

# AP-RZA3-0A

## wolfSSL サンプルプログラム解説

第1.0版 2023年11月01日

<b>1. 概要</b>	<b>2</b>
1.1 概要	2
1.2 接続概要	2
1.3 本サンプルプログラムについて	3
1.4 開発環境について	3
1.5 ワークスペースについて	4
1.6 wolfSSL について	4
<b>2. サンプルプログラムの構成</b>	<b>5</b>
2.1 フォルダ構成	5
2.2 ファイルの構成	6
<b>3. プログラム作成方法</b>	<b>8</b>
3.1 プロジェクトの準備	8
3.2 プロジェクト設定	9
3.3 ソースファイルの変更	15
3.4 ソースファイルのビルド設定	16
3.4.1 除外ソースファイル一覧	16
3.4.2 除外手順	16
<b>4. 動作説明</b>	<b>18</b>
4.1 サンプルプログラムの動作	18
4.1.1 サンプルプログラム動作説明	18
4.1.2 サンプルプログラム サーバ動作詳細	18
4.1.3 ネットワーク設定	19
4.1.4 通信プロトコルの確認方法	22
4.2 e2 studio を用いたプロジェクトのビルド・デバッグ	25
<b>5. wolfSSL の入手方法</b>	<b>26</b>
5.1 wolfSSL の入手	26
5.2 ライブラリの作成、差し替え	27
5.3 PC アプリの作成	28
<b>6. 開発環境使用時の各設定値</b>	<b>29</b>
6.1 スイッチ設定	30

## 1. 概要

### 1.1 概要

本アプリケーションノートでは、弊社製 CPU ボード AP-RZA3-0A(RZ/A3UL CPU)を用いて、wolfSSL を使用したネットワーク通信を動作させるサンプルプログラムについて説明します。

サンプルプログラム	動作内容
AP-RZA3-0A wolfSSL サンプルプログラム	<ul style="list-style-type: none"><li>・ネットワーク通信</li><li>・セキュリティ (wolfSSL)</li></ul>

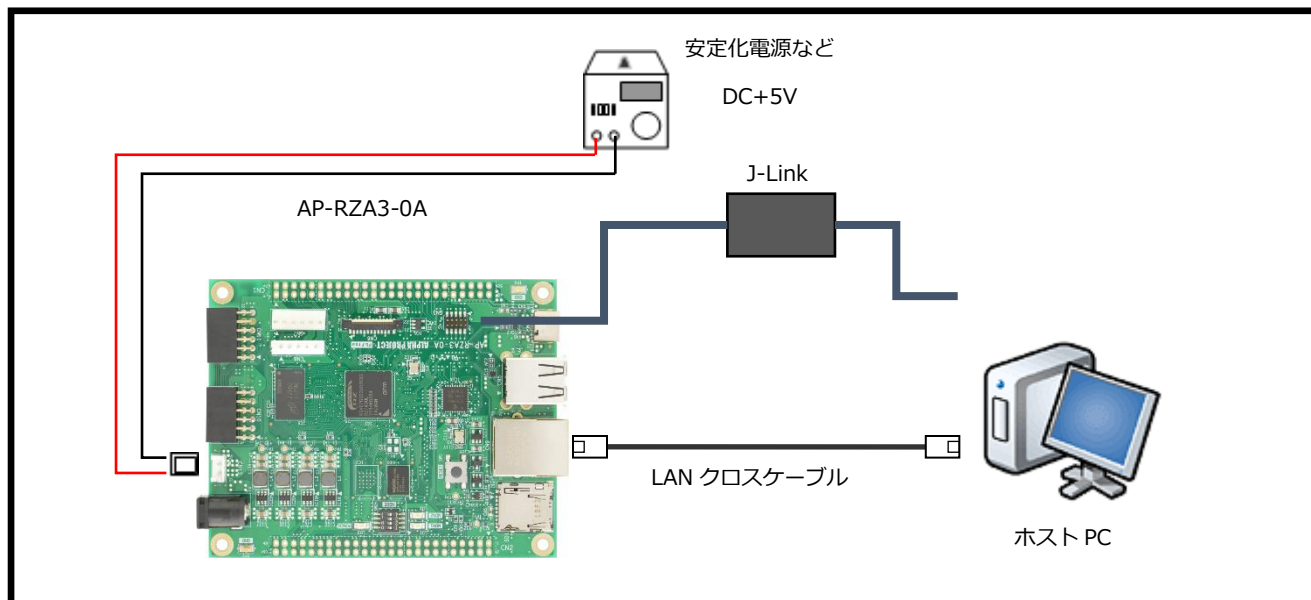
### 1.2 接続概要

「wolfSSL サンプルプログラム」の動作を確認する上で必要な CPU ボードとホスト PC 間の接続例を以下に示します。詳細な接続に関しては後述の「4. 動作説明」を参照してください。

※AP-RZA3-0A と J-Link を直接接続することはできません。

AP-RZA3-0A 側(ハーフピッチコネクタ)と J-Link 側(フルピッチコネクタ)を接続するための変換アダプタが必要となります。

変換アダプタについては、J-Link 取扱店へご確認ください。



### 1.3 本サンプルプログラムについて

本サンプルプログラムおよび本書含むアプリケーションノートは、弊社 Web サイトのボード紹介ページで公開されています。

株式会社アルファプロジェクト

AP-RZA3-0A 製品ページ <https://www.apnet.co.jp/product/rza/ap-rza3-0a.html>

### 1.4 開発環境について

本サンプルプログラムは、統合開発環境「e2 studio」と「Flexible Software Package（以下、FSP）」を用いて開発されています。

本サンプルプログラムに対応する開発環境、コンパイラのバージョンは次の通りです。

ソフトウェア	バージョン	備考
e2studio	2023-01	-
GCC	10.3.1.20210621	A-Profile AArch64 bare-metal
FSP	1.2.1	Flexible Support Package for Renesas RZ/A シリーズ
Visual Studio 2019 Professional	16.7.4	ツールセット Visual Studio 2019 (v142) ※動作確認用 PC アプリ作成に使用

デバッガ	ハードウェアバージョン	備考
J-Link	V11	Segger Microcontroller Systems 社 ハードウェアバージョン V10 以下はご使用になれませんのでご注意ください。

## 1.5 ワークスペースについて

本サンプルプログラムのプロジェクトファイルは次のフォルダに格納されています。

サンプルプログラム	フォルダ
wolfSSL サンプルプログラム プロジェクトフォルダ	¥sample¥ap_rza3_0a_wolfssl_sample

## 1.6 wolfSSL について

wolfSSL とは、wolfSSL 社製のマイコン組込みシステム向けの軽量 SSL/TLS ライブラリです。

最新プロトコル標準、暗号アルゴリズムにも対応し、世界中の組込み製品で利用されています。

wolfSSL の製品は、オープンソース版と商用版の 2 種類があります。デバイスや商用ソフトウェアに wolfSSL 製品の採用を希望される場合、商用版のご契約が必要です。

詳細や使用条件などにつきましては、wolfSSL 社 Web サイトをご覧ください。

wolfSSL 日本語サイト : <https://www.wolfssl.jp/>

本サンプルプログラムでは、「オープンソース版 v5.6.0」を使用しています。

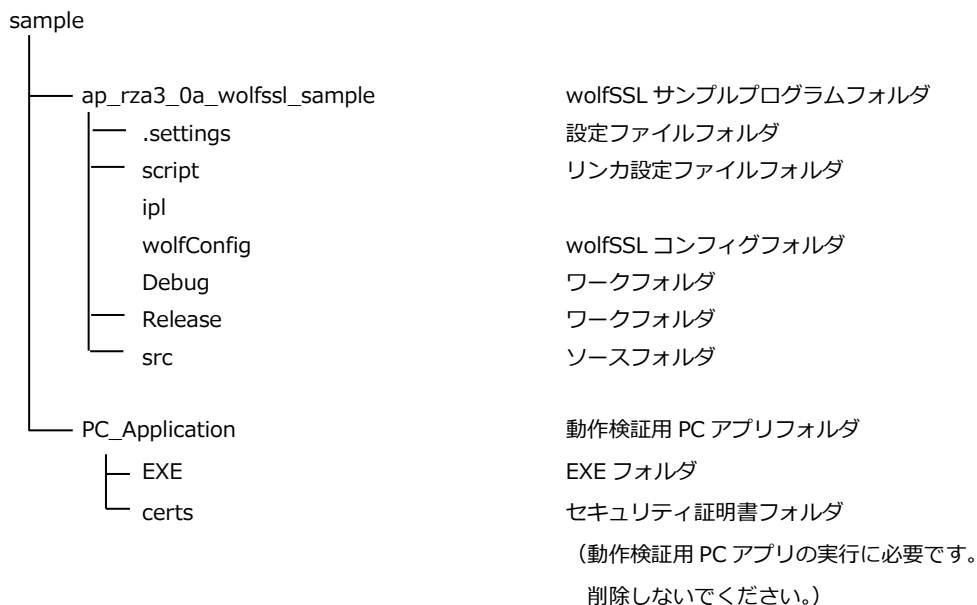
オープンソース版の使用条件につきましては、wolfSSL 社 Web サイト 組込み SSL ライブラリ オープンソース版ダウンロードページにて、「使用許諾契約」をご確認ください。

wolfSSL 社 組込み SSL ライブラリページ : <https://www.wolfssl.jp/products/wolfssl/>

## 2. サンプルプログラムの構成

### 2.1 フォルダ構成

サンプルプログラムは下記のようなフォルダ構成になっています。



## 2.2 ファイルの構成

本サンプルプログラムは以下のファイルで構成されています。

本章では、ミドルウェア・ドライバ等の既存のファイルに関しては説明を省略してあります。

### <¥sample¥CustomBsp フォルダ内>

AlphaProject.ap\_rza3\_0a.1.2.1.pack ... AP-RZA3-0A パックファイル

### <¥sample ¥PC\_Application フォルダ内>

client.bat ... client.exe 起動用バッチファイル

### <¥sample ¥PC\_Application¥EXE フォルダ内>

client.exe ... クライアント動作 PC アプリケーション

### <¥sample¥ap\_rza3\_0a\_wolfssl\_sample フォルダ内>

.cproject ... CPROJECT ファイル  
.project ... PROJECT ファイル  
configuration.xml ... FSP コンフィギュレータファイル  
ap\_rza3\_0a.pincfg ... AP-RZA3-0A ピンコンフィグファイル  
ap\_rza3\_0a\_wolfssl\_sample ... AP-RZA3-0A Ethernet(wolfSSL)サンプルプログラム  
Debug.launch ... デバッグおよびランタイム設定ファイル (Debug ビルド用)  
ap\_rza3\_0a\_wolfssl\_sample ... AP-RZA3-0A Ethernet(wolfSSL)サンプルプログラム  
Debug.jlink ... J-Link 設定ファイル (Debug ビルド用)  
ap\_rza3\_0a\_wolfssl\_sample ... AP-RZA3-0A Ethernet(wolfSSL)サンプルプログラム  
Release.launch ... デバッグおよびランタイム設定ファイル (Release ビルド用)  
ap\_rza3\_0a\_wolfssl\_sample ... AP-RZA3-0A Ethernet(wolfSSL)サンプルプログラム  
Release.jlink ... J-Link 設定ファイル (Release ビルド用)

### <¥sample¥ap\_rza3\_0a\_wolfssl\_sample¥script フォルダ内>

postbuild.sh ... バイナリファイル作成用スクリプトファイル  
rza3ul\_smarc\_qspi.ld ... リンカスクリプトファイル

<¥sample¥ap\_rza3\_0a\_wolfssl\_sample¥src フォルダ内>

i2c	...	i2c ソースファイルフォルダ
wolfssl	...	wolfSSL ソースファイルフォルダ
common_app.h	...	サンプルプログラム用定義ヘッダファイル
hal_entry.c	...	アプリケーションソースファイル
mmu_page_table.c	...	MMU ページテーブル定義ソースファイル
net_thread_entry.c	...	ネットワーク通信ソースファイル
sections.c	...	セクション定義ソースファイル
tcp_echo_server_thread_entry.c	...	wolfSSL サーバ動作 TCP サーバ (エコーバック) 通信ソースファイル
udp_echo_server_thread_entry.c	...	UDP (エコーバック) 通信ソースファイル

<¥sample¥ap\_rza3\_0a\_wolfssl\_sample¥src¥i2c フォルダ内>

eeeprom.c	...	EEPROM 通信用ソースファイル
eeeprom.h	...	EEPROM 通信用ヘッダファイル
i2c.c	...	I2C 通信用ソースファイル
i2c.h	...	I2C 通信用ヘッダファイル

### 3. プログラム作成方法

本章では、AP-RZA3-0A の Ethernet のサンプルプログラムに wolfSSL を追加する方法を説明します。  
なお、本サンプルプログラムを動作させる場合には、本章の手順は必要ありません。  
動作方法に関しては、「4. 動作説明」をご覧ください。

#### 3.1 プロジェクトの準備

AP-RZA3-0A のサンプルプログラムに wolfSSL を追加するために、以下のプログラムを準備してください。

- ① AP-RZA3-0A サンプルプログラムの、Ethernet サンプルプログラム  
(弊社 Web サイト AP-RZA3-0A 製品ページよりダウンロード可能なサンプルプログラム  
『AP-RZA3-0A サンプルプログラム』に含まれています。)
  - ・ ap\_rza3\_0a\_ethernet\_sample
- ② 本サンプルプログラム
  - ・ ap\_rza3\_0a\_wolfssl\_sample

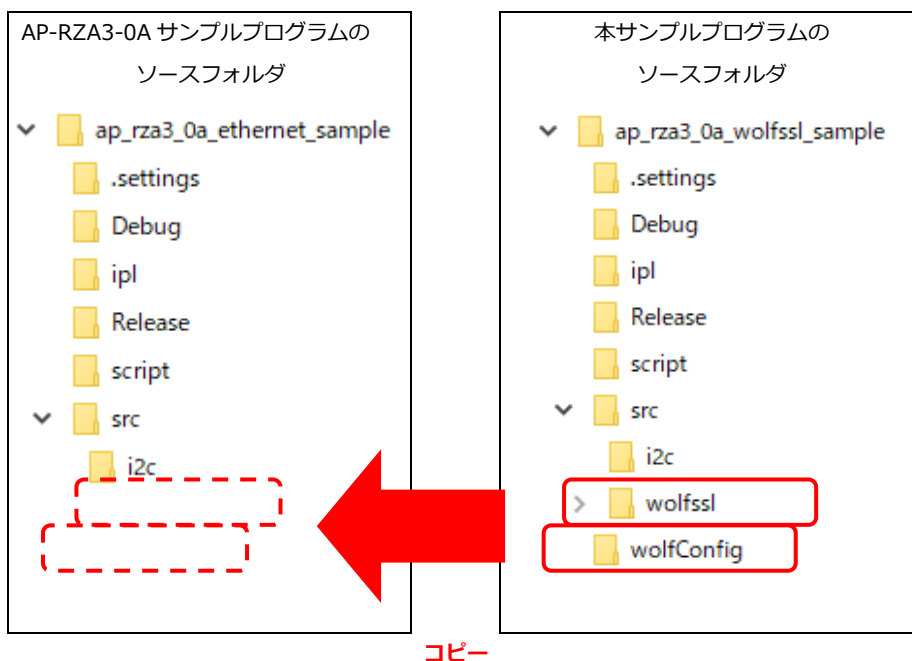


## 3.2 プロジェクト設定

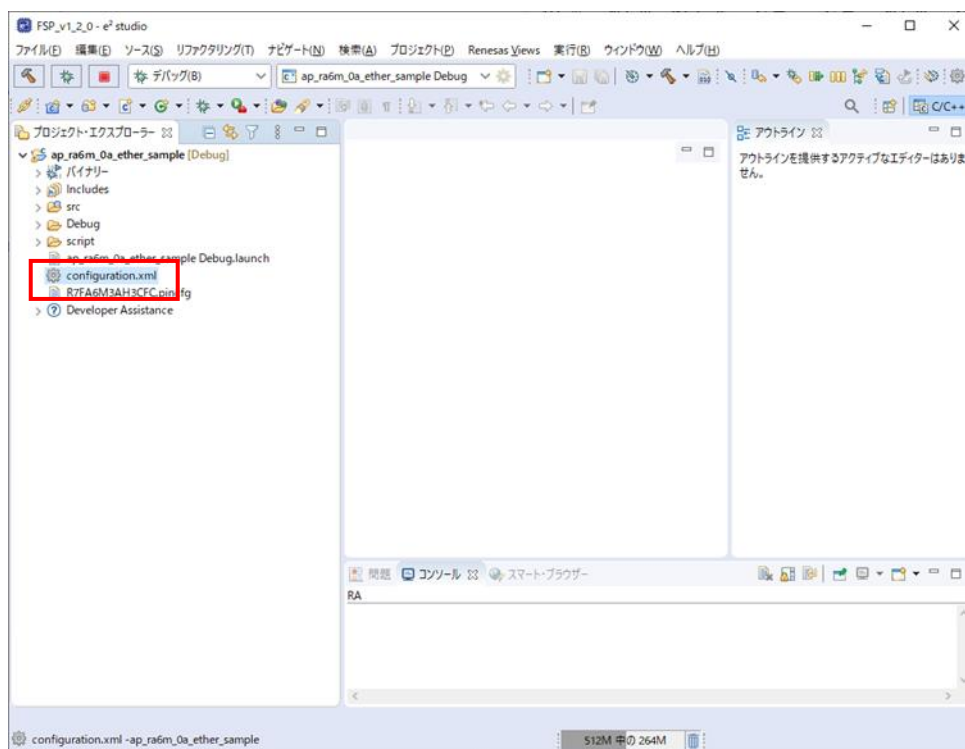
「[3.1 プロジェクトの準備](#)」で用意した、元とする AP-RZA3-0A の Ethernet サンプルプログラムを e2 studio にインポートし、以下の手順でプロジェクトの設定を行ってください。

サンプルプログラムの e2 studio へのインポート方法は後述の「[4.2 e2 studio を用いたプロジェクトのビルド・デバッグ](#)」をご参照ください。

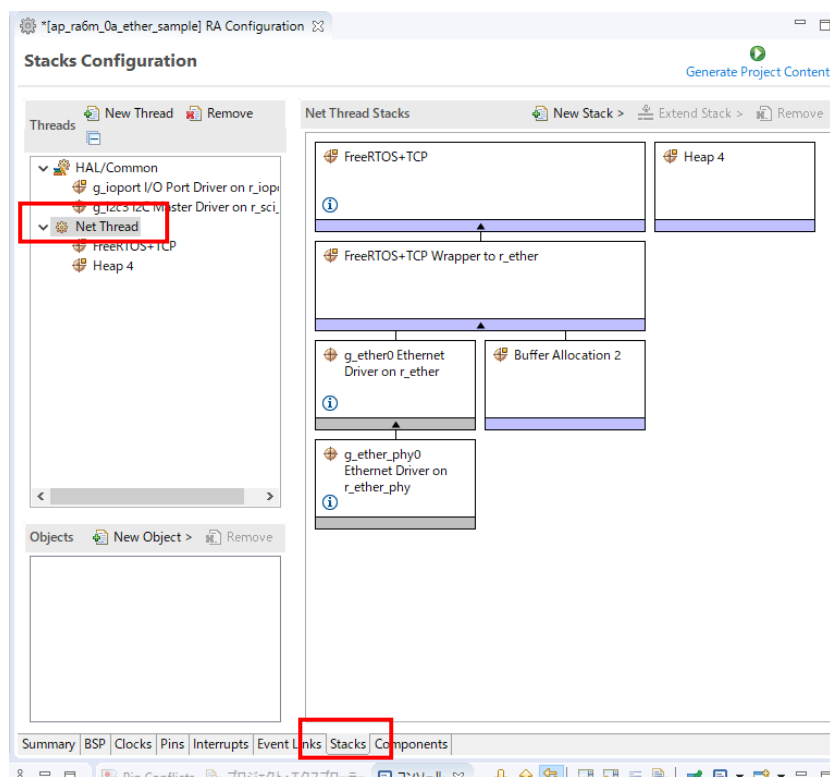
- ① エクスプローラ上で、本サンプルプログラムのフォルダ内から、元とするサンプルプログラムのフォルダに、以下のフォルダをコピーします。
  - ¥wolfConfig フォルダ
  - ¥src¥wolfssl フォルダ



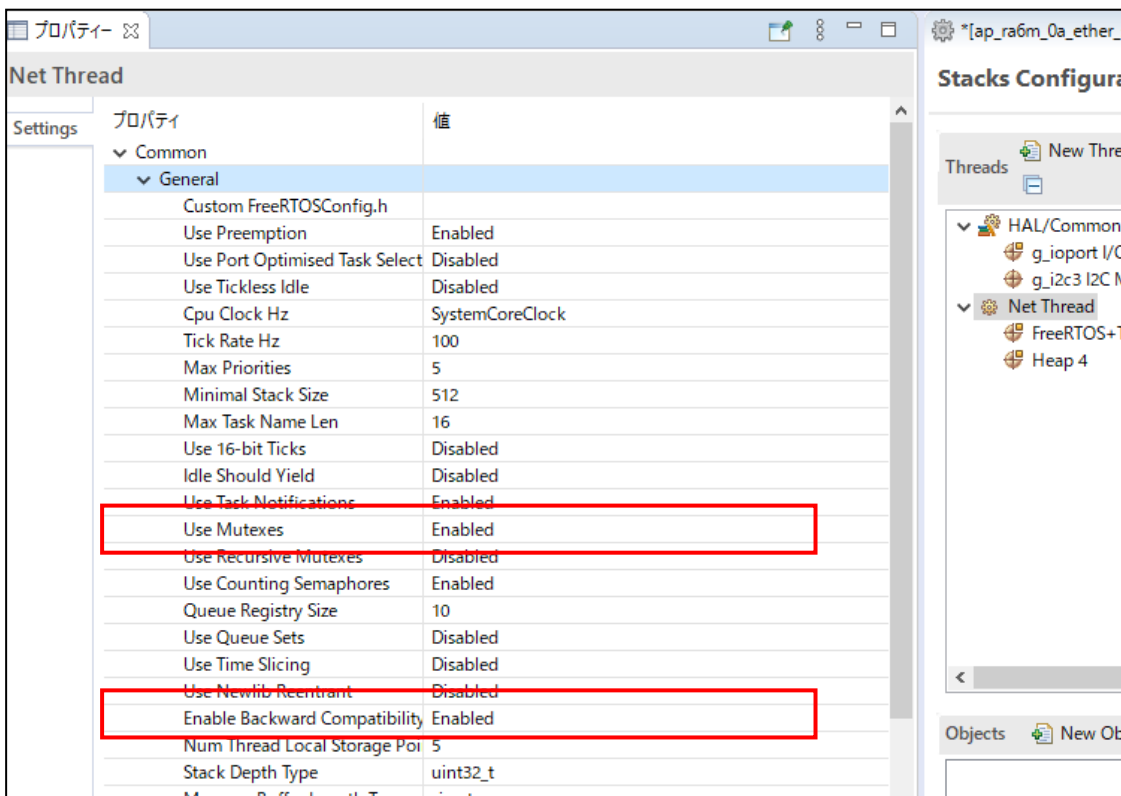
- ② e2 studio 上で、サンプルプログラムのコンフィギュレータファイルを開きます。



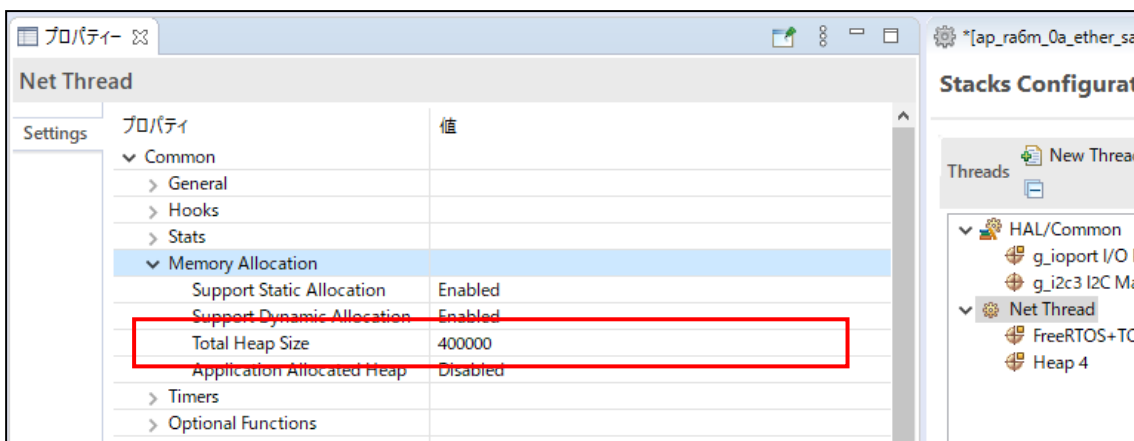
- ③ Stacksタブを開き、Net Thread を選択します。



④ 「プロパティ」ウィンドウを開き、Common 以下の一部パラメータを変更します。

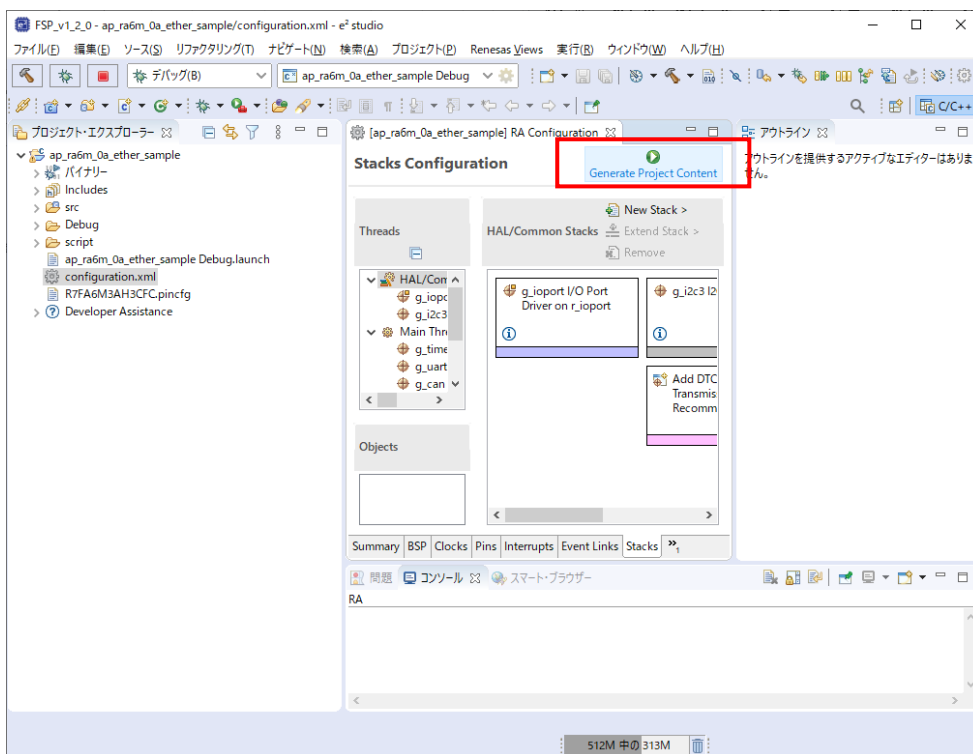


プロパティ	値	備考
▼Common		
▼General		
Use Mutexes	<b>Enable</b>	default : Disabled
Enable Backward Compatibility	<b>Enable</b>	default : Disabled

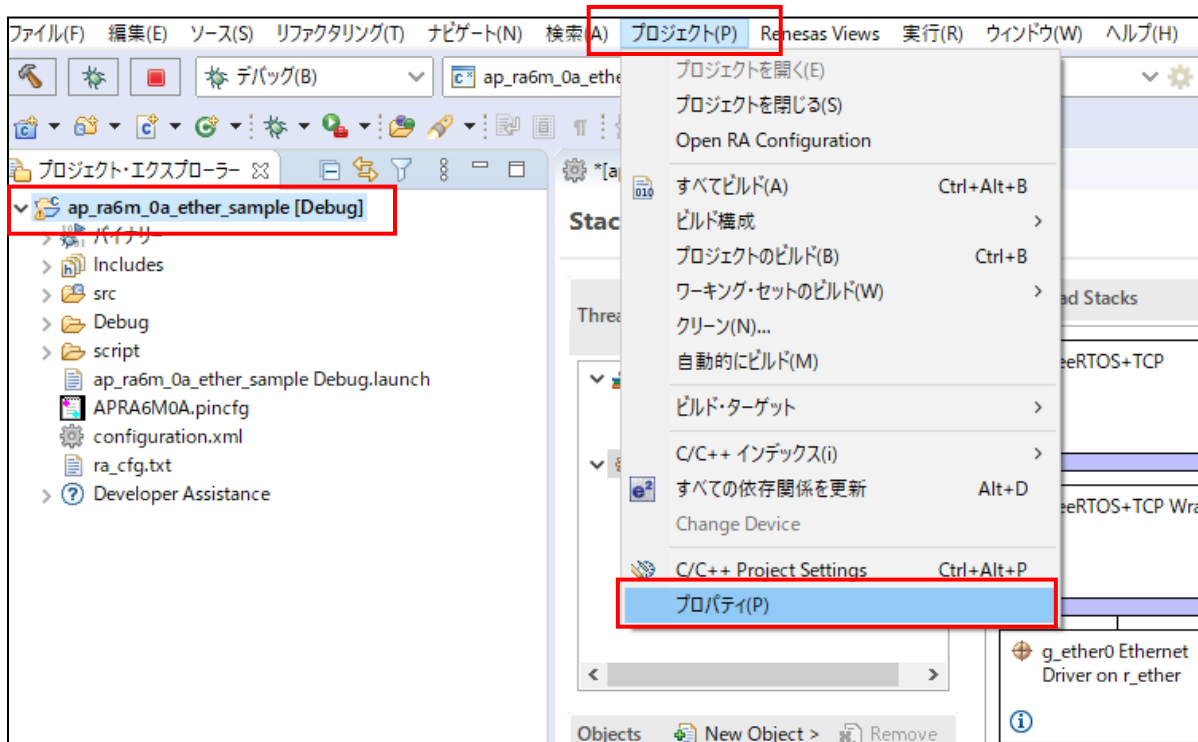


プロパティ	値	備考
▼Common		
▼Memory Allocation		
Total Heap Size	<b>400000</b>	default : 51200

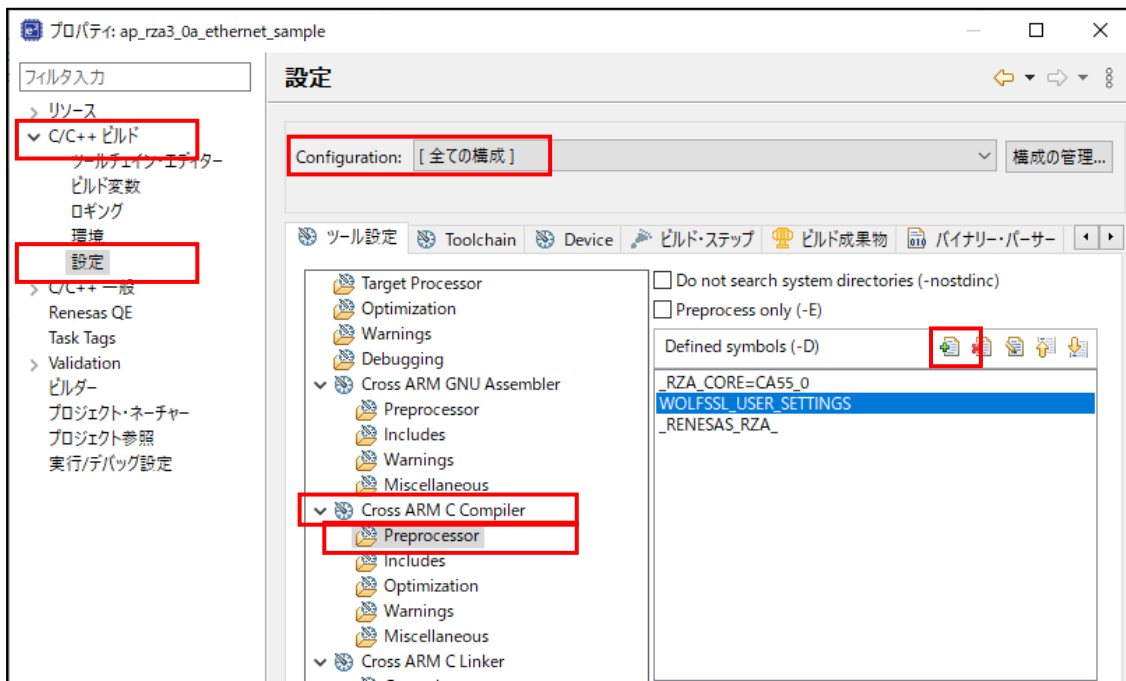
- ⑤ 「Generate Project Content」 をクリックし、自動作成ファイルを出かして設定をプロジェクトに適用します。



- ⑥ プロジェクト・エクスプローラーにて、サンプルプログラムを選択したうえでメニューバー「プロジェクト」>「プロパティ」を選択します。



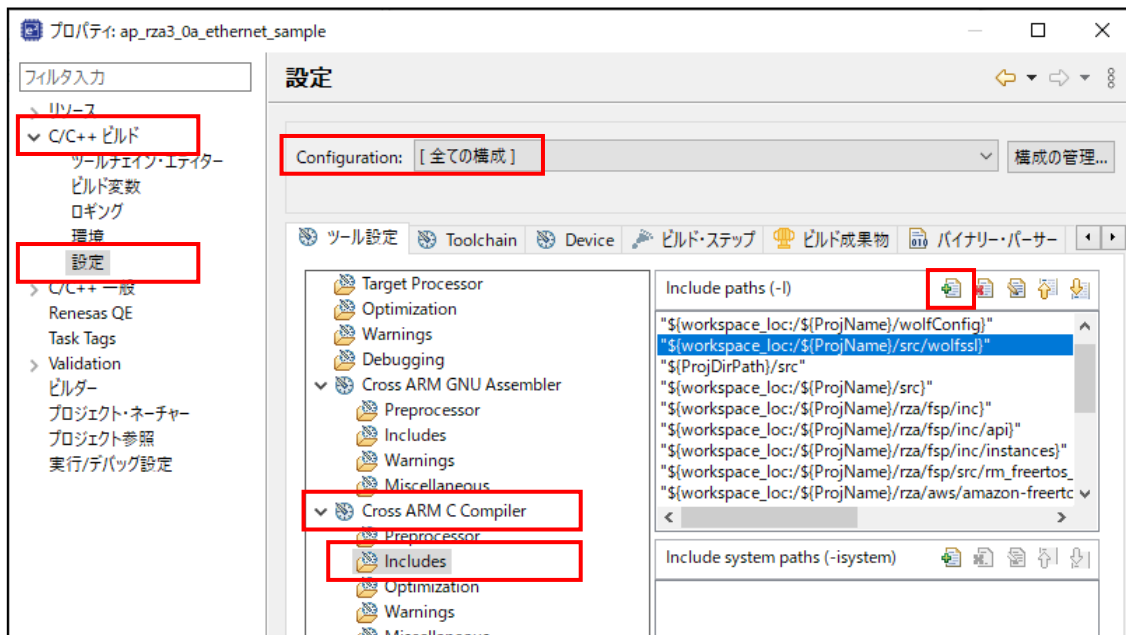
- ⑦ 表示されたプロパティウィンドウの「C/C++ビルド」>「設定」を開き、「全ての構成」における「Cross ARM C Compiler」>「Preprocessor」の「Defined symbols」に以下のマクロを追加します。



マクロ

WOLFSSL\_USER\_SETTINGS

- ⑧ 続けてプロパティウィンドウ内の「C/C++ビルド」>「設定」を開き、「全ての構成」における「Cross ARM C Compiler」>「Includes」の「Include paths」に以下のパスを追加します。

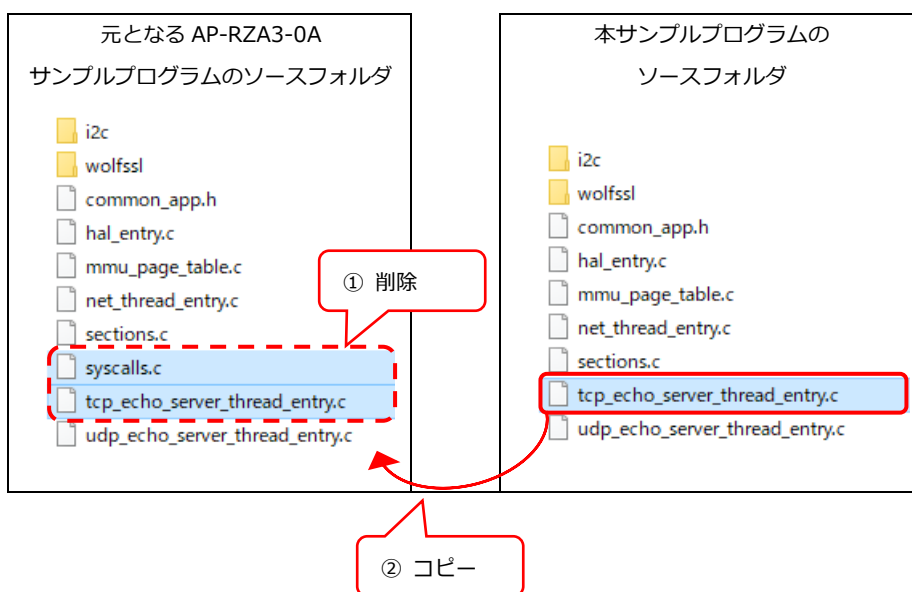


パス
"\${workspace_loc}/\${ProjName}/wolfConfig"
"\${workspace_loc}/\${ProjName}/src/wolfssl"

### 3.3 ソースファイルの変更

元となる AP-RZA3-0A サンプルプログラムのソースに対し、以下の変更を行います。

- ① ソースファイル「src¥syscalls.c」、及び「src¥ tcp\_echo\_server\_thread\_entry.c」を削除します。
- ② 本サンプルプログラム内のソースファイル「src¥tcp\_echo\_server\_thread\_entry.c」を、元とするサンプルプログラムの「src」フォルダにコピーします。



### 3.4 ソースファイルのビルド設定

ここでは、本機能のビルドに不要なファイルを「ビルドから除外」する設定を行います。

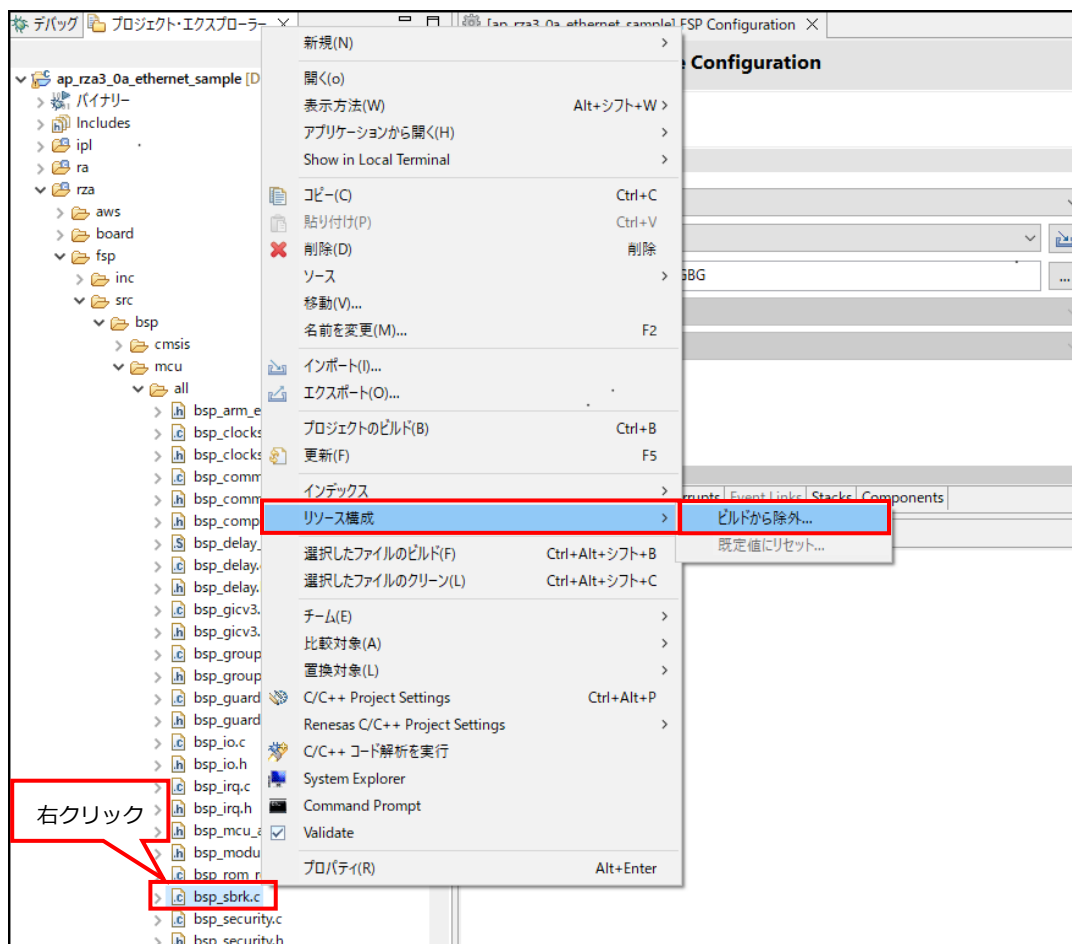
#### 3.4.1 除外ソースファイル一覧

除外対象のソースファイルは以下になります。

- rza¥fsp¥src¥bsp¥mcu¥all¥bsp\_sbrk.c
- src¥wolfssl¥src¥bio.c
- src¥wolfssl¥wolfcrypt¥src¥evp.c
- src¥wolfssl¥wolfcrypt¥src¥misc.c

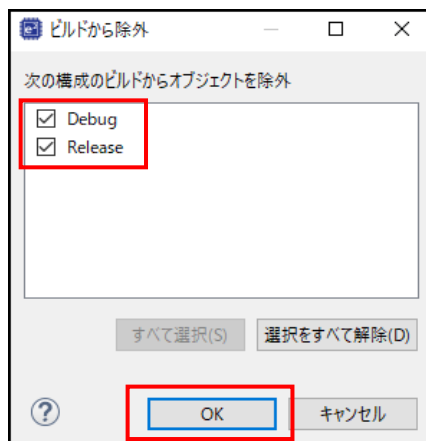
#### 3.4.2 除外手順

- ① 除外するソースファイルを右クリックしてメニューを開き、メニュー「リソース構成」>「ビルドから除外...」を選択します。





- ② 「Debug」「Release」のチェックボックスにチェックをして、「OK」を押してください。



以上で、Ethernet サンプルプログラムに wolfSSL 機能が実装できました。

後述の「[4.2 e2 studio を用いたプロジェクトのビルド・デバッグ](#)」を参考にプログラムをビルド・デバッグして、プログラムが「[4.1 サンプルプログラムの動作](#)」の通りに動作することをご確認ください。

## 4. 動作説明

### 4.1 サンプルプログラムの動作

#### 4.1.1 サンプルプログラム動作説明

本サンプルプログラムは、下記の動作を行います。

- ネットワーク通信

Ethernet で wolfSSL を利用したサーバ動作を行います。

#### 4.1.2 サンプルプログラム サーバ動作詳細

wolfSSL を利用したネットワーク動作（サーバ）の確認は、以下の手順に従って行ってください。

IP アドレス等を変更する場合には、「[4.1.3 ネットワーク設定](#)」を参照してください。

- ① 「[1.2 接続概要](#)」を参考に CPU ボードとホスト PC を接続します。  
LAN クロスケーブルは、LAN コネクタ（CN5）と接続してください。
- ② ホスト PC 上でネットワークの設定を行います。  
CPU ボードの設定に合わせるため、ホスト PC のネットワーク設定を下記の内容に変更してください。

IP アドレス	192.168.0.101
サブネットマスク	255.255.255.0
ゲートウェイ	192.168.0.3

- ③ CPU ボードに電源を投入し、サンプルプログラムを動作させます。
- ④ ホスト PC 上で、フォルダ「¥sample¥PC\_Application」内の「client.bat」を実行します。  
「client.bat」の実行により「¥sample¥PC\_Application¥EXE¥client.exe」が起動し、ホスト PC がクライアントとして動作します。  
※ 「client.bat」「client.exe」の実行には、「certs」フォルダが必要です。「certs」フォルダは削除しないでください。  
※ 「client.bat」は、CPU ボードが以下の設定であることを前提に作成されています。

IP アドレス	192.168.0.100
ポート番号	10000

- ⑤ CPU ボード（サーバ）とホスト PC（クライアント）の接続が完了すると、ホスト PC（client.exe）は CPU ボードに「hello wolfssl!」を送信します。

以上で、サーバ動作の確認は終了です。

SSL 通信の確認に関しては、「[4.1.4 通信プロトコルの確認方法](#)」を参照してください。

※ CPU ボードの IP アドレスやポート番号を初期値から変更している場合、「client.bat」の変更が必要です。  
バッチファイルの書き換えについては、「[5.3 PC アプリの作成](#)」を参考にしてください。

### 4.1.3 ネットワーク設定

- 推奨環境

本サンプルプログラムに実装されたネットワーク通信の確認に必要な推奨環境は以下の通りです。

ホスト PC	PC/AT 互換機
OS	Windows 10/11
LAN ポート	10/100BASE-TX 以上対応の LAN ポート
LAN ケーブル	クロスケーブル

- ネットワーク設定

本 CPU ボードのネットワーク設定は以下の通りです。

項目	内容
IP アドレス	192.168.0.100
サブネットマスク	255.255.255.0
ゲートウェイ	192.168.0.3
MAC アドレス	00-0C-7B-56-XX-XX ※ XX-XX の値は製品ごとに異なります。

IP アドレス・サブネットマスク・ゲートウェイ・ポート番号の設定は、下記のソースファイル上で定義しています。

- IP アドレス・サブネットマスク・ゲートウェイのネットワーク設定

(ap\_rza3\_0a\_ethernet\_sample¥src¥net\_thread\_entry.c)

<ソース例>

```
#include "net_thread.h"
#include "FreeRTOS_IP.h"
#include "i2c¥eeprom.h"
#include "common_app.h"

static uint8_t ucIPAddress[ 4 ]      = {192, 168, 0, 100}; ←IP アドレス設定
static uint8_t ucNetMask[ 4 ]       = {255, 255, 255, 0}; ←サブネットマスク設定
static uint8_t ucGatewayAddress[ 4 ] = {192, 168, 0,3}; ←ゲートウェイ設定
static uint8_t ucDNSServerAddress[ 4 ] = {10, 60, 1, 2};

/* Net thread entry function */
/* pvParameters contains TaskHandle_t */
void net_thread_entry(void *pvParameters)
{
```

- ポート番号のネットワーク設定 (ap\_rza3\_0a\_ethernet\_sample¥src¥tcp\_echo\_server\_thread\_entry.c)

<ソース例>

```
(前略)

xTCPListenSocket = FreeRTOS_socket( FREERTOS_AF_INET, FREERTOS SOCK_STREAM,
FREERTOS_IPPROTO_TCP );
if( xTCPListenSocket == FREERTOS_INVALID_SOCKET )
{
    vTaskDelete(NULL);
}
xServerAddress.sin_family = FREERTOS_AF_INET;
xServerAddress.sin_port = FreeRTOS_htons( 10000 ); ←ポート番号設定
lResult = FreeRTOS_bind( xTCPListenSocket, &xServerAddress, sizeof(xServerAddress) );
if( lResult < 0 )
{
    vTaskDelete(NULL);
}

(後略)
```

また、MACアドレスはEEPROMの先頭6Byteに格納されています。

アドレス	格納値
先頭アドレス + 0x00	0x00
+ 0x01	0x0C
+ 0x02	0x7B
+ 0x03	0x56
+ 0x04	0xXX
+ 0x05	0xXX

※ 0xXXの値は製品ごとに異なります

本製品のMACアドレスは、弊社が米国電気電子学会（IEEE）より取得したアドレスとなります。

MACアドレスを変更される際は、お客様にてIEEEよりMACアドレスを取得し、設定してください。

## 4.1.4 通信プロトコルの確認方法

## ● 使用ソフトウェア

通信プロトコルの確認には、ネットワークアナライザを使用します。

ここでは、フリーソフトウェア「Wireshark」を使用した確認方法を説明します。

「Wireshark」は下記の Web サイトからダウンロードできます。

Wireshark 公式サイト： <https://www.wireshark.org/>

## ● 確認方法

本節の説明では、「Wireshark v3.0.0」を使用しています。

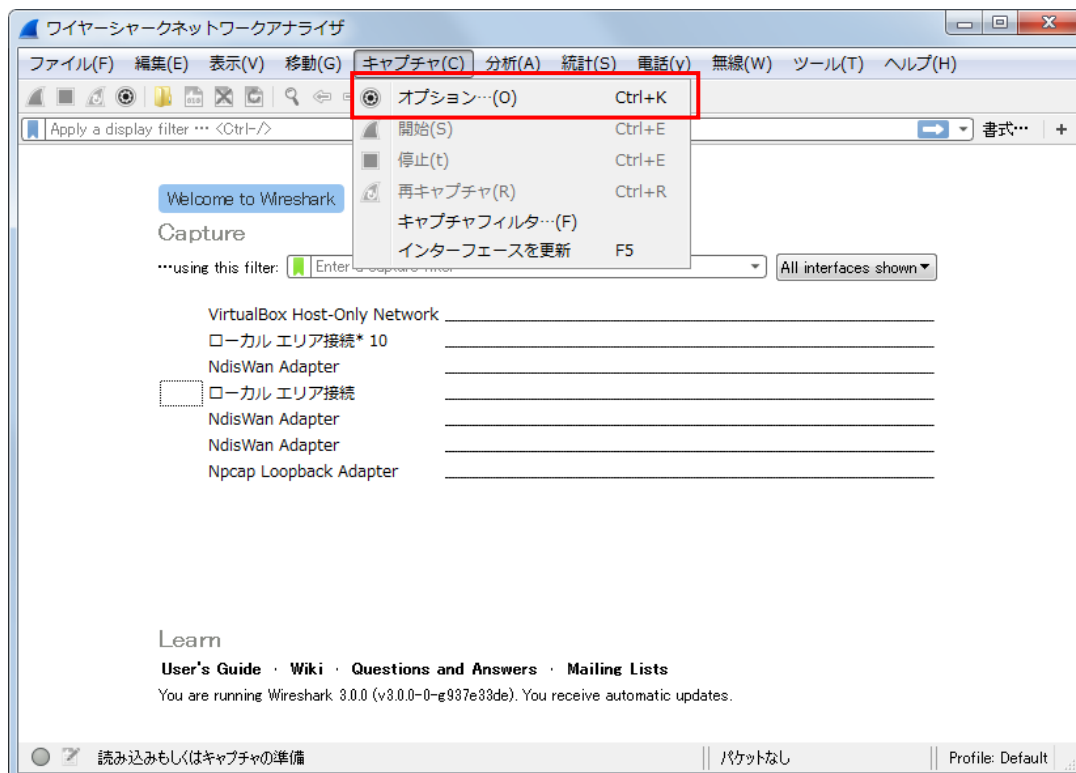
その他のバージョンをご利用の場合、読み替えて行ってください。

※Wireshark のバージョンにより、詳細なプロトコルが表示されないことがあります。

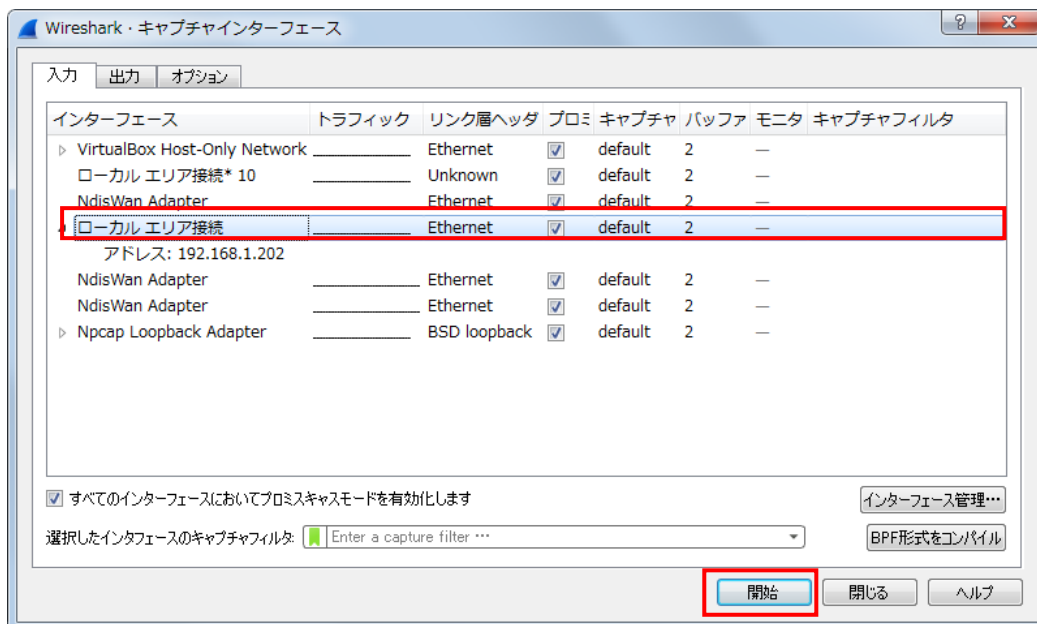
その場合、Wireshark のバージョンアップを行ってください。

※本節の説明ではホスト PC 側 IP アドレスは「192.168.0.101」、CPU ボード側 IP アドレスは「192.168.0.100」で説明しております。

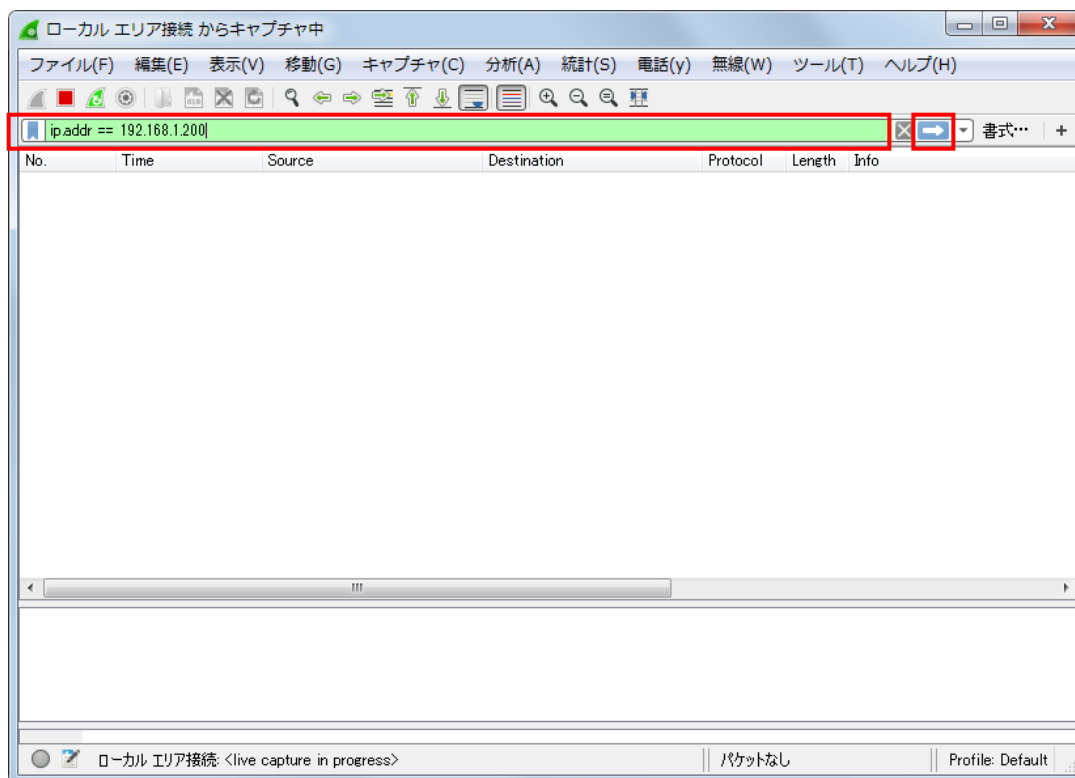
① Wireshark を起動し、メニューの「キャプチャ」-「オプション」を選択します。



- ② ホスト PC で使用するインタフェースをクリックし、「開始」を押します。  
ここでは、「ローカル エリア接続」（アドレス：192.168.0.101）を選択します。

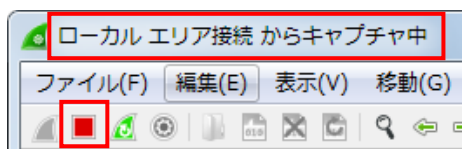


- ③ CPU ボードの IP アドレスを含むパケットのみキャプチャするように、フィルタを設定します。  
フィルタ入力欄に「ip.addr == 192.168.0.100」を入力します。  
入力後は Enter キーを押すか、Apply ボタンを押して確定してください。



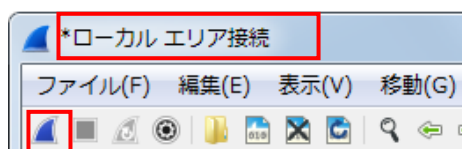
- ④ 接続を選択すると自動でキャプチャが開始します。  
 手動でキャプチャを開始する場合、開始ボタンを押してキャプチャを開始してください。  
 キャプチャを開始した後、サンプルプログラムを動作させます。  
 サンプルプログラムの動作方法に関しては、「[4.1.2 サンプルプログラム サーバ動作詳細](#)」を参照してください。

<キャプチャ中の表示>



キャプチャ停止  
ボタン

<停止中の表示>



キャプチャ開始  
ボタン

- ⑤ 取得したデータを確認します。  
 「Protocol」に「**TLSvX.X**」(X.Xはバージョン)の記載があれば、その通信は暗号化通信をしています。

No.	Time	Source	Destination	Protocol	Length	Info
45	12.639908	192.168.1.202	192.168.1.200	TCP	54	55277 → 50000 [ACK] Seq=446 Ack=
46	12.650201	192.168.1.200	192.168.1.202	TCP	111	50000 → 55277 [ACK] Seq=1855 Ack=
47	12.650201	192.168.1.200	192.168.1.202	TLSv1.3	60	Application Data
48	12.650281	192.168.1.202	192.168.1.200	TCP	54	55277 → 50000 [ACK] Seq=446 Ack=
49	12.651070	192.168.1.202	192.168.1.200	TLSv1.3	112	Application Data
50	12.651635	192.168.1.200	192.168.1.202	TCP	60	50000 → 55277 [ACK] Seq=1913 Ack=
51	12.654433	192.168.1.202	192.168.1.200	TLSv1.3	90	Application Data
52	12.654994	192.168.1.200	192.168.1.202	TCP	60	50000 → 55277 [ACK] Seq=1913 Ack=
53	12.719935	192.168.1.200	192.168.1.202	TCP	89	50000 → 55277 [ACK] Seq=1913 Ack=
54	12.719935	192.168.1.200	192.168.1.202	TLSv1.3	60	Application Data
55	12.720050	192.168.1.202	192.168.1.200	TCP	54	55277 → 50000 [ACK] Seq=540 Ack=
56	12.720404	192.168.1.202	192.168.1.200	TLSv1.3	78	Application Data
57	12.720477	192.168.1.202	192.168.1.200	TCP	54	55277 → 50000 [FIN, ACK] Seq=56
58	12.720623	192.168.1.200	192.168.1.202	TCP	60	50000 → 55277 [ACK] Seq=1949 Ack=
59	12.720623	192.168.1.200	192.168.1.202	TCP	60	50000 → 55277 [ACK] Seq=1949 Ack=

▶ Frame 15: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0  
 ▶ Ethernet II, Src: HewlettP\_7b:aa:66 (dc:4a:3e:7b:aa:66), Dst: AlphaPro\_47:ff:00 (00:0c:7b:47:ff:00)  
 ▶ Internet Protocol Version 4, Src: 192.168.1.202, Dst: 192.168.1.200  
 ▶ Transmission Control Protocol, Src Port: 55277, Dst Port: 50000, Seq: 0, Len: 0

以上で通信プロトコルの確認は終了です。

Wiresharkの詳細な使用方法については、Wiresharkのマニュアル等をご覧ください。



## 4.2 e2 studio を用いたプロジェクトのビルド・デバッグ

サンプルプログラムを CPU ボード上で実行するためには、e2 studio 上に一度サンプルプログラムをインポートし、ビルドを行う必要があります。

また、サンプルプログラムを CPU ボード上で実行するためには、ビルドしたサンプルプログラムの実行ファイルを CPU ボードにダウンロードする必要があります。

サンプルプログラムのビルド方法、CPU ボードにサンプルプログラムをダウンロードする方法、ボードのシリアル FlashROM へ書き込んで実行する方法については、以下のアプリケーションノートに詳細な手順が記されています。

- ・ **AN1653 RZ/A3UL 開発チュートリアル**

## 5. wolfSSL の入手方法

本章では、wolfSSL 組込み向け軽量 SSL/TLS ライブラリの最新版を入手し、サンプルプログラムに組み込む方法を説明します。

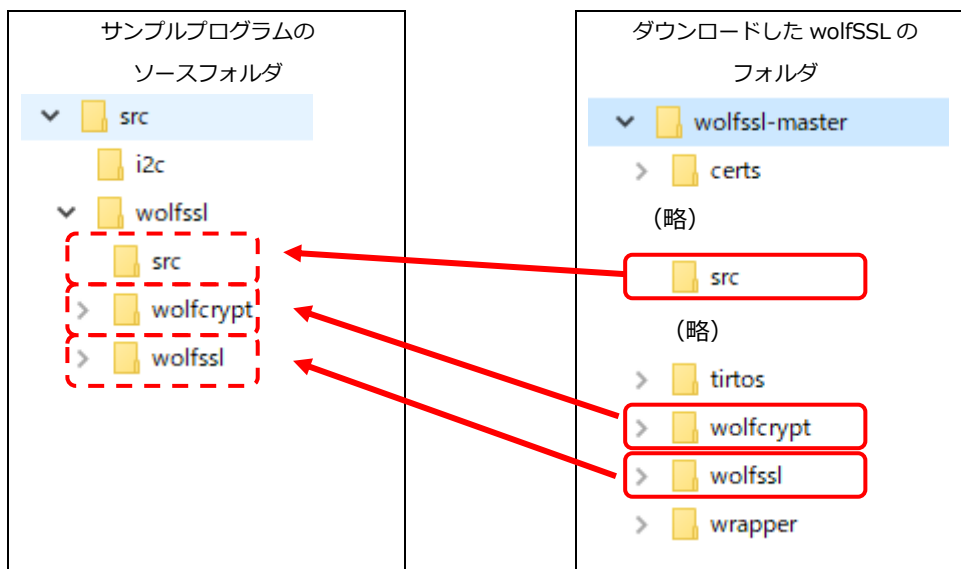
### 5.1 wolfSSL の入手

wolfSSL は、wolfSSL 社 Web サイトから入手できます。必要なバージョンをダウンロードしてください。ライセンスや商用利用につきましては、「1.6 wolfSSL について」および wolfSSL 社 Web サイトをご覧ください。

wolfSSL 社 組込み SSL ライブラリページ：<https://www.wolfssl.jp/products/wolfssl/>  
(wolfSSL 日本語サイト：<https://www.wolfssl.jp/>)

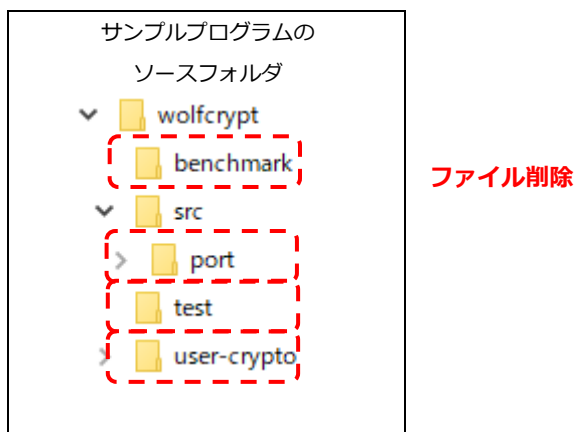
## 5.2 ライブラリの作成、差し替え

wolfSSL ファイルを、サンプルプログラム内のファイルと差し替えます。  
差し替えるファイルは、サンプルプログラム内の同名ファイルです。



ただし、「wolfcrypt」フォルダ内の以下のものを削除します。

- 「benchmark」フォルダ
- 「test」フォルダ
- 「user-crypto」フォルダ
- 「src」フォルダ内の「port」フォルダ
- 「src」フォルダ内の拡張子が「c」ではないファイル



### 5.3 PC アプリの作成

サンプルプログラムの動作確認に使用する PC アプリを作成します。

ここでの説明は、「Microsoft Visual Studio 2019 Professional」を元に行います。

Visual Studio の使用方法については、Visual Studio のマニュアル等をご覧ください。

- ① Microsoft Visual Studio を用いて、ダウンロードしたファイルに含まれる「wolfssl64.sln」を読み込みます。  
「wolfssl64.sln」は、「wolfssl-master.zip」をダウンロードした場合、「¥wolfssl-master」に存在します。
- ② 各プロジェクトのプロパティを開き、ソースファイルの文字コードを設定します。  
「wolfssl-master.zip」をダウンロードした場合、ソースファイルの文字コードは「UTF-8」のため、「プロパティ」-「C/C++」-「コマンドライン」から、「/source-charset:utf-8」を入力します。
- ③ メニューの「ビルド」-「ソリューションのリビルド」を実行し、ソリューション内の全てのプロジェクトをビルドします。  
ビルドが完了すると、フォルダ「¥Debug」に実行ファイル（拡張子.exe）が生成されます。
- ④ 実行ファイルを使用するためのバッチファイルを作成します。  
例（サンプルプログラム内のバッチファイル）

<client.exe 用バッチファイル>

```
cd .¥EXE
client.exe -v 4 -h "192.168.0.100" -p 10000 -A ./certs/1024/ca-cert.pem
```

※接続するサーバの IP アドレス、ポート番号を指定します。

※「-v 4」は、TLSv1.3 を指定しています。

- ※ ダウンロードした wolfssl のフォルダから「./examples/server」の「server.c」の 1395 行目を下記のように変更します。  
また、「./examples/client」の「client.c」の 1835 行目も同様な変更を行います。

```
{ "ヘルプ", 0, 258 }
↓
{ "help, 0, 258" }
```

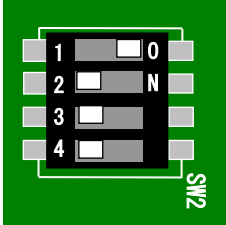
## 6. 開発環境使用時の各設定値

開発環境を使用する際の、AP-RZA3-0A 固有の設定を以下に示します。

ビルド・動作確認方法	
項目名	設定値
サンプルプログラムフォルダ	sample¥ap_rza3_0a_ethernet_sample
プロジェクト	ap_rza3_0a_ethernet_sample
デバッグ時のボード設定	「6.1 スイッチ設定」参照
デバッグ用出力フォルダ	ap_rza3_0a_ethernet_sample ¥Debug
デバッグ用実行ファイル	ap_rza3_0a_ethernet_sample.elf
Debug hardware	J-Link ARM
Target Device	R9A07G063U02GBG
SerialFlash 書き込み用フォルダ	ap_rza3_0a_ethernet_sample ¥Release
書き込みファイル	ap_rza3_0a_ethernet_sample.srec

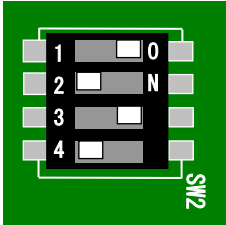
## 6.1 スイッチ設定

・デバッグをする場合



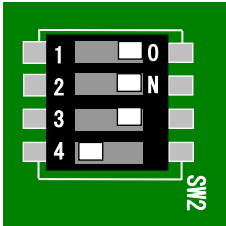
<SW2 設定>	
SSCG 設定	: 不問
ブートモード	: SPI Flash ブートモード(OFF)
デバッグ設定	: デバッグモード(OFF)

・SPI Flash ブートをする場合



<SW2 設定>	
SSCG 設定	: 不問
ブートモード	: SPI Flash ブートモード(OFF)
デバッグ設定	: ノーマルモード(ON)

・SD ブートをする場合



<SW2 設定>	
SSCG 設定	: 不問
ブートモード	: SD ブートモード(ON)
デバッグ設定	: ノーマルモード(ON)

Fig6.1-1 デバッグ・各ブート時のボード設定

## ご注意

- ・本文書の著作権は株式会社アルファプロジェクトが保有します。
- ・本文書の内容を無断で転載することは一切禁止します。
- ・本文書に記載されているサンプルプログラムの著作権は株式会社アルファプロジェクトが保有します。
- ・本サンプルプログラムで使用されているミドルウェアおよびドライバの著作権はルネサス エレクトロニクス株式会社が保有します。
- ・本文書に記載されている内容およびサンプルプログラムについてのサポートは一切受け付けておりません。
- ・本文書の内容およびサンプルプログラムに基づき、アプリケーションを運用した結果、万一損害が発生しても、弊社では一切責任を負いませんのでご了承ください。
- ・本文書の内容については、万全を期して作成いたしました。万一ご不審な点、誤りなどお気づきの点がありましたら弊社までご連絡ください。
- ・本文書の内容は、将来予告なしに変更されることがあります。

## 商標について

- ・ RZ および RZ/A3UL は、ルネサス エレクトロニクス株式会社の登録商標、商標または商品名称です。
- ・ Arm® は Arm Ltd. の登録商標です。
- ・ e2 studio は、ルネサス エレクトロニクス株式会社の登録商標、商標または商品名称です。
- ・ J-Link は、SEGGER Microcontroller GmbH & Co. KG の登録商標もしくは商標です。
- ・ Flexible Software Package は、ルネサス エレクトロニクス株式会社の登録商標、商標または商品名称です。
  
- ・ Windows® の正式名称は Microsoft® Windows® Operating System です。
- ・ Microsoft、Windows は、米国 Microsoft Corporation. の米国およびその他の国における商標または登録商標です。
- ・ Windows® 10、Windows® 11 は、米国 Microsoft Corporation. の商品名称です。  
本文書では下記のように省略して記載している場合がございます。ご了承ください。  
Windows® 10 は Windows 10 もしくは Win10  
Windows® 11 は Windows 11 もしくは Win11
  
- ・ その他の会社名、製品名は、各社の登録商標または商標です。



株式会社アルファプロジェクト  
〒431-3114  
静岡県浜松市中央区積志町 8 3 4  
<https://www.apnet.co.jp>  
E-Mail : [query@apnet.co.jp](mailto:query@apnet.co.jp)