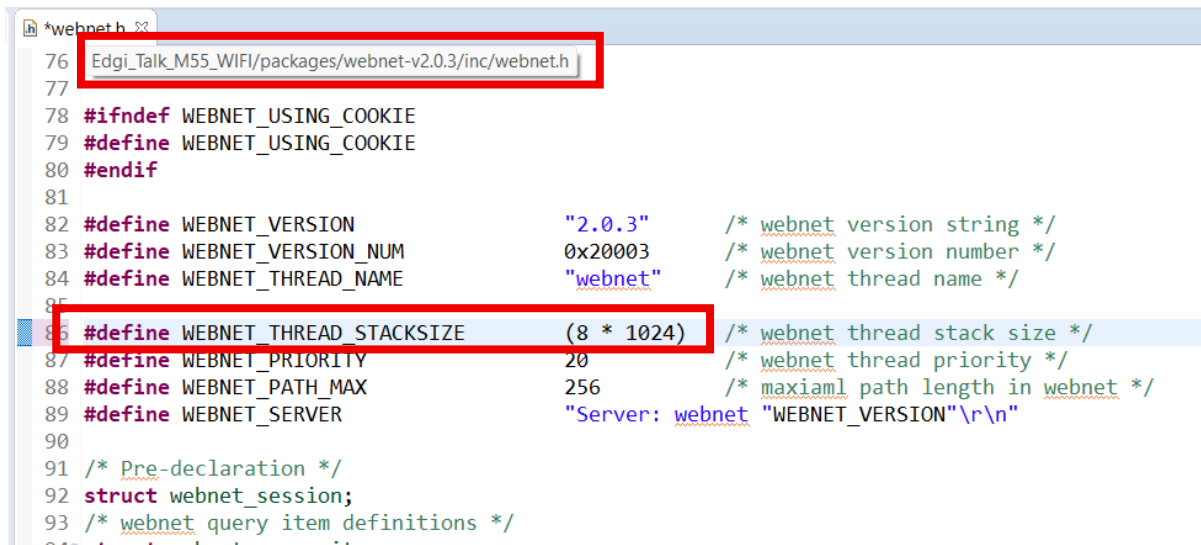
Utilities	
Enable Ymodem	☒
Enable CRC Table in Ymodem	☐
Enable file transfer feature	☒
Enable ulog	☐
Enable utest (RT-Thread test framework)	☐
Enable Var Export	☐
Enable resource id	☐
Enable ADT(abstract data type)	☐
RT-Link	☐
Enable VBUS	☐

16. Save (Ctrl + S) and exit the configuration. Studio will automatically pull the software packages:



17. Change the webnet thread stack size from (4*1024) to (8*1024):

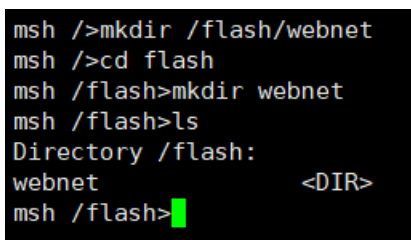
File: Edgi_Talk_M55_WIFI/packages/webnet-v2.0.3/inc/webnet.h



```
76 Edgi_Talk_M55_WIFI/packages/webnet-v2.0.3/inc/webnet.h
77
78 #ifndef WEBNET_USING_COOKIE
79 #define WEBNET_USING_COOKIE
80 #endif
81
82 #define WEBNET_VERSION "2.0.3" /* webnet version string */
83 #define WEBNET_VERSION_NUM 0x20003 /* webnet version number */
84 #define WEBNET_THREAD_NAME "webnet" /* webnet thread name */
85
86 #define WEBNET_THREAD_STACKSIZE (8 * 1024) /* webnet thread stack size */
87 #define WEBNET_PRIORITY 20 /* webnet thread priority */
88 #define WEBNET_PATH_MAX 256 /* maxiaml path length in webnet */
89 #define WEBNET_SERVER "Server: webnet" "WEBNET_VERSION"\r\n"
90
91 /* Pre-declaration */
92 struct webnet_session;
93 /* webnet query item definitions */
```

18. Compile the firmware.

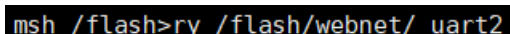
19. Open the board's serial terminal using XShell, and create the webnet folder: mkdir /flash/webnet



```
msh />mkdir /flash/webnet
msh />cd flash
msh /flash>mkdir webnet
msh /flash>ls
Directory /flash:
webnet          <DIR>
msh /flash>
```

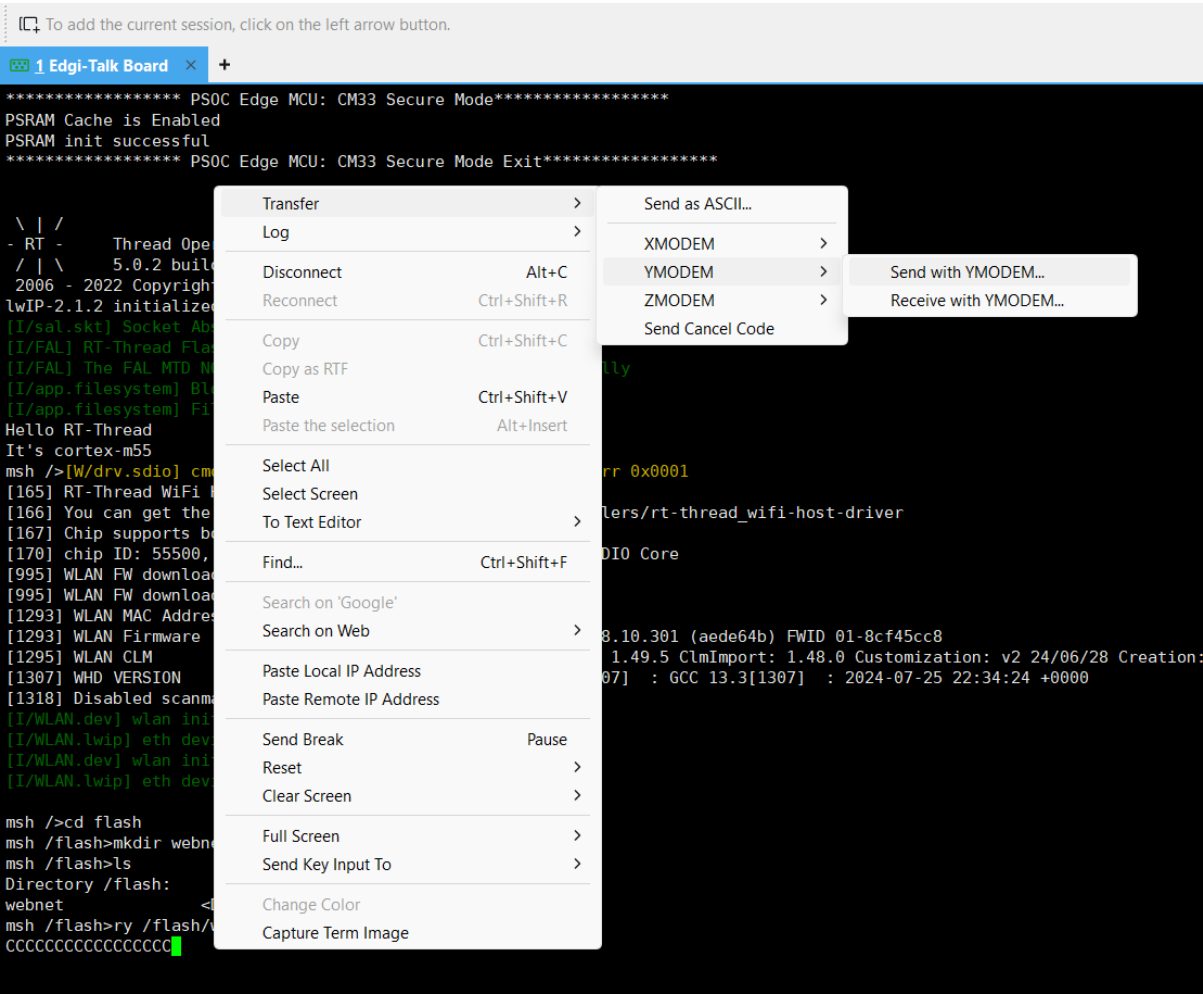
20. Transfer the html file to the development board using ymodem:

Command: ry /flash/webnet/ uart2



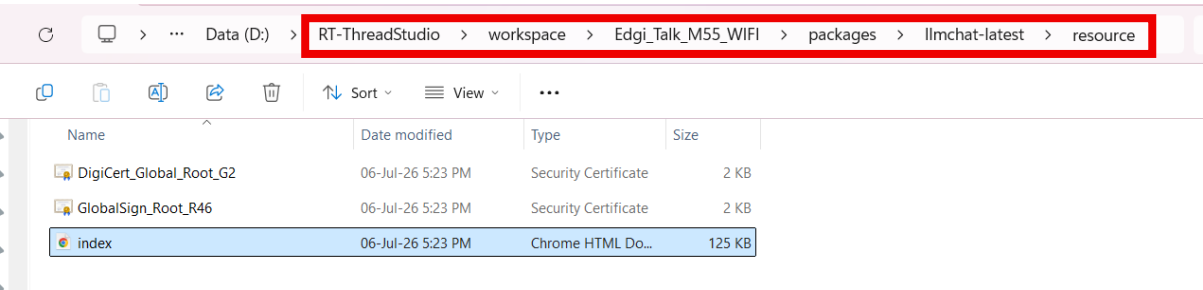
```
msh /flash>ry /flash/webnet/ uart2
```

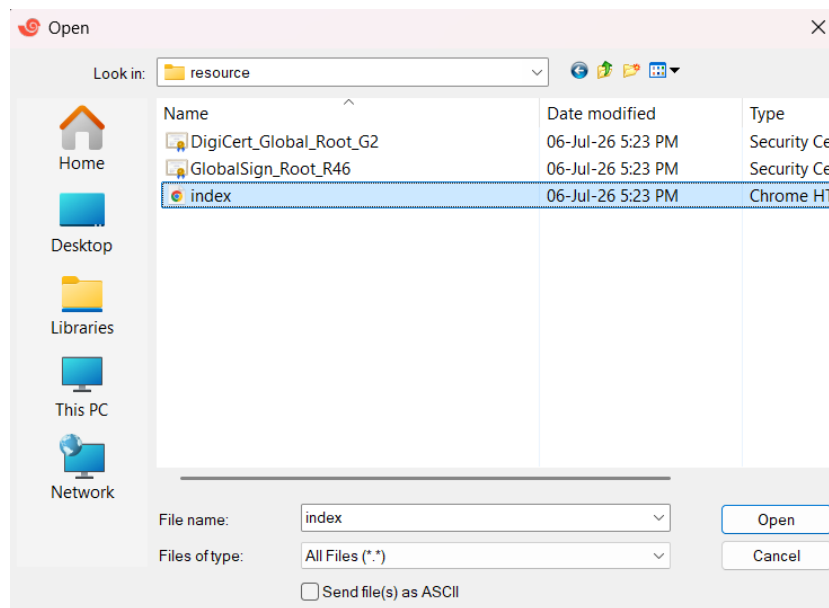
21.Select the asset file to transfer in XShell: Right click -> Transfer ->YMODEM -> Send with YMODEM...



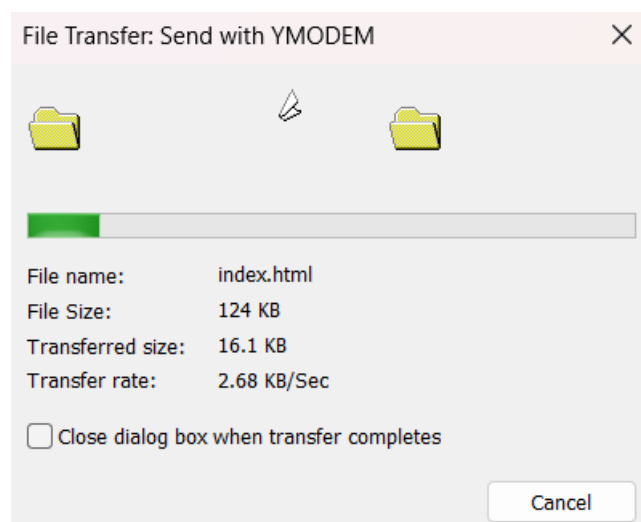
Select the following html file:

RT-ThreadStudio > workspace > Edgi_Talk_M55_WIFI> packages > llmchat-latest > resource > index.html





22. File transferring in progress:



Note :If the file cannot be transferred, try the file(index.html) D:\RT-ThreadStudio\workspace\Edgi_Talk_M55_WIFI\packages\llmchat-latest\resource\index.html

23. Next, verify if the file exists in the filesystem:

```
msh /flash/webnet>ls
Directory /flash/webnet:
index.html          127134
```

24. Connect to WiFi by entering: wifi join [wifi_AP_name] [wifi_AP_password], and wait for the successful acquisition of an IP address.

```
WiFi: Command not found.
msh />wifi join LKL-2.4Ghz@unifi Lkl@13579
[I/WLAN.mgmt] wifi connect success ssid:LKL-2.4Ghz@unifi
msh />[I/WLAN.lwip] Got IP address : 192.168.101.138
```

25. Run a quick test via the command line: Type llm to enter the chat terminal. Press CTRL+D to exit the chat session and return to the MSH prompt.

```
msh />wifi join LKL-2.4Ghz@unifi Lkl@13579
[I/WLAN.mgmt] wifi connect success ssid:LKL-2.4Ghz@unifi
msh />[I/WLAN.lwip] Got IP address : 192.168.101.138

msh />llm
msh />
Press CTRL+D to exit llm shell.
Enter command: hi
Hi there! 😊 How can I help you today?
Enter command: █
```

Note: If the issue: The certificate is not correctly signed by the trusted

CA occur,

```
msh />cd /
msh />wifi join LKL-2.4Ghz@unifi Lkl@13579
[I/WLAN.mgmt] wifi connect success ssid:LKL-2.4Ghz@unifi
msh />[I/WLAN.lwip] Got IP address : 192.168.101.138

msh />llm
msh />
Press CTRL+D to exit llm shell.
Enter command: 你好
[E/mbdttls.clnt] verify peer certificate fail....
[E/mbdttls.clnt] verification info: ! The certificate is not correctly signed by the trusted CA

[E/web] connect failed, https client connect return: -0xff
Webclient POST request failed, response status: -6
Enter command: .
Enter command: 你好
[E/mbdttls.clnt] verify peer certificate fail....
[E/mbdttls.clnt] verification info: ! The certificate is not correctly signed by the trusted CA

[E/web] connect failed, https client connect return: -0xff
Webclient POST request failed, response status: -6
Enter command: █
```

i. change the date to now in XShell,

command: date <year> <month> <day> <hour> <minute> <second>

(For example: date 2026 07 16 14 30 00)

```
msh />ntp_sync
ntp_sync: command not found.
msh />date
local time: Mon Jan 1 08:14:52 2018
timestamps: 1514765692
timezone: UTC+08:00:00
msh />█

serial://:22
```

ii. change the code in tls_client.c,

```
160 }
161
162 ret = mbedtls_ssl_config_defaults(&session->conf,
163                                 MBEDTLS_SSL_IS_CLIENT,
164                                 MBEDTLS_SSL_TRANSPORT_STREAM,
165                                 MBEDTLS_SSL_PRESET_DEFAULT);
166
167 if (ret != 0)
168 {
169     LOG_E("mbedtls_ssl_config_defaults error, return -0x%x", -ret);
170     return ret;
171 }
172
173 mbedtls_ssl_conf_authmode(&session->conf, MBEDTLS_SSL_VERIFY_NONE);
174 mbedtls_ssl_conf_ca_chain(&session->conf, &session->cacert, NULL);
175 mbedtls_ssl_conf_rng(&session->conf, mbedtls_ctr_drbg_random, &session->ctr_drbg);
176
177 mbedtls_ssl_conf_dbg(&session->conf, _ssl_debug, NULL);
178
179 ret = mbedtls_ssl_setup(&session->ssl, &session->conf);
180 if (ret != 0)
181 {
182     LOG_E("mbedtls_ssl_setup error, return -0x%x\n", -ret);
183     return ret;
184 }
185 LOG_D("mbedtls client context init success...");
```

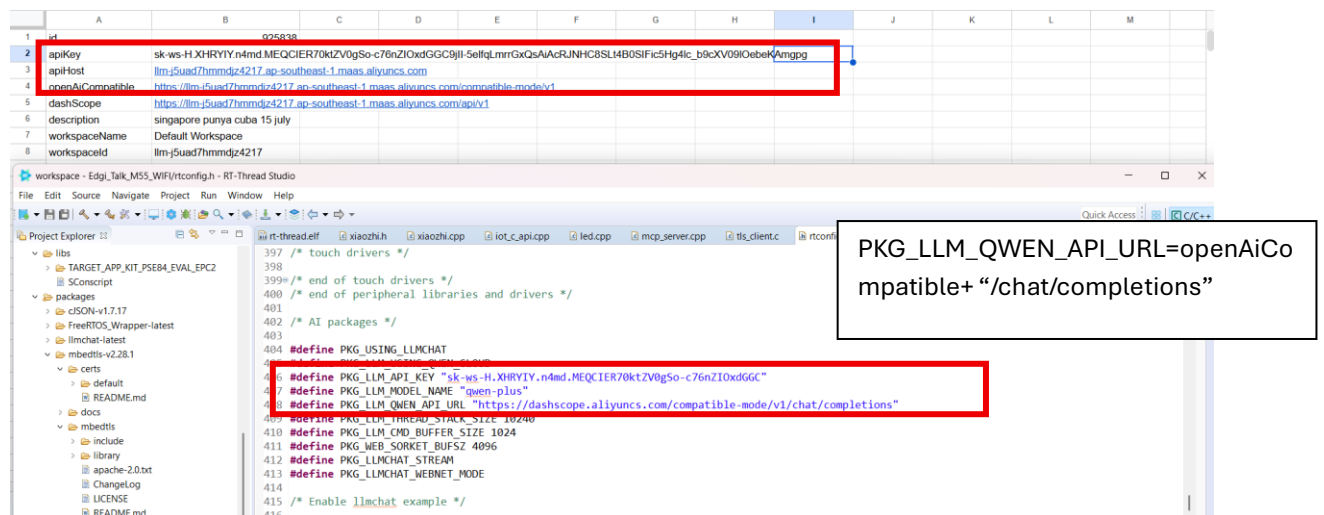
Change the code from
MBEDTLS_SSL_VERIFY_REQUIRED
-> MBEDTLS_SSL_VERIFY_NONE

Note: If the issue: Webclient POST request failed, response status: 402 occur

```
msh />wifi join LKL-2.4Ghz@unifi Lkl@13579
[I/WLAN.mgmt] wifi connect success ssid:LKL-2.4Ghz@unifi
msh />[I/WLAN.lwip] Got IP address : 192.168.101.138

msh />llm
msh />
Press CTRL+D to exit llm shell.
Enter command: 你好
Webclient POST request failed, response status: 402
Enter command: █
```

i. check whether the apiKey and openAiCompatible are exactly the same value with the coding in source code.

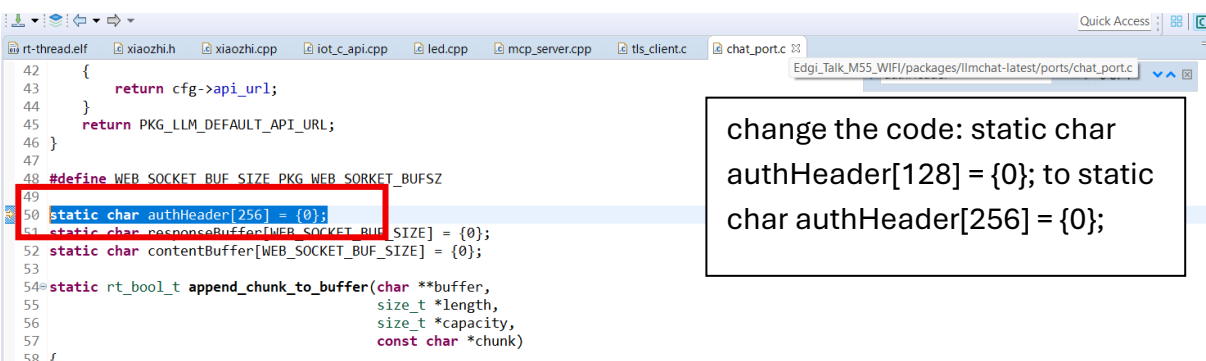


The screenshot shows a spreadsheet at the top with configuration values for apiKey, apiHost, openAiCompatible, dashScope, description, workspaceName, and workspaceId. Below it is a code editor showing the corresponding C code definitions in xiaozi.h. A red box highlights the apiKey and openAiCompatible values in both the spreadsheet and the code editor. A text box on the right states: "PKG_LLM_QWEN_API_URL=openAiCompatible+ "/chat/completions"".

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	id		025838										
2	apiKey		sk-ws-H.XHRYIY.n4md.MEQCIER70ktZV0gSo-c76nZIOxdGGC9jll-5eltqlmrrGxQsUAcRLNH8SLMB0SIFic5Hg4ic_b6cXV09iOebeKAmggg										
3	apiHost		llm-5uad7hmdjz4217.ap-southeast-1.maas.aliyuncs.com										
4	openAiCompatible		https://llm-5uad7hmdjz4217.ap-southeast-1.maas.aliyuncs.com/compatible-mode/v1										
5	dashScope		https://llm-5uad7hmdjz4217.ap-southeast-1.maas.aliyuncs.com/api/v1										
6	description		singapore punya cuba 15 july										
7	workspaceName		Default Workspace										
8	workspaceId		llm-5uad7hmdjz4217										

```
397 /* touch drivers */
398
399 /* end of touch drivers */
400 /* end of peripheral libraries and drivers */
401
402 /* AI packages */
403
404 #define PKG_USING_LLMCHAT
405 #define PKG_LLM_API_KEY "sk-ws-H.XHRYIY.n4md.MEQCIER70ktZV0gSo-c76nZIOxdGGC"
406 #define PKG_LLM_MODEL_NAME "qwen-plus"
407 #define PKG_LLM_QWEN_API_URL "https://dashscope.aliyuncs.com/compatible-mode/v1/chat/completions"
408 #define PKG_LLM_THREAD_STACK_SIZE 10240
409 #define PKG_LLM_CMD_BUFFER_SIZE 1024
410 #define PKG_WEB_SOCKET_BUF_SIZE 4096
411 #define PKG_LLMCHAT_STREAM
412 #define PKG_LLMCHAT_WEBNET_MODE
413 #define PKG_LLMCHAT_WEBNET_MODE
414
415 /* Enable llmchat example */
416
```

ii. in chat_port.c, modify the code



The screenshot shows the chat_port.c file in a code editor. A red box highlights the line: static char authHeader[256] = {0};. A text box on the right states: "change the code: static char authHeader[128] = {0}; to static char authHeader[256] = {0};".

```
42 {
43     return cfg->api_url;
44 }
45 return PKG_LLM_DEFAULT_API_URL;
46 }
47
48 #define WEB_SOCKET_BUF_SIZE PKG_WEB_SOCKET_BUF_SIZE
49 static char authHeader[256] = {0};
50 static char responseBuffer[WEB_SOCKET_BUF_SIZE] = {0};
51 static char contentBuffer[WEB_SOCKET_BUF_SIZE] = {0};
52
53 static rt_bool_t append_chunk_to_buffer(char **buffer,
54                                         size_t *length,
55                                         size_t *capacity,
56                                         const char *chunk)
57 {
58 }
```

The fix: simply increasing that buffer from 128 to 256 bytes gave enough room for the full header + full key to fit without truncation, so the board now sends the complete, correct key and authentication succeeds.

26. After connecting to the network successfully, type the command `webnet_llm_mode` on the development board:

```
msh />webnet_llm_mode
[I/wn] RT-Thread webnet package (V2.0.3) initialize success.
[I/llm_chat] === Current LLM Configuration ===
[I/llm_chat] API Key: sk-0c25918b01e3b4c1a3a5c1e570xc44d839
[I/llm_chat] Model Name: qwen-plus
[I/llm_chat] API URL: https://dashscope.aliyuncs.com/compatible-mode/v1/
[I/llm_chat] Is Configured: YES
[I/llm_chat] LLM Handle: CREATED
[I/llm_chat] =====
msh />
```

27. Open a web browser and input the IP address of the development board to view the webpage.

Note: Files transferred to /flash can sometimes get corrupted — ls will show a normal file size, but the content inside is broken (starts with junk `yyyy` bytes instead of real HTML), which causes the browser to show 404 Not Found even though the file "exists."