Rev 1.00.10

Renesas S5D9 用サンプル(e2studio WIRE_DHCP_TCP_RSA)の説明

(e2studio Version:2022-10 / SSP Version 2.4.0)

1．Sample の免責について

・Sample に関する Tel/Fax でのご質問に関してはお受けできません。ただし、メールでのご質問に関してはお答えするよう努力はしますが、都合によりお答えできない場合もありますので予めご了承願います。

・Sample ソフトの不具合が発見された場合の対応義務はありません。また、この関連ソフトの使用方法に関する質問の回答義務もありませんので承知の上ご利用下さい。

・Sample ソフトは、無保証で提供されているものであり、その適用可能性も含めて，いかなる保証も行いません。また，本ソフトウェアの利用により直接的または間接的に生じたいかなる損害に関しても，その責任を負わないものとします。

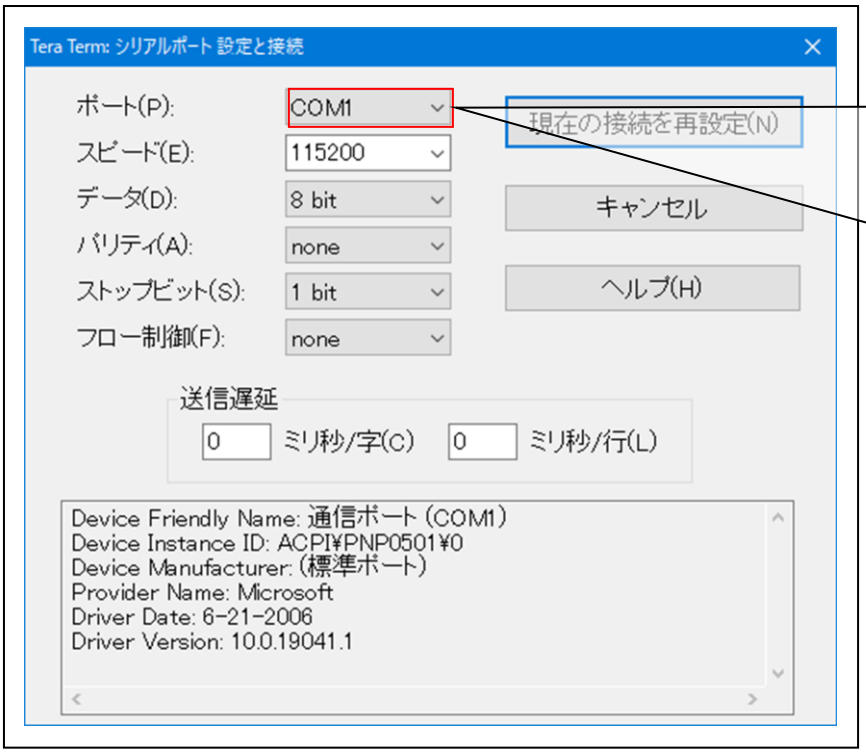
2．サンプルのプロジェクト名

ワークスペース名	概要	プロジェクト名
S5D9_e2std_WIRE_NetX_1	有線 LAN 接続した DHCP と TCP 通信のサンプル セキュリティエンジン(SCE7) Crypto ドライバーの RSA を使用したサンプル（暗号・復号）	WIRE_DHCP_TCP_RSA_ETH Azure RTOS モードで動作 NetX DHCP Client (g_dhcp_client0) TCP 通信(Client) (nx_tcp_socket_.....) 暗号・復号(RSA) (g_sce_rsa_r.p_api->encrypt) (g_sce_rsa_r.p_api->decrypt)

統合開発環境
Renesas e2studio(Version 2022-10)
SSP(Version2.4.0)

3．Tera Term Pro のインストール

①「teraterm-4.106.exe」を検索してダウンロードする。
②PC にインストールし実行する
③シリアルポートの設定

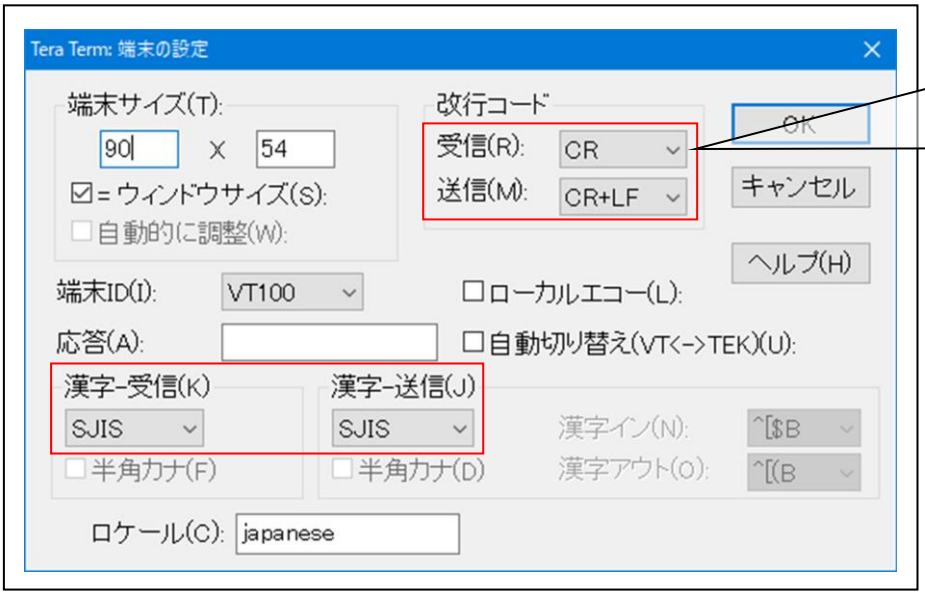


COM 番号は、
PC 側でシリアル通信可能な番号を指定する。

115200BPS
8bit
none
1bit
none

の仕様にする。

④端末の設定

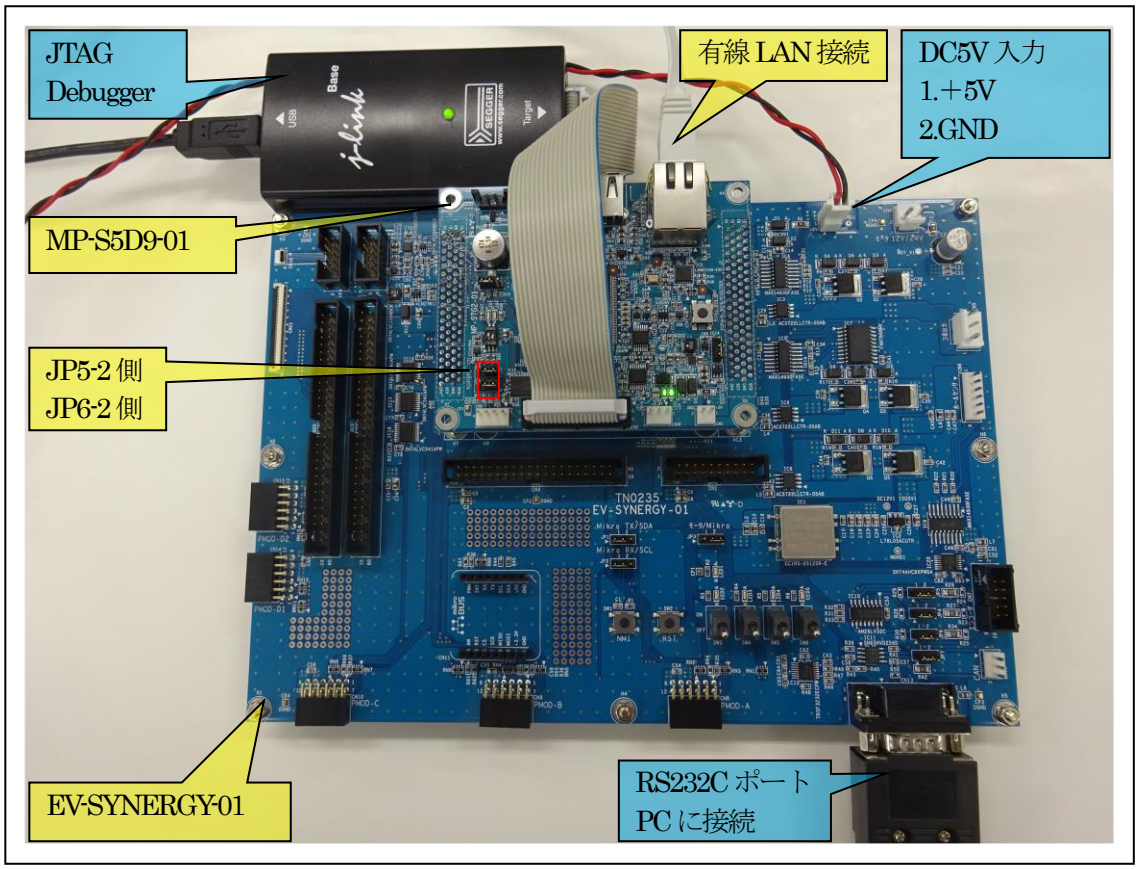


USB シリアルコンバータ使用時に CR コードがカットされる設定の場合は、受信：LF にして下さい。

赤枠の設定にする。

4．動作構成

【EV-SYNERGY-01】を使用の場合



①PC 機と接続する RS232C ケーブルは、市販「クロスケーブル」でも可能です。
②USB-シリアル変換ケーブルを使用される場合は、「StarTech.com 社　ICUSB232FIN」推奨
③自作する場合は、下記の配線になります。

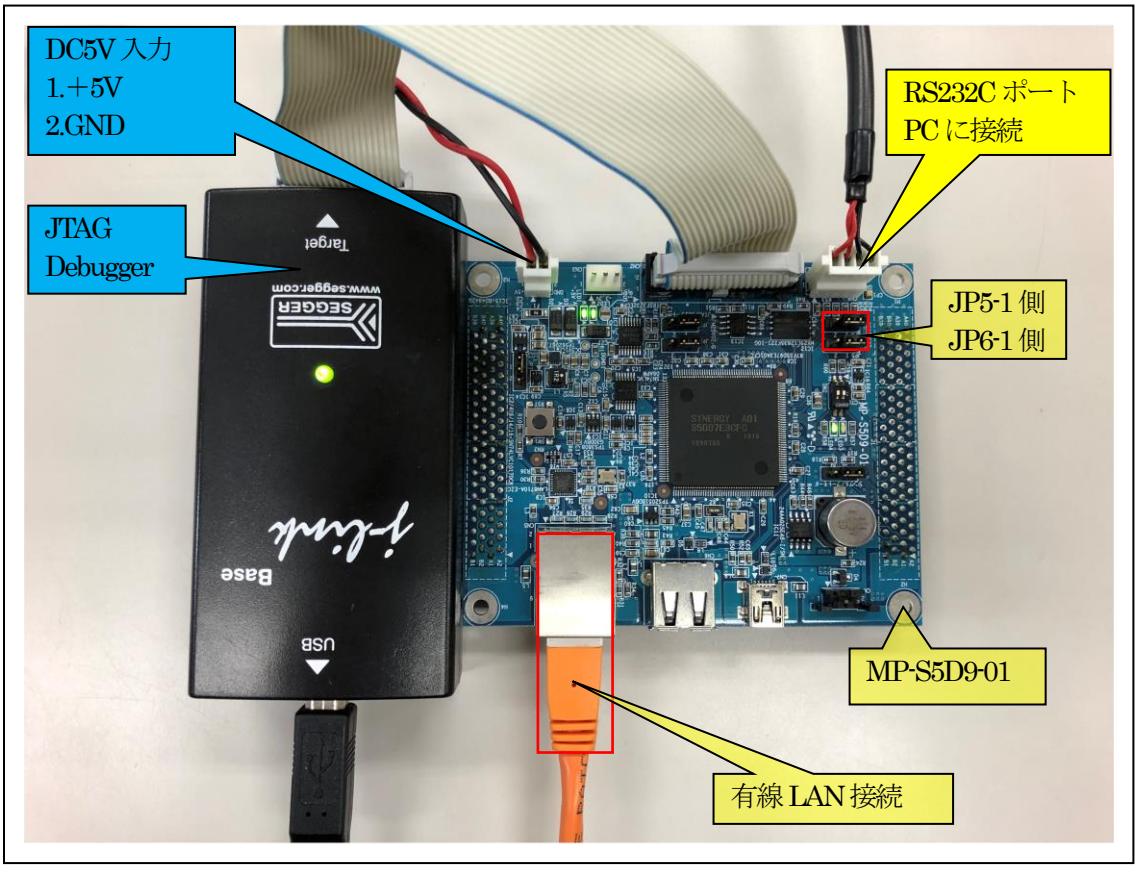
EV-SYNERGY-xx

DSUB 9pin オス	
1	NC
2	RXD
3	TXD
4	NC
5	GND
6	NC
7	RTS
8	CTS
9	NC

PC-Windows

DSUB 9pin オス	
1	NC
2	RXD
3	TXD
4	DTR
5	GND
6	DSR
7	RTS
8	CTS
9	NC

【MP-S5D9-01】のみ使用の場合



①PC 機と接続する RS232C ケーブルは、製作が必要です。
②「RS232C-Driver」は、下記 URL の「RS232CAB4」を推奨します。
http://tool-kobo.ddo.jp/Files/Product/RS232_422/RS232CAB.htm

MP-S5D9-01(CN8)

4pin メス	
1	DC3.3V
2	TXD2
3	RXD2
4	GND

H4P-SHF-AA

RS232C Driver

PC-Windows

DSUB 9pin メス	
1	NC
2	RXD
3	TXD
4	DTR
5	GND
6	DSR
7	RTS
8	CTS
9	NC

５．「S5D9_e2std_WIRE_NetX_1」サンプルの説明

５－１．「WIRE_DHCP_TCP_RSA_ETH」フォルダ構成とファイル名

S5D9_e2std_WIRE_NetX_1¥WIRE_DHCP_TCP_RSA_ETH			
Debug		WIRE_DHCP_TCP_RSA_ETH.elf	ELF ファイル、JTAG で使用
		WIRE_DHCP_TCP_RSA_ETH.map	MAP ファイル、アドレス情報管理
		WIRE_DHCP_TCP_RSA_ETH.srec	モトローラーHEX ファイル
		その他	自動生成ファイル
Script		S5D9.ld	ロケーション定義ファイル
Src		Command.c	コマンド入力処理のサンプルファイル
		Command.h	Command.c 用ヘッダーファイル
		tcp_cmd_thread_entry.c	TCP Cmd Thread サンプルファイル
		wire_dhcp_thred_entry.c	DHCP client Thread サンプルファイル
	sce7_rsa	sce7_rsa.c	RSA の暗号・復号化サンプルファイル
		sce7_rsa.h	sce7_rsa.c 用ヘッダーファイル
	MP-S5D9-01（リンク指定）	e2p.c	E2PROM 処理モジュール
		e2p.h	e2p.c 用ヘッダーファイル
		led.c	LED 処理モジュール
		led.h	led.c 用ヘッダーファイル
		sic2.c	シリアル通信処理モジュール
		sci2.h	sci2.c 用ヘッダーファイル
		stchar.c	文字系処理モジュール
		stcahr.h	stchar.c 用ヘッダーファイル-
	synergy_gen	Generate を行うと作成されるファイル	
Synergy		Generate を行うと作成されるファイル	
synergy_cfg			
Configuration.xml		プロジェクト Generation ファイル	
PIN-MP-S5D9-01.pincfg		PIN configuration 用ファイル	
WIRE_DHCP_TCP_RSA_ETH.jlink		Jlink デバッガー用ファイル	
その他		自動生成ファイル	

５－２．サンプルの動作説明

＜WIRE DHCP Thread＞

１）DHCP による IP アドレスの取得を待つ

Term 画面

＜１＞「"<tcp_thread waiting to get IP address>"」

＜２＞「"<interface status check>"」

MP 基板に実装した EEPROM より MAC アドレス読みドライバにセットする。

＜３＞「<Start WIRE NetX DHCP>"」

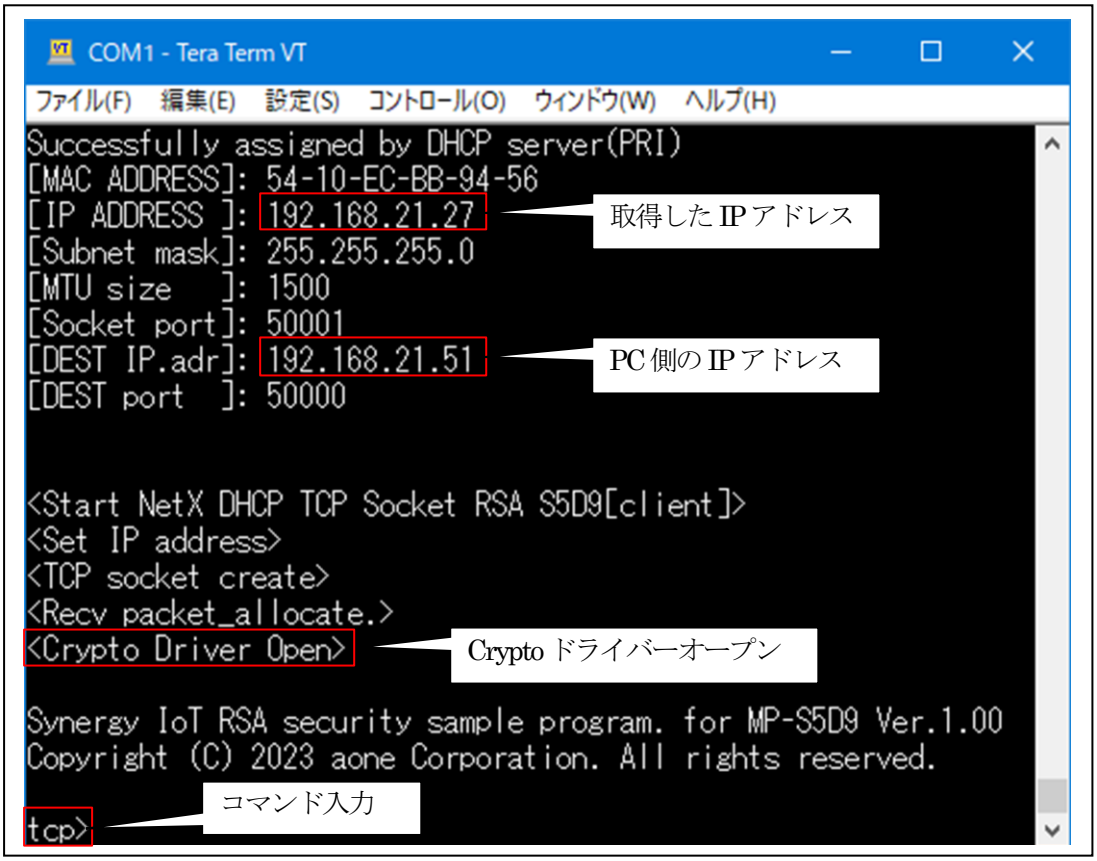
＜４＞「"<Wait DHCP State Change>"」

＜５＞「"<Start WIRE NetX DHCP>"」

＜６＞「"<Wait DHCP BOUND>"」の順次処理して表示する。

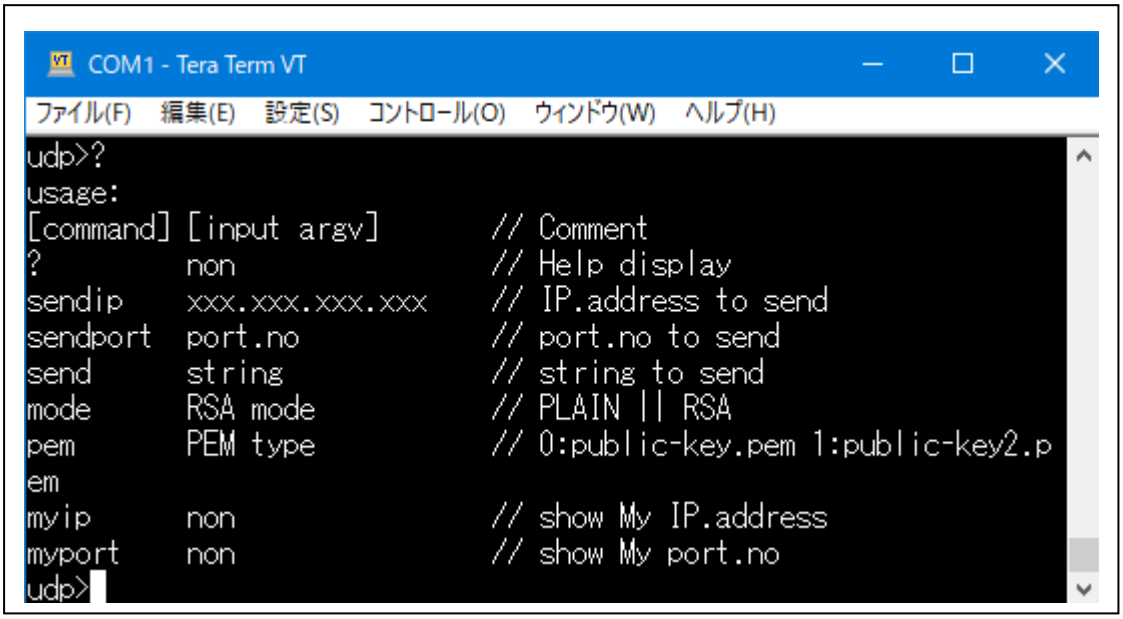
・IP アドレス取得成功により、MP 基板上の LED3 を led blink thread で 100msec 毎に点滅

２）IP ドレス取得情報を Term 画面に表示する。



＜TCP Cmd Thread＞
3）オープニングメッセージを表示してコマンド入力を待つ。
・TCP Thread 起動後、MP 基板上の LED4 を 100msec 毎に点滅

4）コマンドの説明
（1）？（ヘルプコマンド）コマンド一覧の表示



（2）ｓｅｎｄｉｐ（送信先ＩＰアドレスの登録と参照）
＜機能＞
・送信先（PC機側）のＩＰアドレスを登録する。

＜例＞
xxx> sendip 192.168.21.9⏎
xxx> sendip⏎
[send IP]: 192.168.21.51

（3）ｓｅｎｄｐｏｒｔ（送信先ポート番号の登録と参照）
＜機能＞
・送信先（PC機側）のポート番号を登録する。

＜例＞
xxx> sendport 50000⏎
xxx> sendport⏎
[send port]: 50000

（4）ｓｅｎｄ（文字列の送信）
＜機能＞
・送信先（PC機側）に文字列を送信する。

＜例＞
xxx> send 12345678123456781234567812345678⏎

（5）ｍｏｄｅ（暗号モードの設定と参照）

＜機能＞

・送信先（PC機側）に文字列を送信するための暗号モードを指定する。

①<PLAIN>平文で送信

②<RSA>文字列を「RSA モード」で暗号化して送信する。

＜例＞

xxx> mode PLAIN↵

xxx> mode RSA↵

xxx> mode↵

[Now mode]: RSA

（6）ｐｅｍ（PEM タイプの指定）

＜機能＞

・暗号・復号化するための公開鍵・秘密鍵の PEM タイプを指定する。

①<0>(公開鍵)public-key.pem ：(秘密鍵) private-key.pem を使用する。

②<1>(公開鍵)public-key2.pem：(秘密鍵) private-key2.pem を使用する。

＜例＞

xxx>pem 0↵

xxx>pem 1↵

（7）ｍｙｉｐ（DHCP で取得した自ＩＰアドレスの参照）

＜機能＞

・DHCP で取得した自ＩＰアドレスを表示する。

＜例＞

xxx>myip↵

[My IP]: 192.168.21.9

（8）ｍｙｐｏｒｔ（基板側で登録したポート番号の参照）

＜機能＞

・基板側で登録したポート番号を表示する。

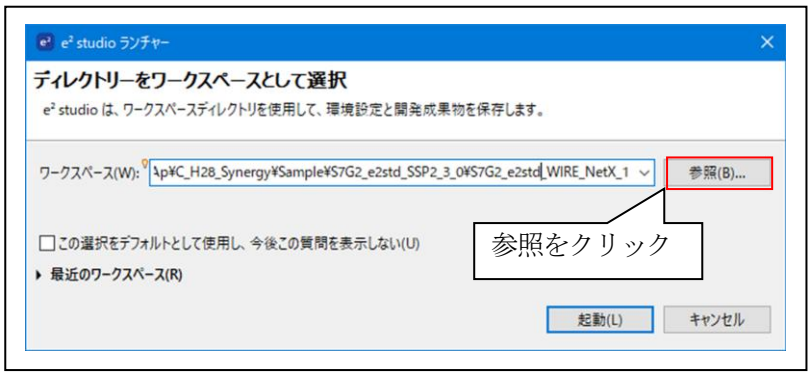
＜例＞

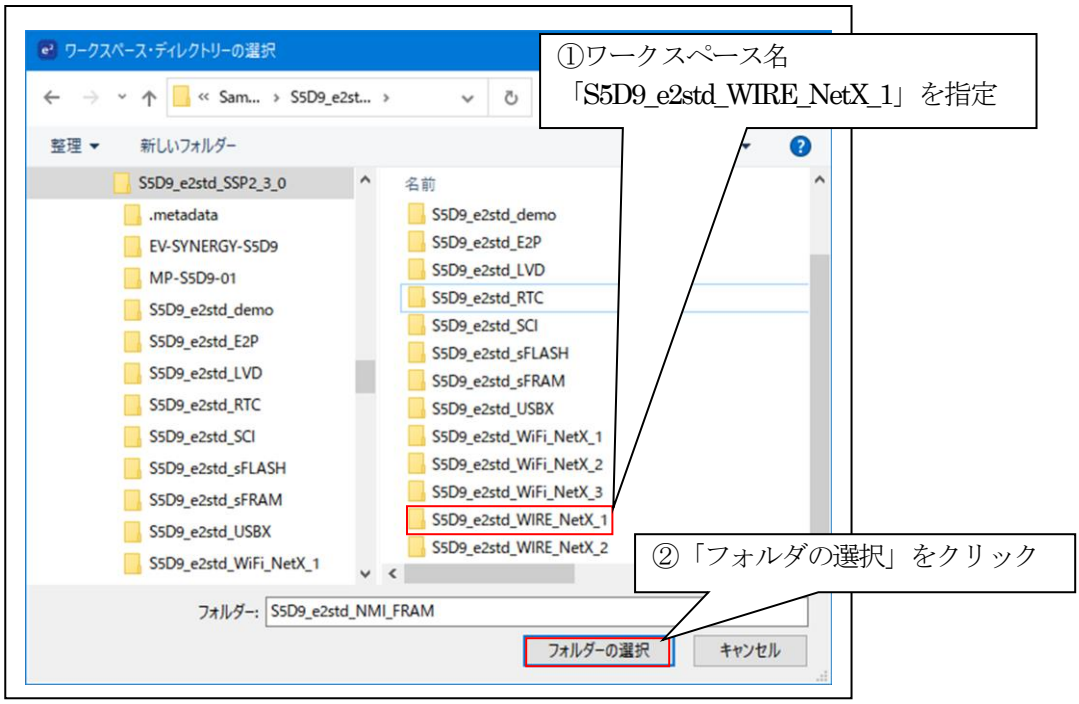
xxx>myport↵

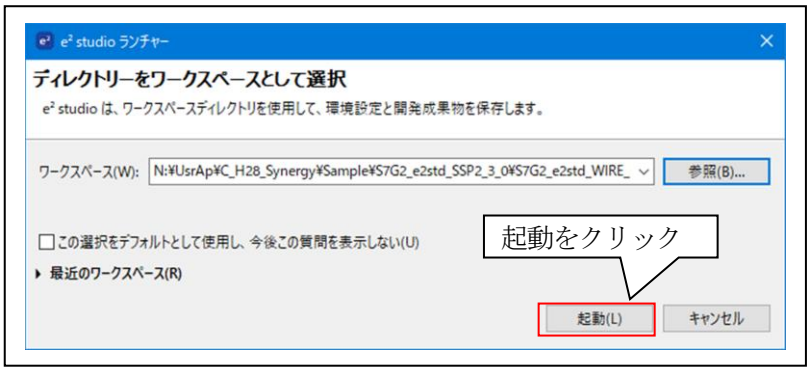
[My port]: 50001

6．「S5D9_e2std_WIRE_NetX_1」のワークスペースを指定する。

6－1．ワークスペース名の指定







6－2．プロジェクトのインポート

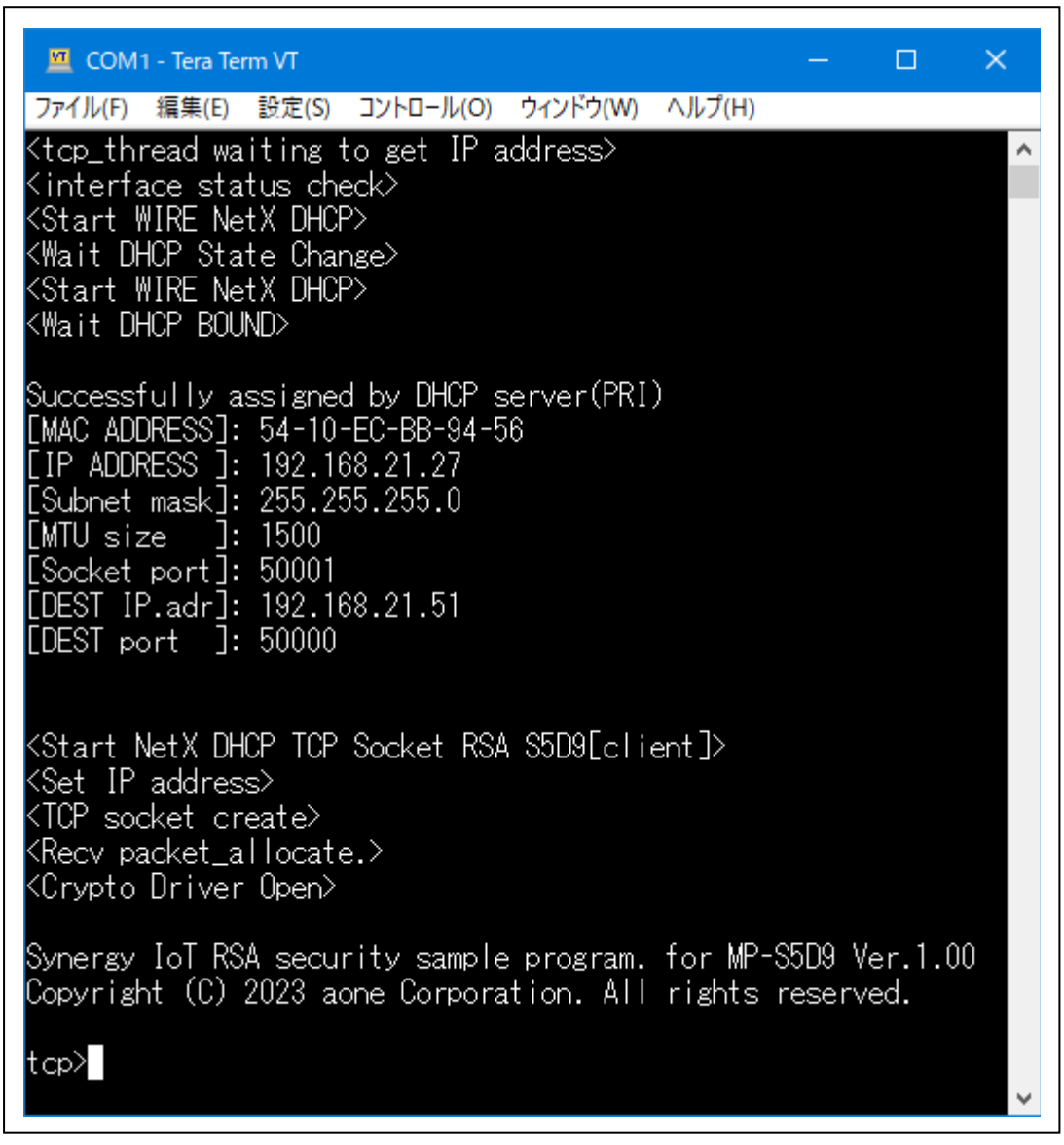
☆詳細操作は「e2studio_synergy_Import.pdf」の２項を参照して下さい。

７．デバッグ操作

７－１．デバッグ構成の設定
☆詳細操作は「e2studio_synergy_Import.pdf」の３－１項を参照して下さい。

７－２．デバッグの開始
☆詳細操作は「e2studio_synergy_Import.pdf」の３－２項を参照して下さい。

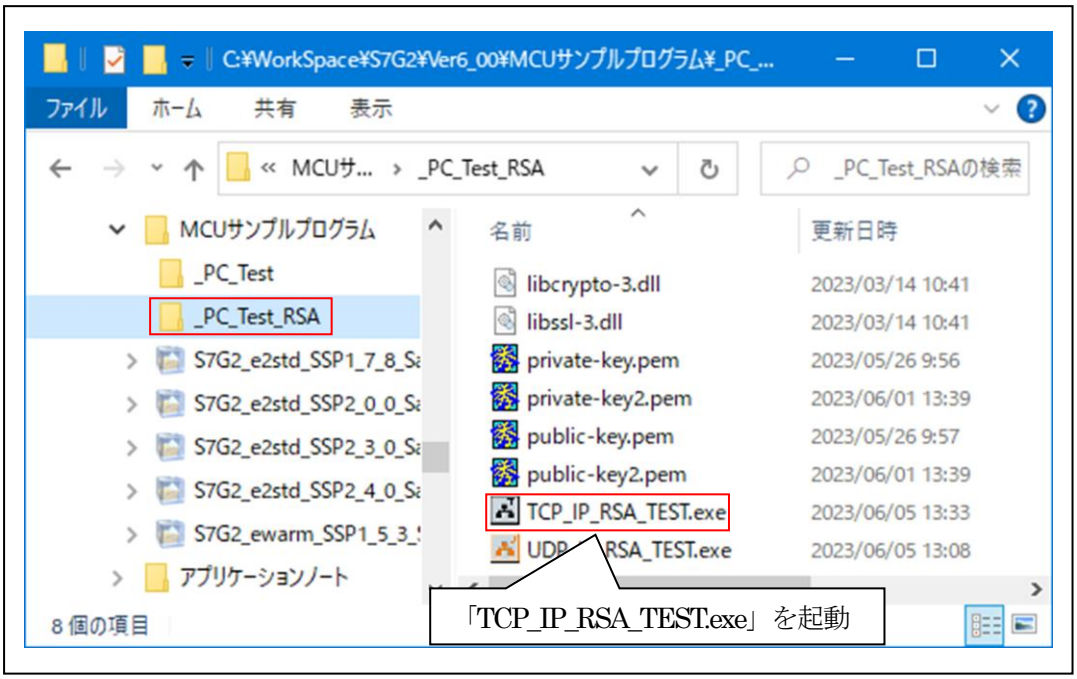
＜WIRE_DHCP_TCP_RSA 実行画面＞



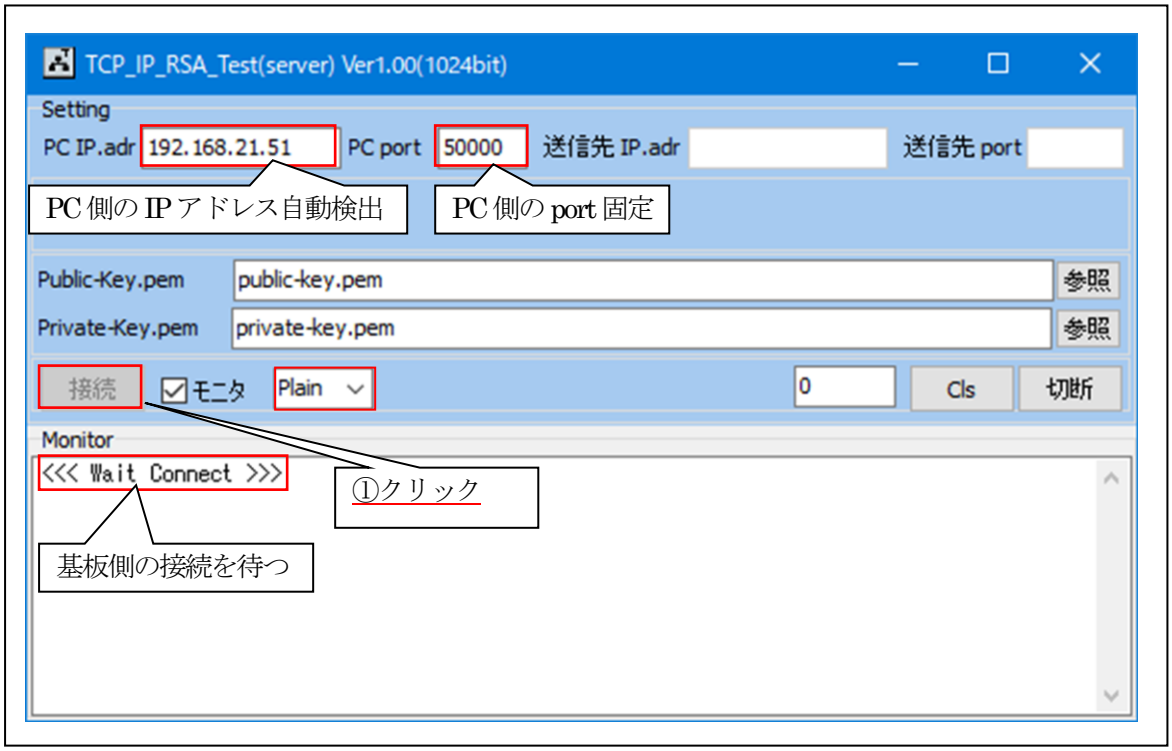
７－３．Windows PC 側のテスト用プログラムで動作確認（PLAIN（平文）モード）

１）「TCP_IP_RSA_TEST」を起動する。

プログラム場所【ご購入 CD¥MCU サンプルプログラム¥_PC_Test_RSA】



２）「TCP_IP_RSA_TEST」の各項目を設定して「基板」側からの「接続」を待つ。



3）「基板」側の各項目の確認と設定。

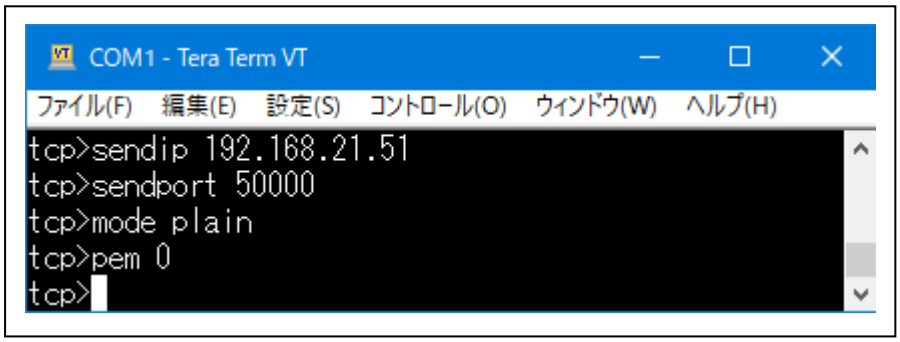
①PC 側の IP アドレスを設定する。
ex) sendip 192.168.21.51⏎

②PC 側のポート番号を設定する。
ex) sendport 50000⏎

③「plain」（平文）モードを指定する。
ex) mode plain⏎

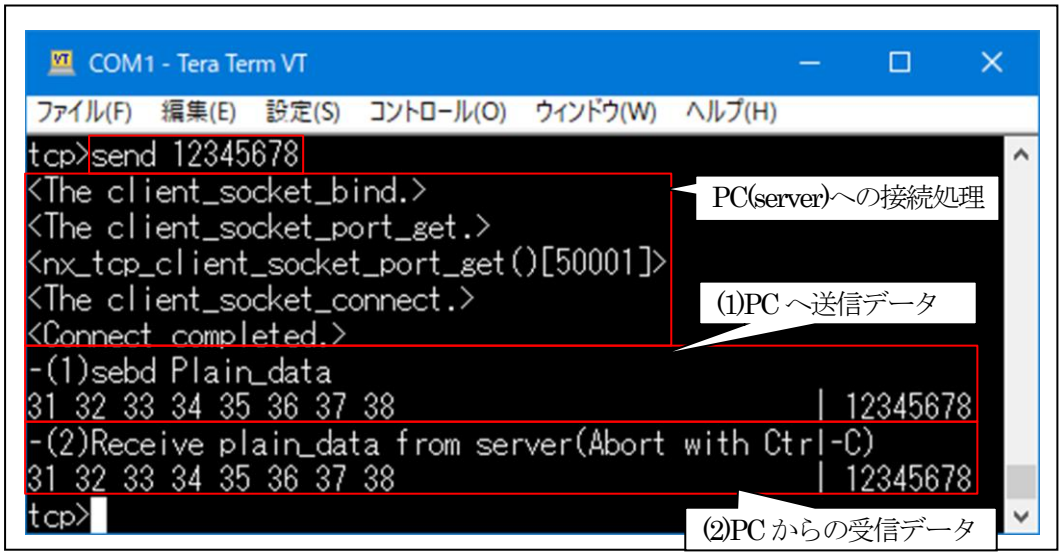
④「pem」（PEM）タイプを指定する。
ex) pem 0⏎

⑤実行画面

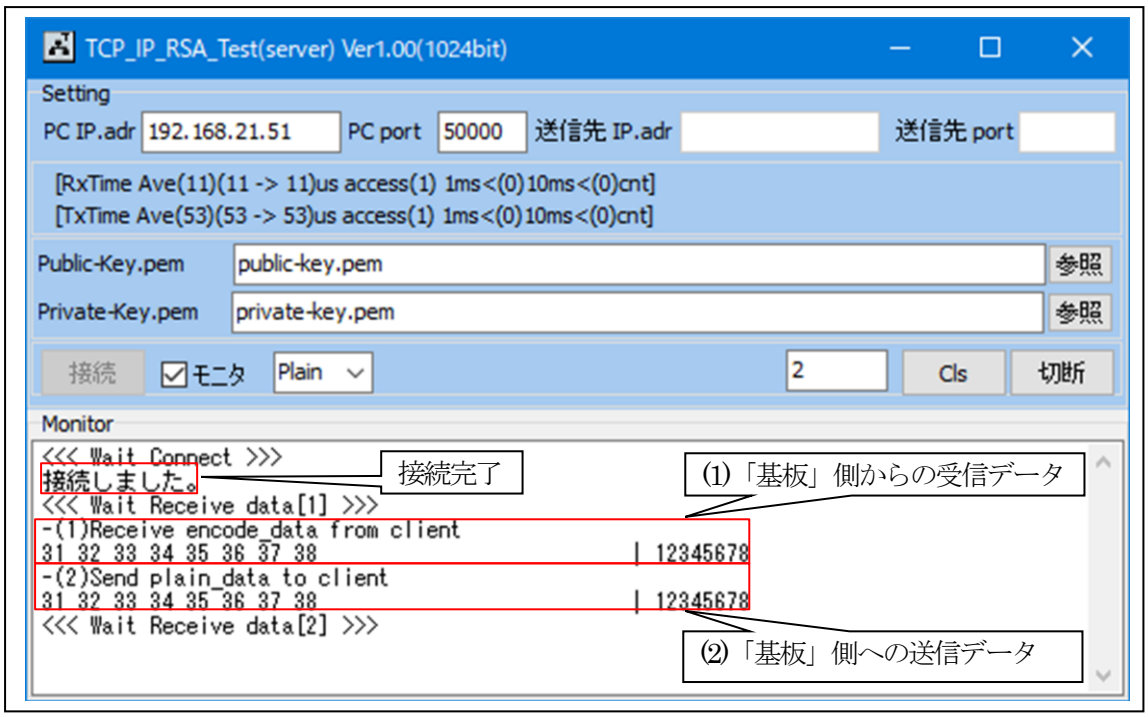


4）「基板」側から「PC」(server) 側へ平文テキストを送信する。
①テキスト文を送信する。（未接続の場合は、接続を実行）

ex) send 12345678

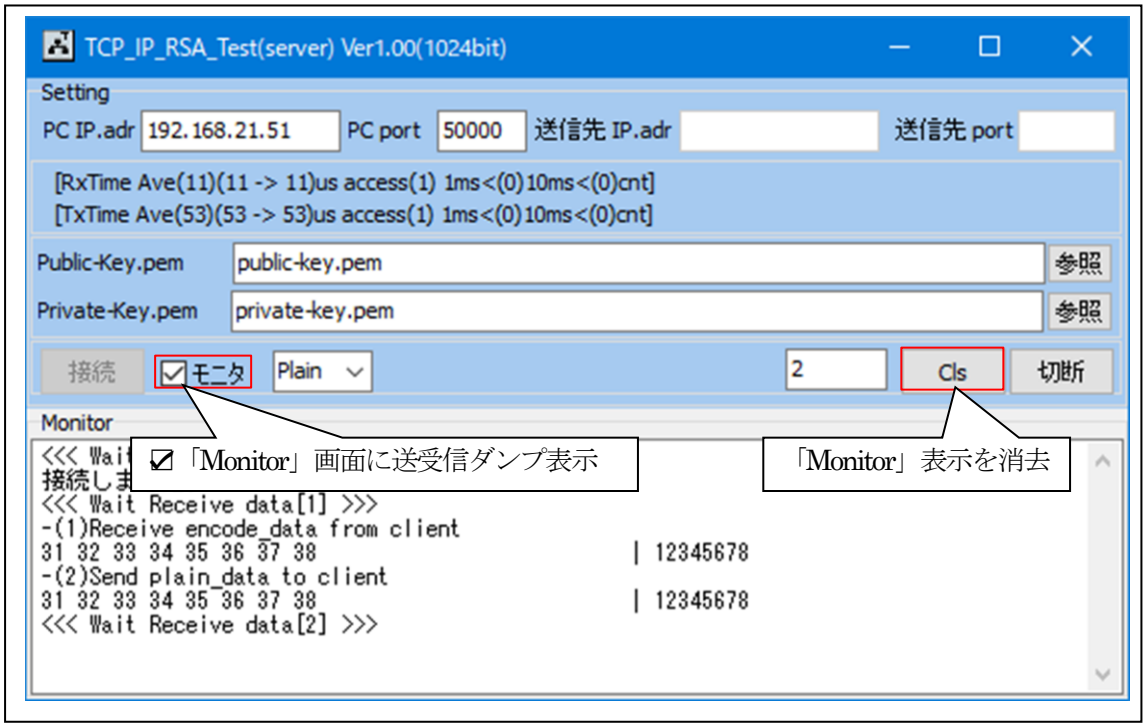


5）「TCP_IP_RSA_TEST」側の送受信を確認する。

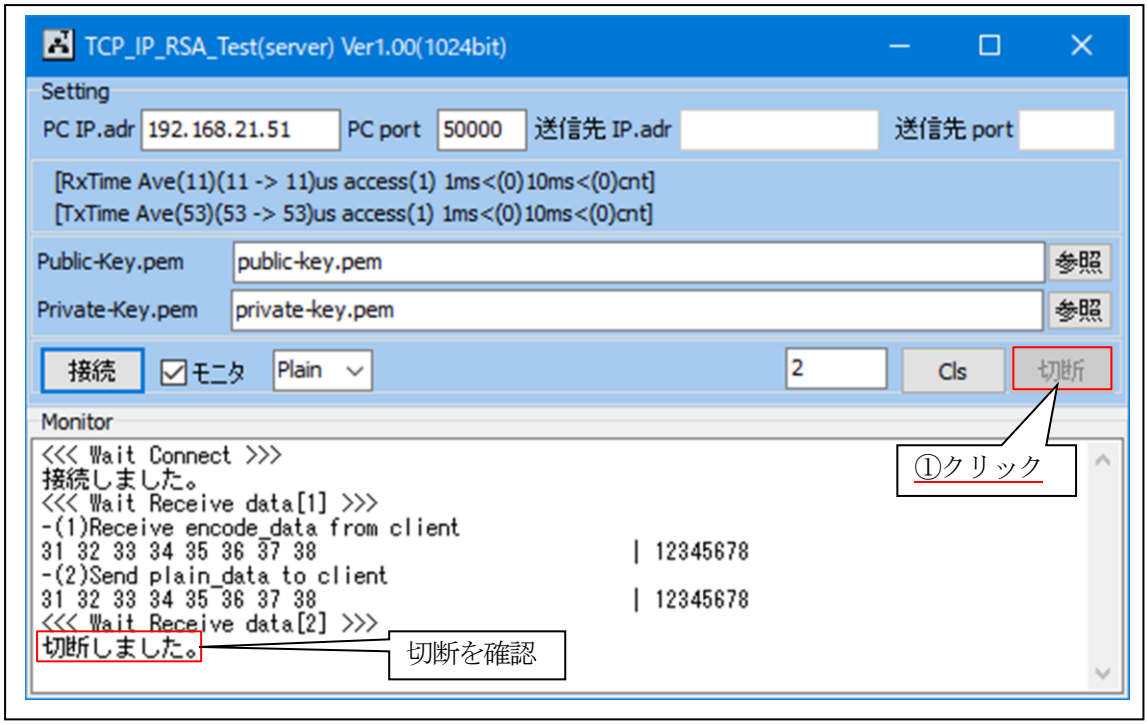


【Error 表示】
・受信時の接続失敗「"error!! client_socket_connect()[error code]"」
・受信異常「" error!! recvfrom()[error code]"」
・送信異常「"error!! sendto()[error code]"」

6）「TCP_IP_RSA_TEST」その他の操作



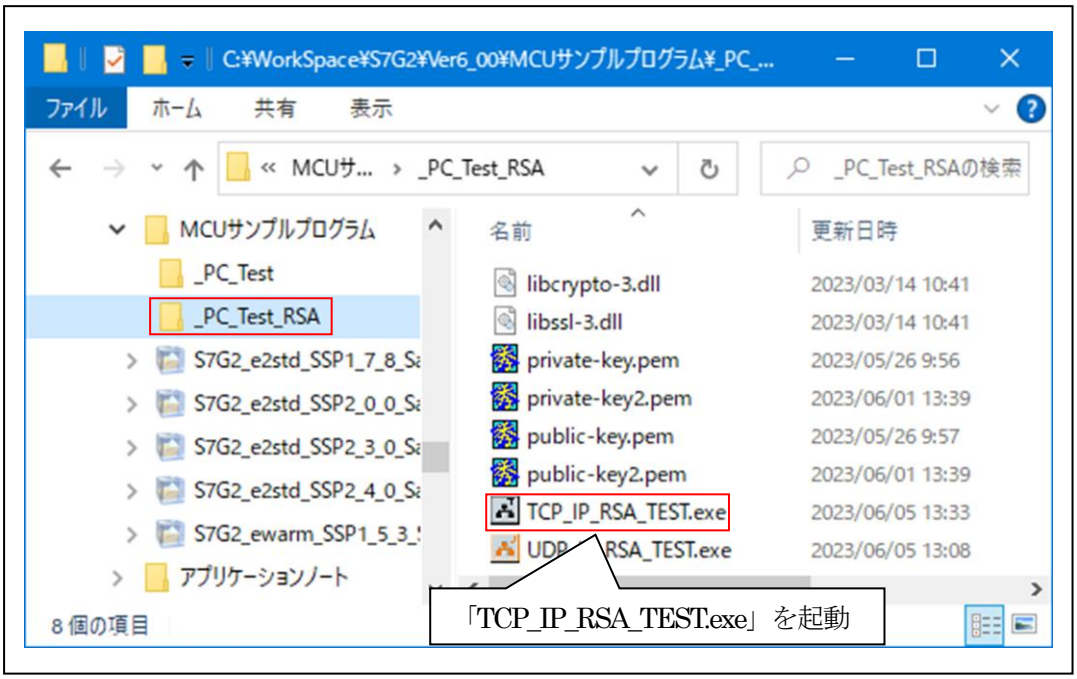
7）TCP_IP-Port を「切断」する。



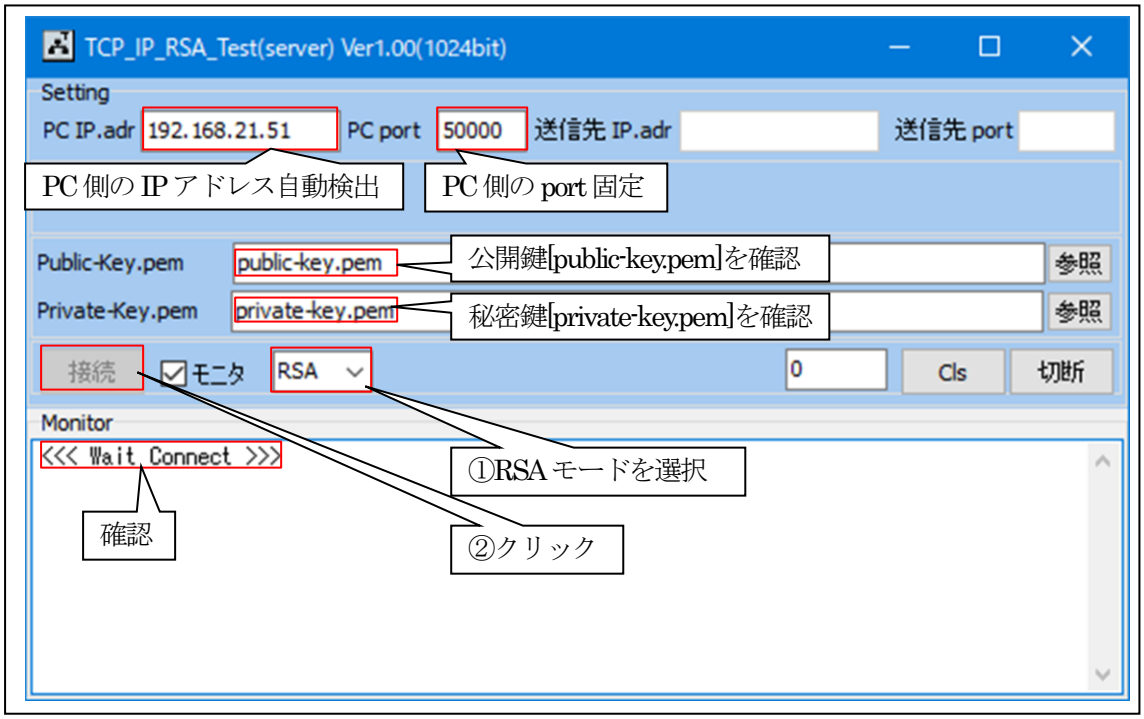
7－4．Windows PC 側のテスト用プログラムで動作確認（RSA モード/public-key.pem タイプ）

1）「TCP_IP_RSA_TEST」を起動する。

プログラム場所【ご購入 CD¥MCU サンプルプログラム¥_PC_Test_RSA】



2）「TCP_IP_RSA_TEST」の各項目を設定して「基板」側からの「接続」を待つ。



3）「基板」側の各項目の確認と設定。

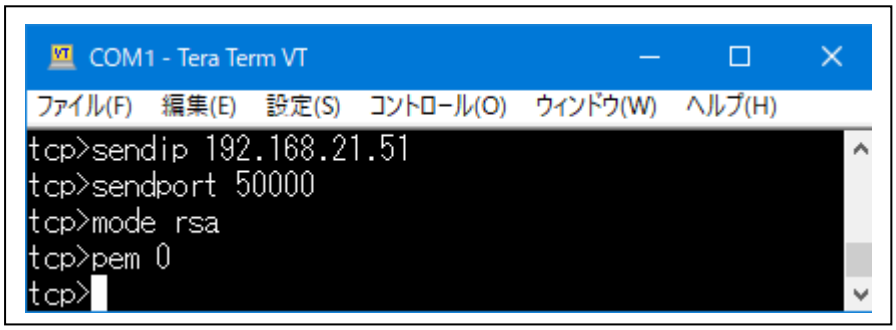
①PC 側の IP アドレスを設定する。
ex) sendip 192.168.21.51⏎

②PC 側のポート番号を設定する。
ex) sendport 50000⏎

③「rsa」（RSA）モードを指定する。
ex) mode rsa⏎

④「pem」（publik-key.pem）タイプを指定する。
ex) pem 0⏎

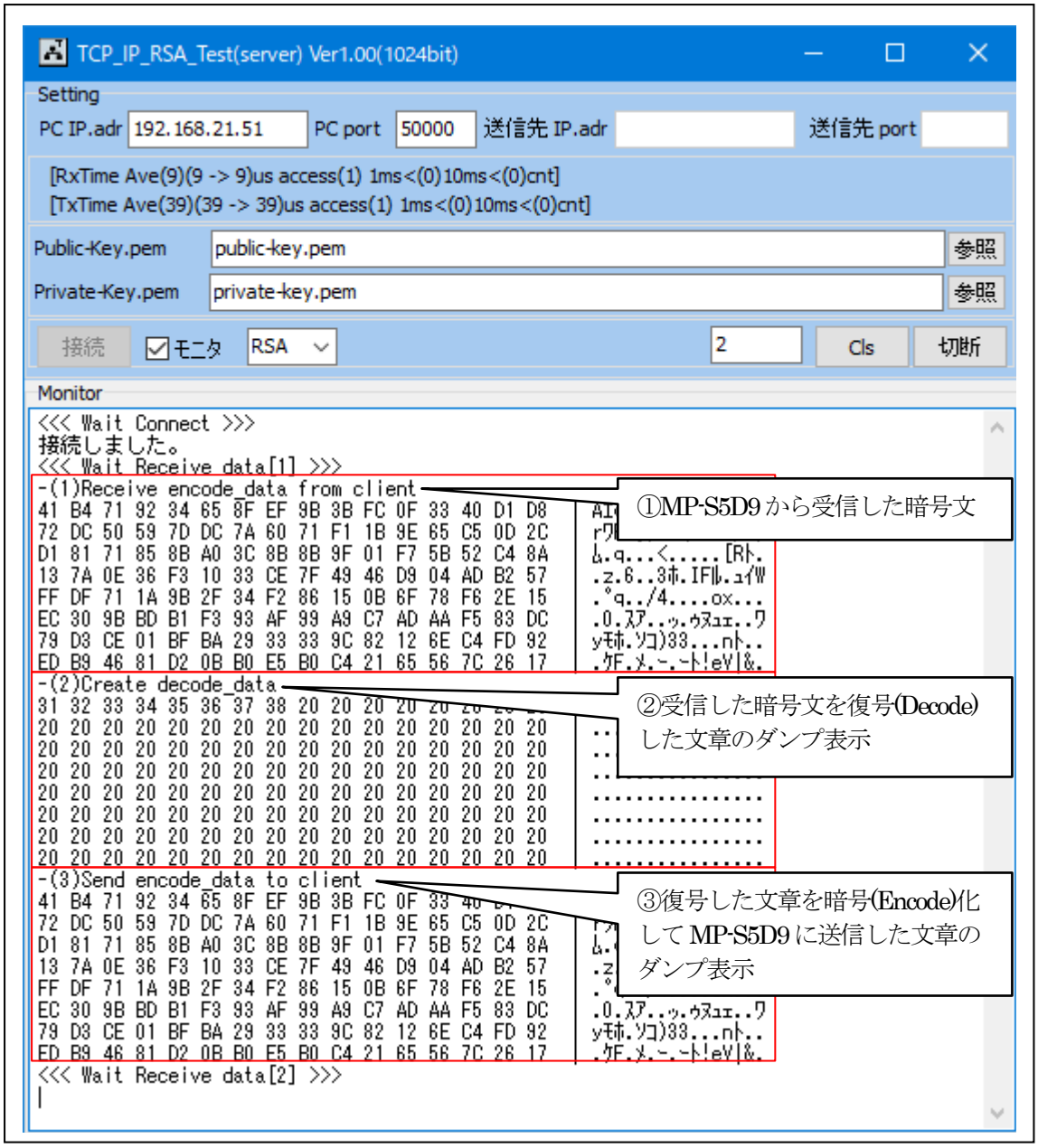
⑤実行画面



4）「基板」側から「PC」(server)側へRSA暗号テキストを送信する。
①テキスト文を送信する。（未接続の場合は、接続を実行）

ex) send 12345678

5）「TCP_IP_RSA_TEST」側の送受信を確認する。



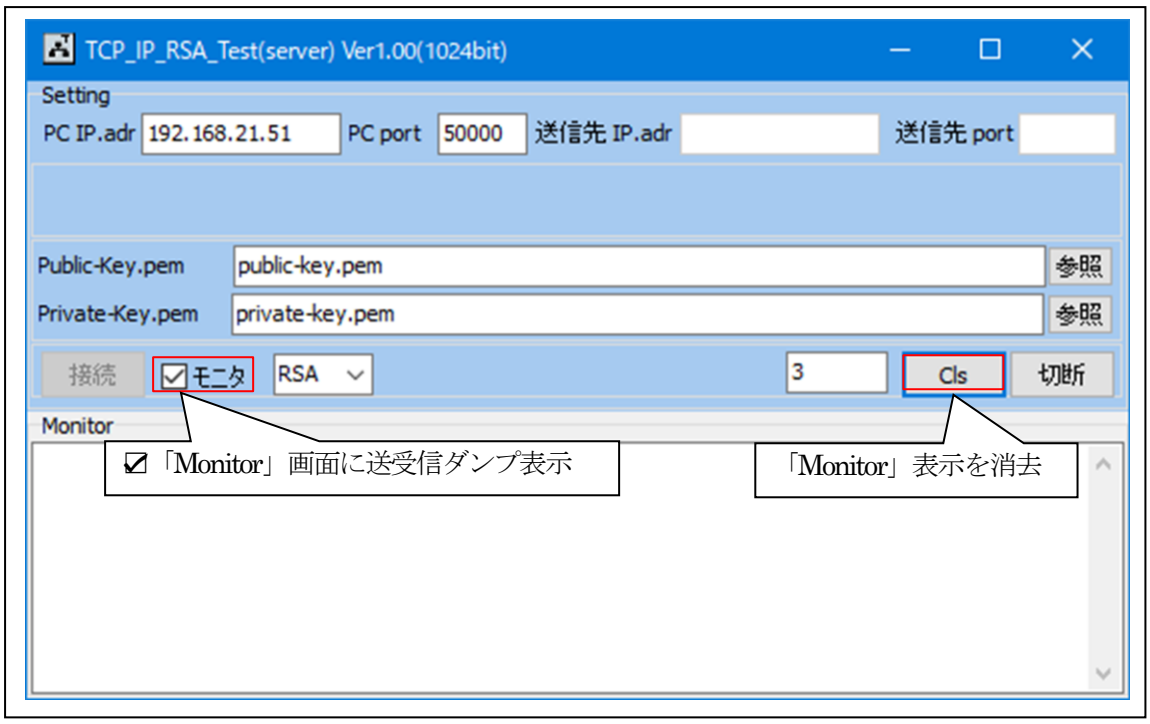
【Error 表示】

・受信時の接続失敗「"error!! client_socket_connect()[error code]"」

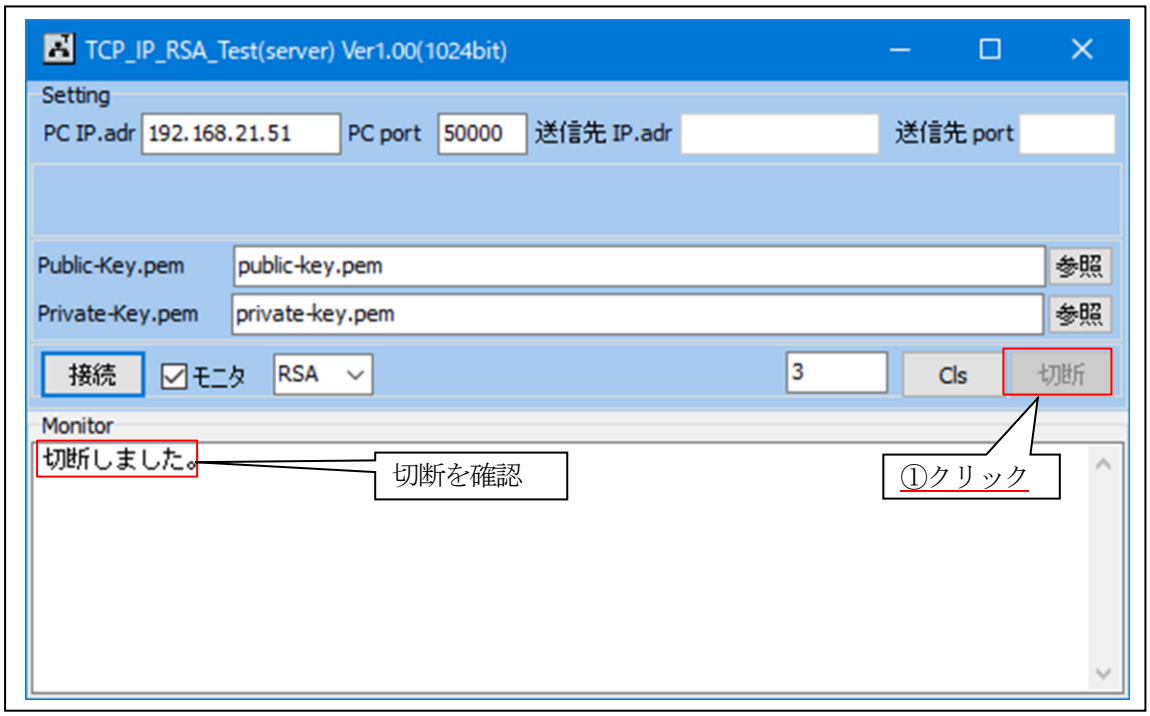
・受信異常「" error!! recvfrom()[error code]"」

・送信異常「"error!! sendto()[error code]"」

6）「TCP_IP_RSA_TEST」その他の操作



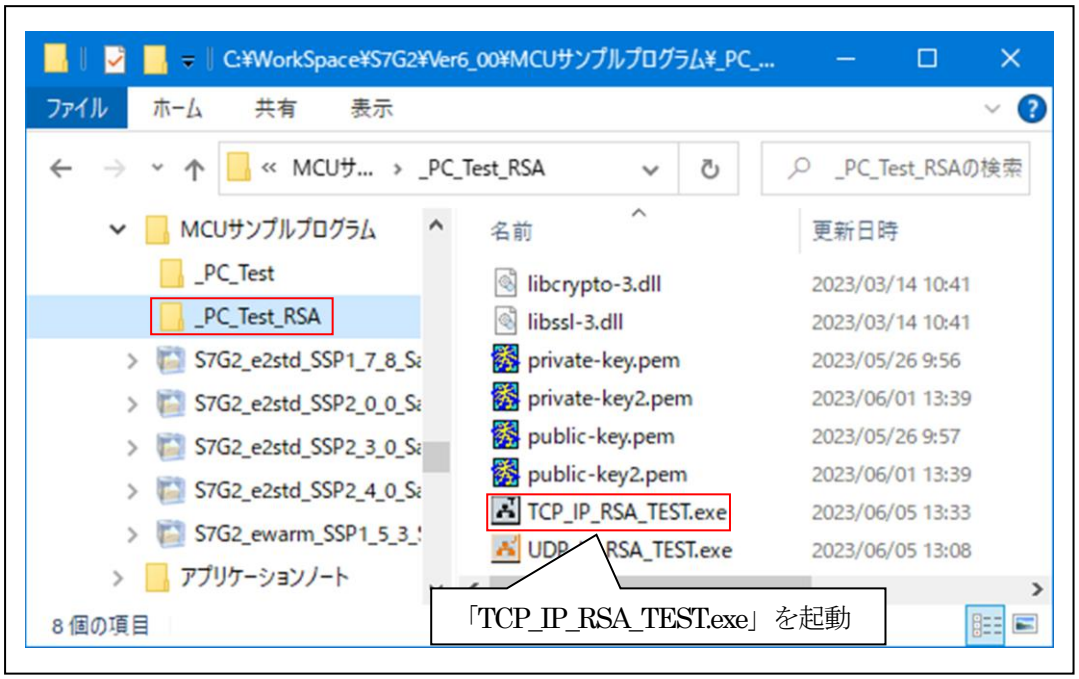
7）TCP_IP-Port を「切断」する。



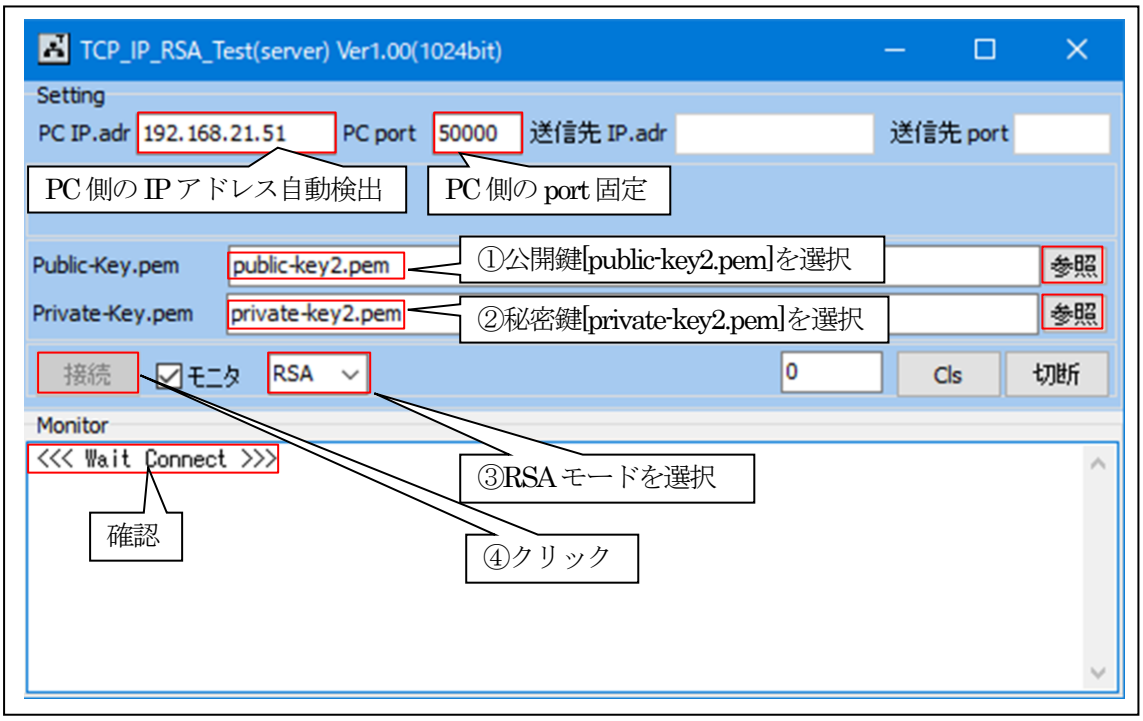
７－５．Windows PC 側のテスト用プログラムで動作確認（RSA モード/public-key2.pem タイプ）

１）「TCP_IP_RSA_TEST」を起動する。

プログラム場所【ご購入 CD¥MCU サンプルプログラム¥_PC_Test】



２）「TCP_IP_RSA_TEST」の各項目を設定して「基板」側からの「接続」を待つ。



3）「基板」側の各項目の確認と設定。

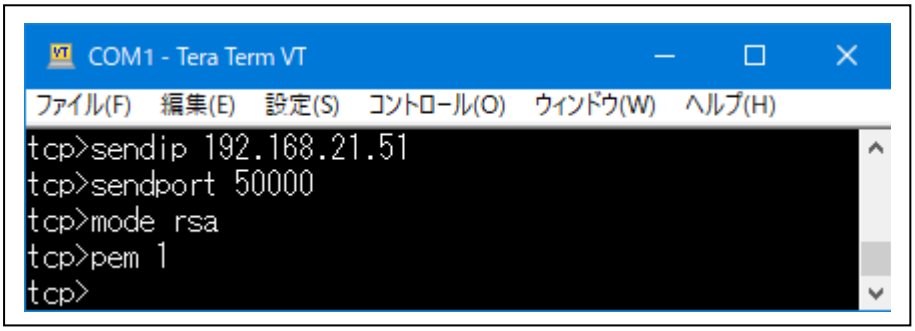
①PC 側の IP アドレスを設定する。
ex) sendip 192.168.21.51⏎

②PC 側のポート番号を設定する。
ex) sendport 50000⏎

③「rsa」（RSA）モードを指定する。
ex) mode rsa⏎

④「pem」（public-key2.pem）タイプを指定する。
ex) pem 1⏎

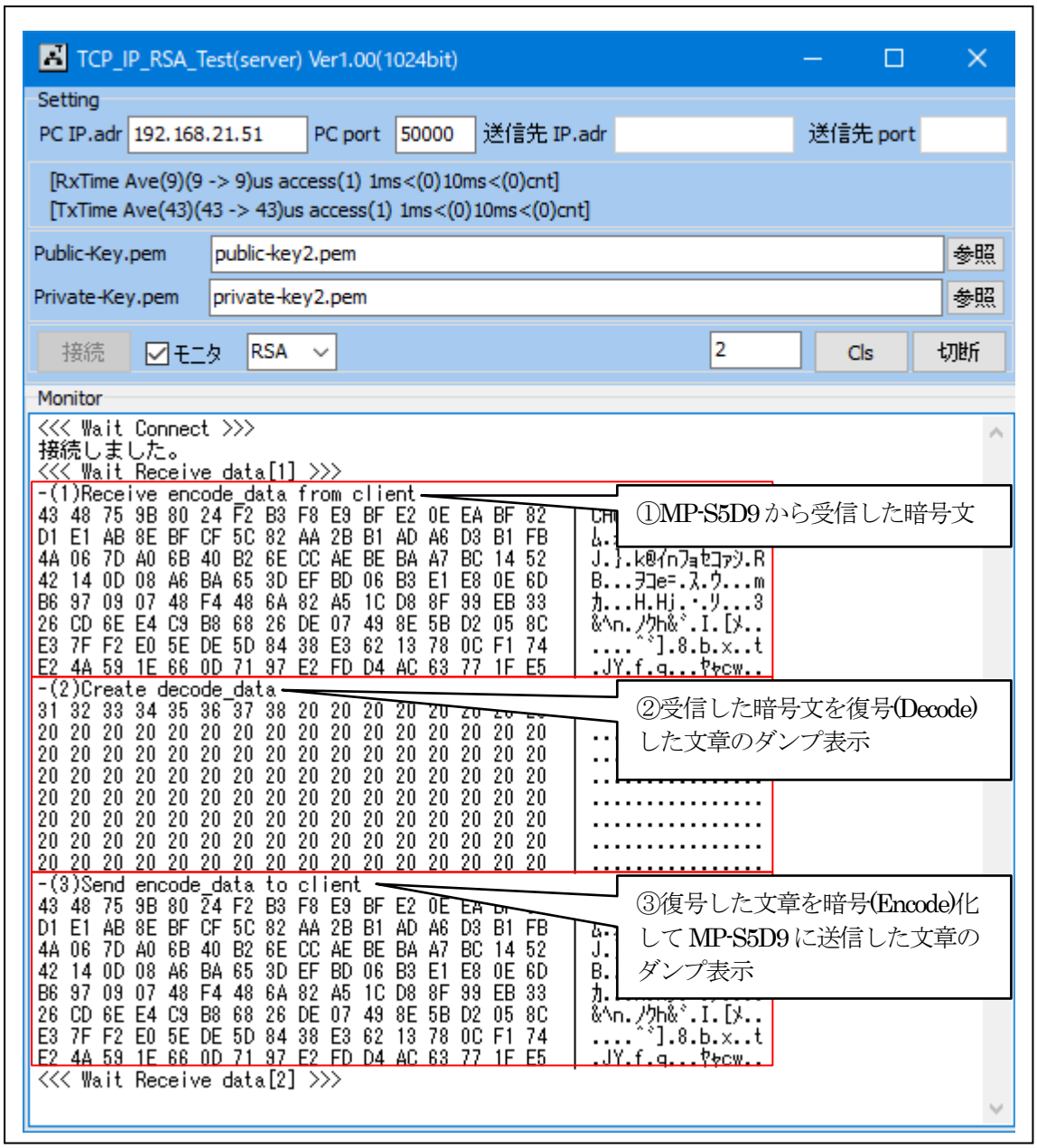
⑤実行画面



4）「基板」側から「PC」(server)側へRSA暗号テキストを送信する。
①テキスト文を送信する。（未接続の場合は、接続を実行）

ex) send 12345678

5）「TCP_IP_RSA_TEST」側の送受信を確認する。



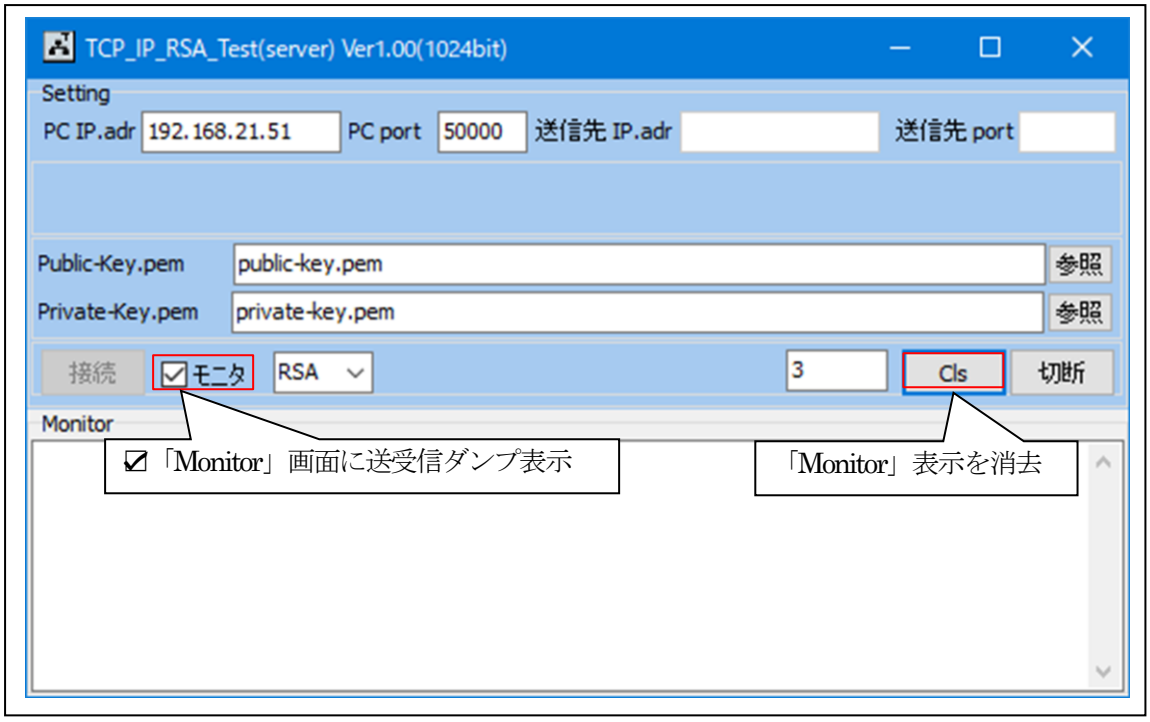
【Error 表示】

・受信時の接続失敗「"error!! client_socket_connect()[error code]"」

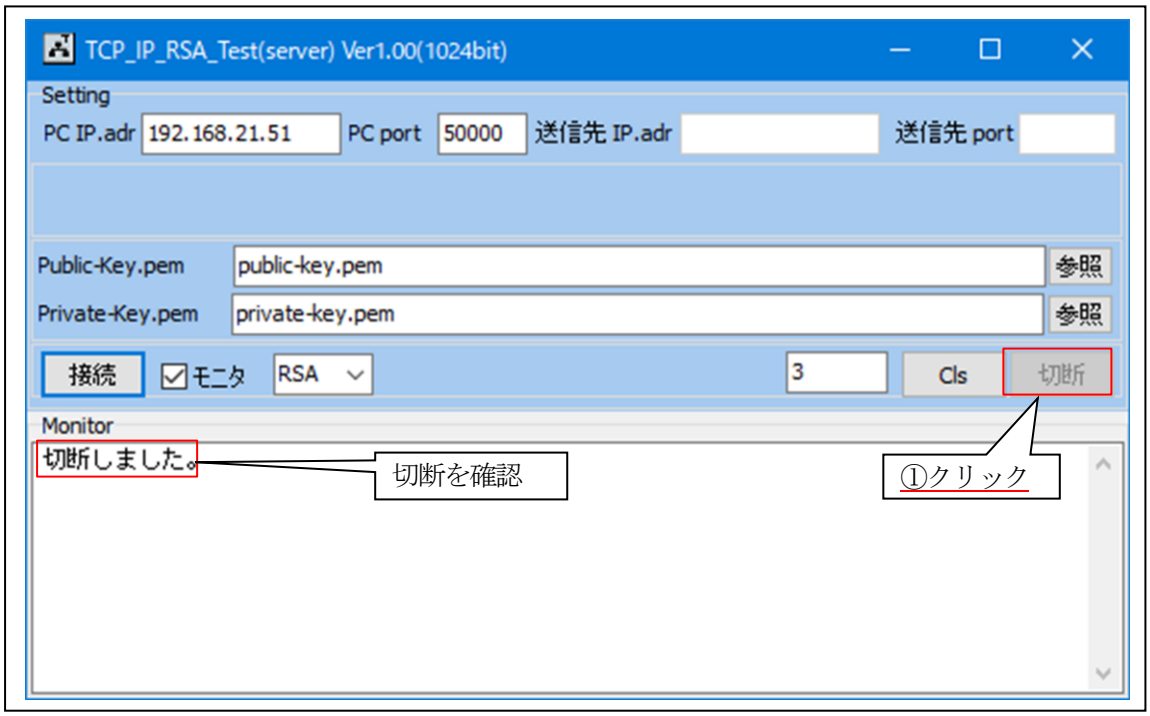
・受信異常「" error!! recvfrom()[error code]"」

・送信異常「"error!! sendto()[error code]"」

6）「TCP_IP_RSA_TEST」その他の操作



7）TCP_IP-Port を「切断」する。

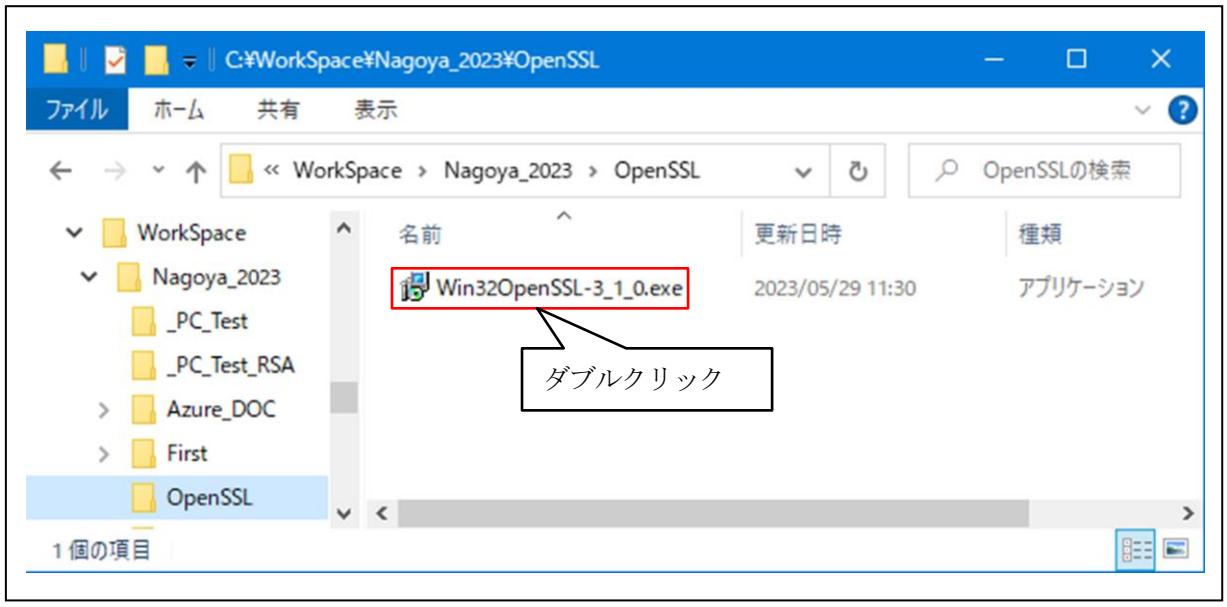


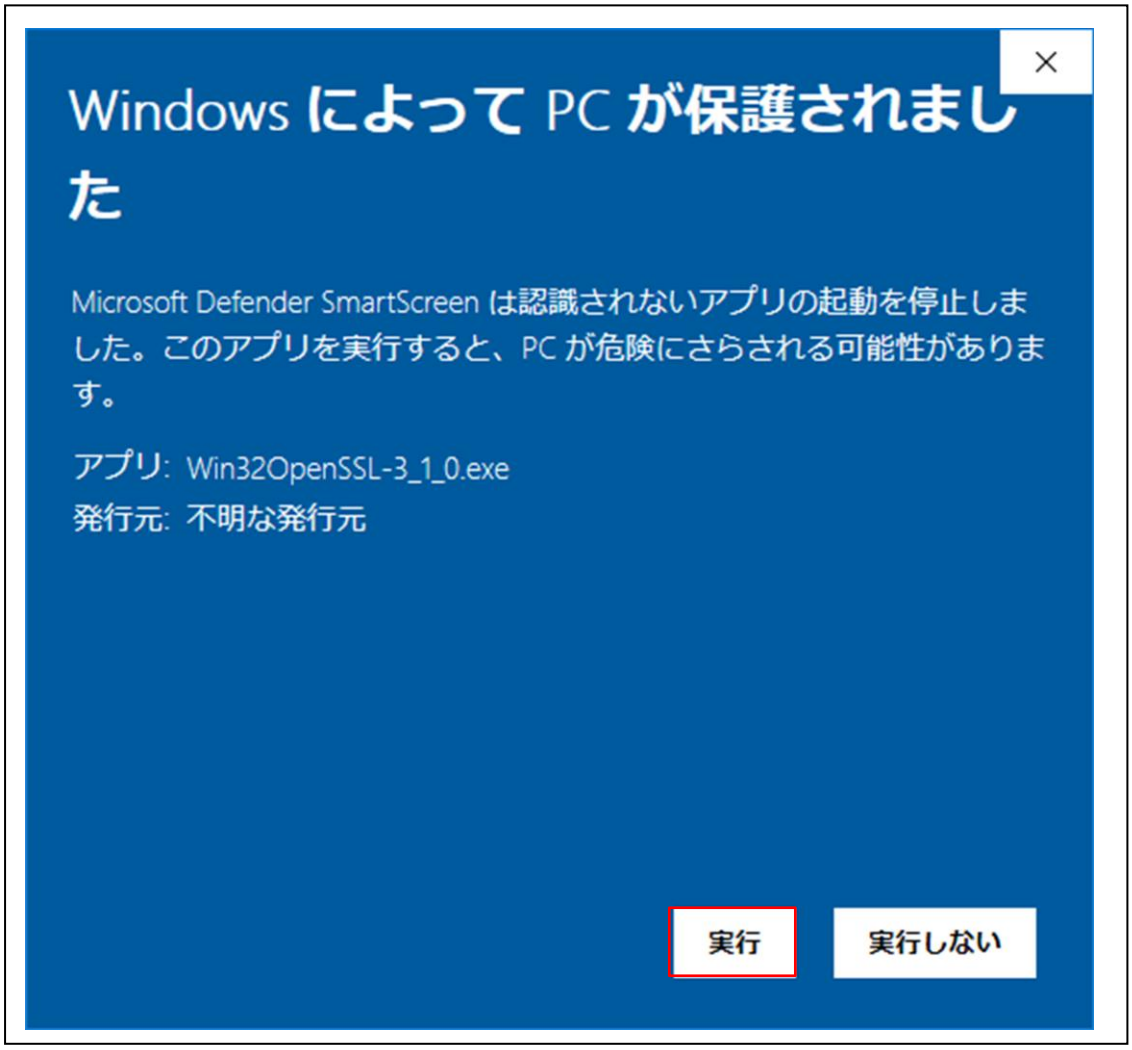
7－6．デバッグの終了

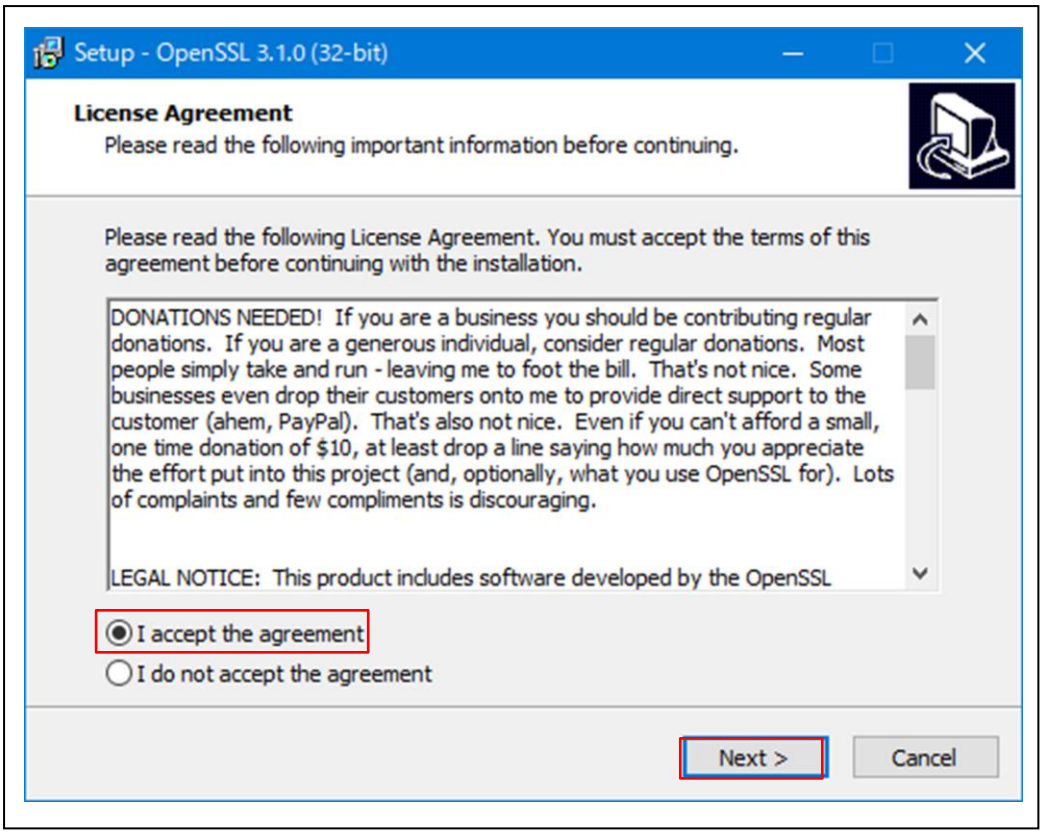
☆詳細操作は「e2studio_synergy_Import.pdf」の３－３項を参照して下さい。

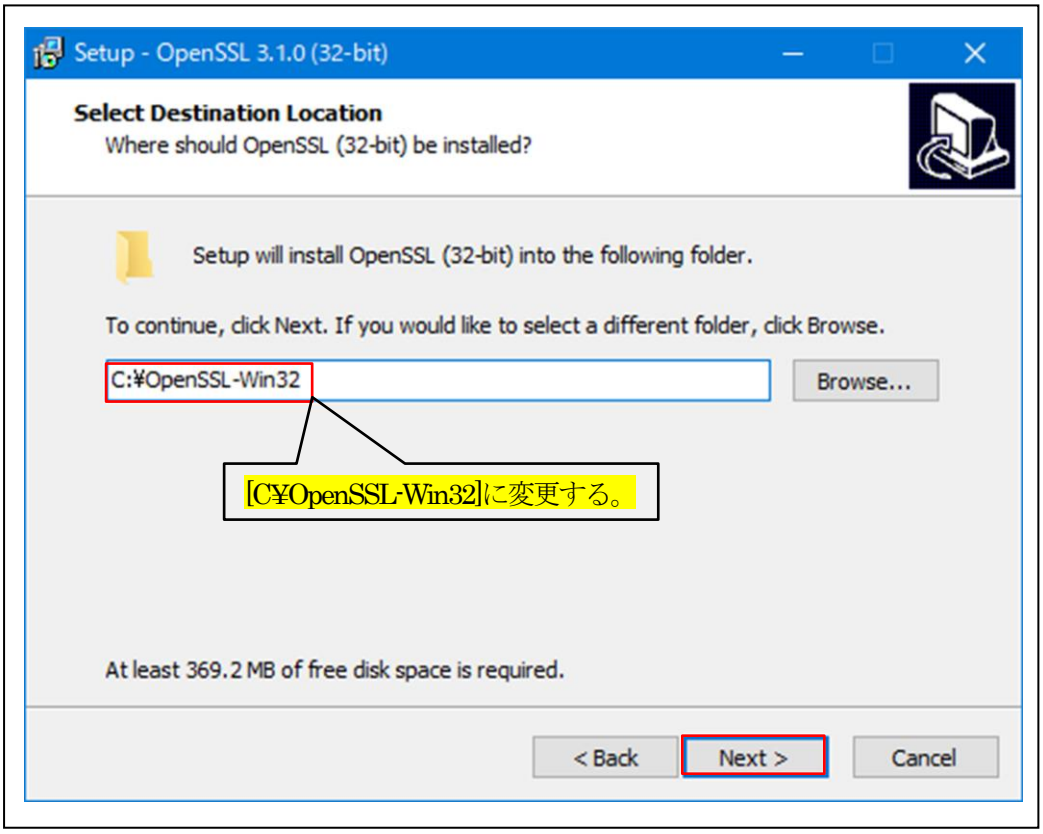
８．公開鍵と秘密鍵を新しく作成する。

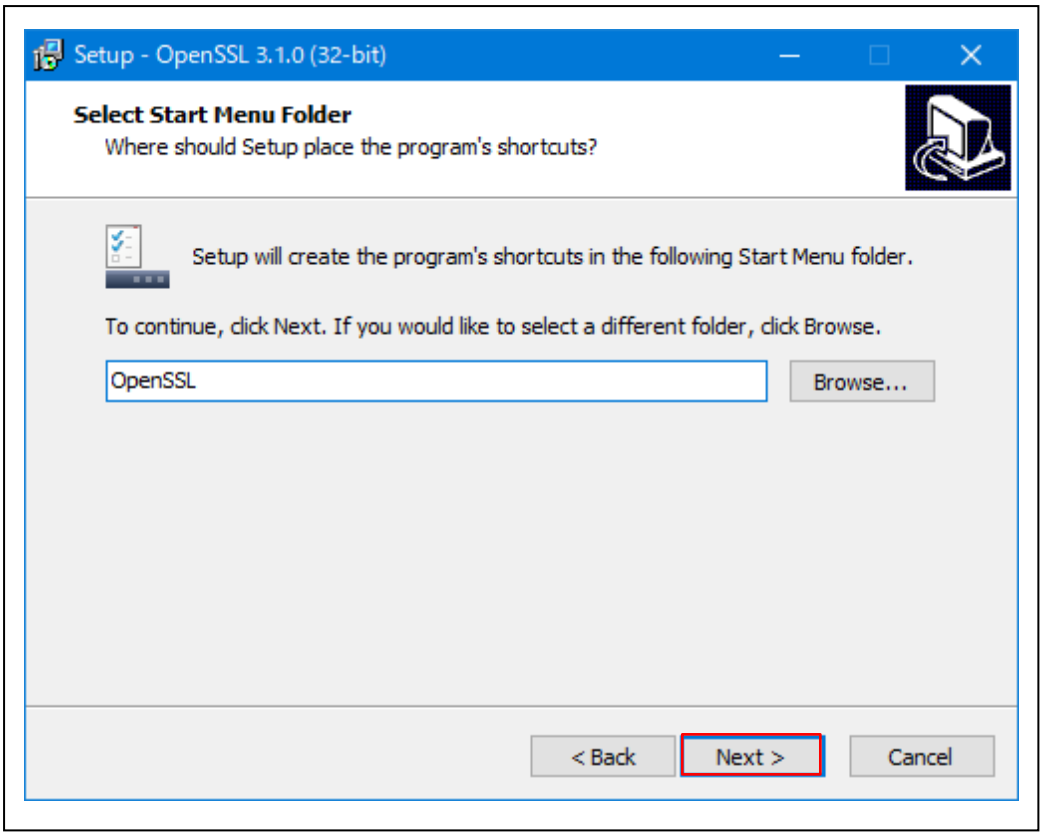
１）OpenSSL をインストールする。
　☆既にインストール済みの場合は２）へ進む

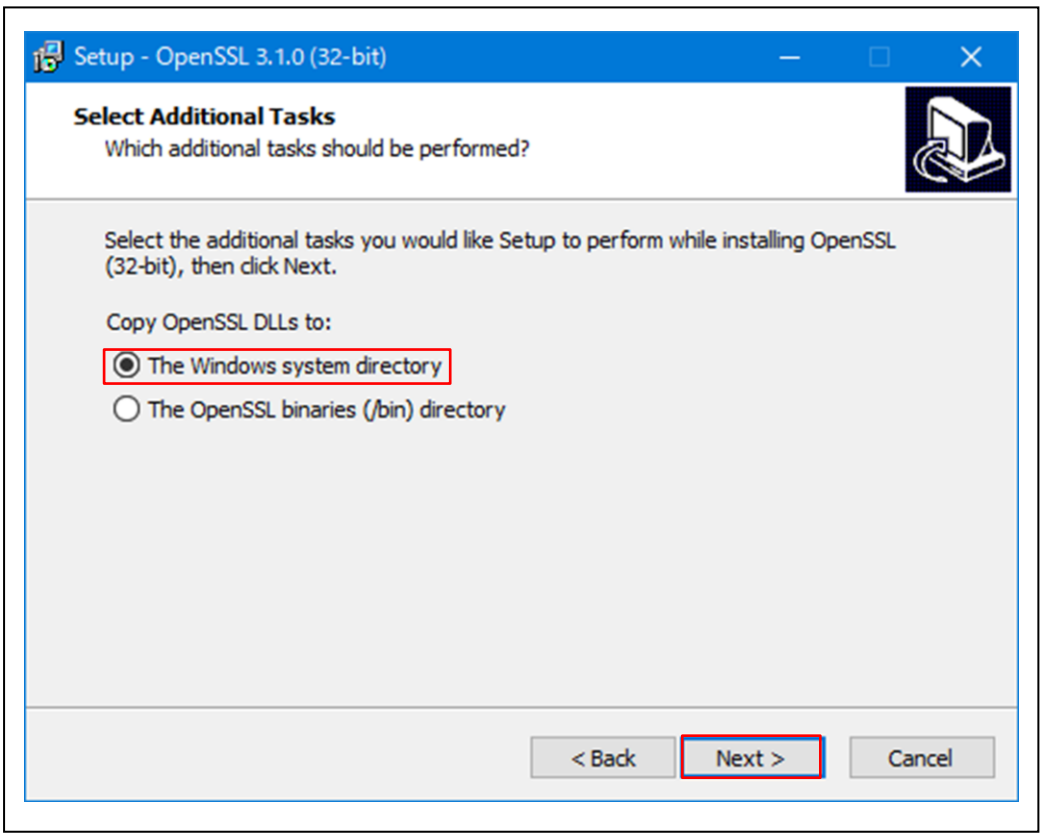


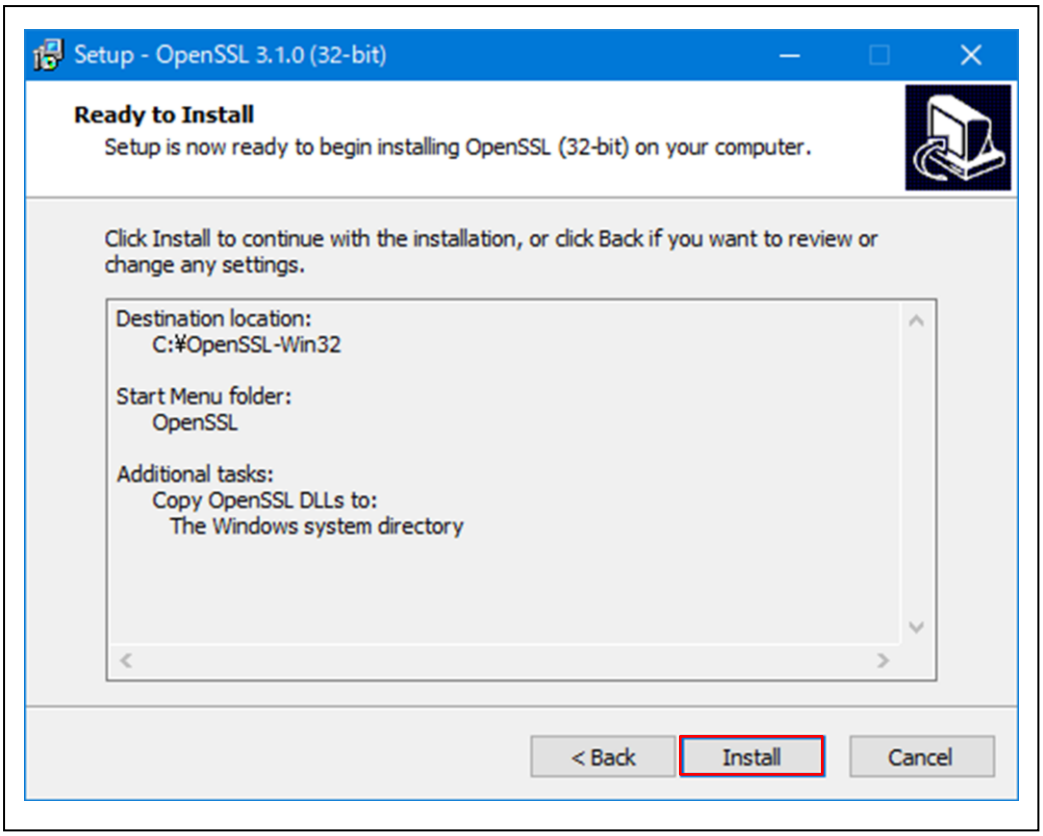


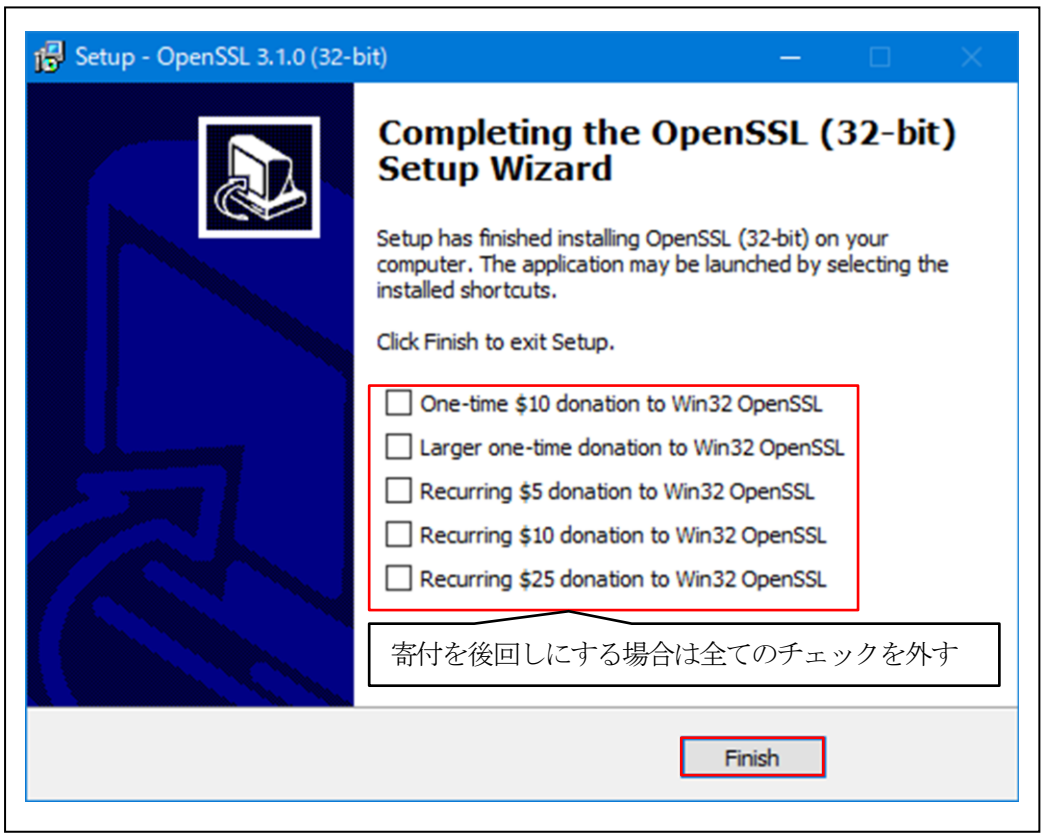

Setup - OpenSSL 3.1.0 (32-bit)
License Agreement
Please read the following important information before continuing.
Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.
DONATIONS NEEDED! If you are a business you should be contributing regular donations. If you are a generous individual, consider regular donations. Most people simply take and run - leaving me to foot the bill. That's not nice. Some businesses even drop their customers onto me to provide direct support to the customer (ahem, PayPal). That's also not nice. Even if you can't afford a small, one time donation of $10, at least drop a line saying how much you appreciate the effort put into this project (and, optionally, what you use OpenSSL for). Lots of complaints and few compliments is discouraging.
LEGAL NOTICE: This product includes software developed by the OpenSSL
I accept the agreement
I do not accept the agreement
Next >
Cancel


Setup - OpenSSL 3.1.0 (32-bit)
Select Destination Location
Where should OpenSSL (32-bit) be installed?
Setup will install OpenSSL (32-bit) into the following folder.
To continue, click Next. If you would like to select a different folder, click Browse.
C:¥OpenSSL-Win32
Browse...
[C¥OpenSSL-Win32]に変更する。
At least 369.2 MB of free disk space is required.
< Back
Next >
Cancel


Setup - OpenSSL 3.1.0 (32-bit)
Select Start Menu Folder
Where should Setup place the program's shortcuts?
Setup will create the program's shortcuts in the following Start Menu folder.
To continue, click Next. If you would like to select a different folder, click Browse.
OpenSSL
Browse...
< Back
Next >
Cancel

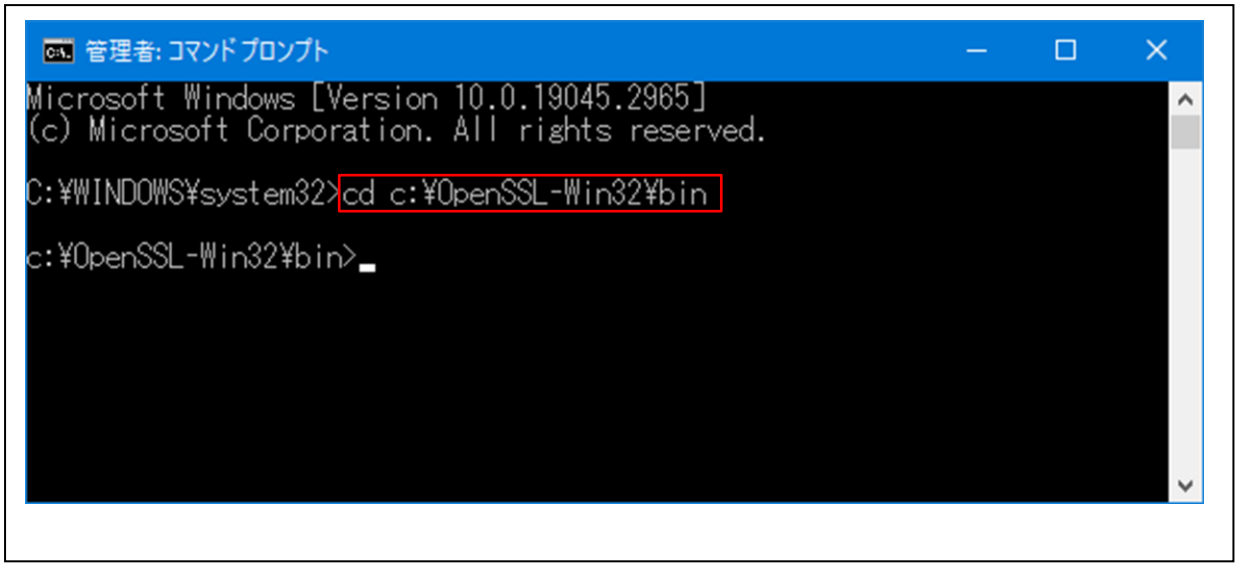

Setup - OpenSSL 3.1.0 (32-bit)
Select Additional Tasks
Which additional tasks should be performed?
Select the additional tasks you would like Setup to perform while installing OpenSSL (32-bit), then click Next.
Copy OpenSSL DLLs to:
The Windows system directory
The OpenSSL binaries (/bin) directory
< Back
Next >
Cancel





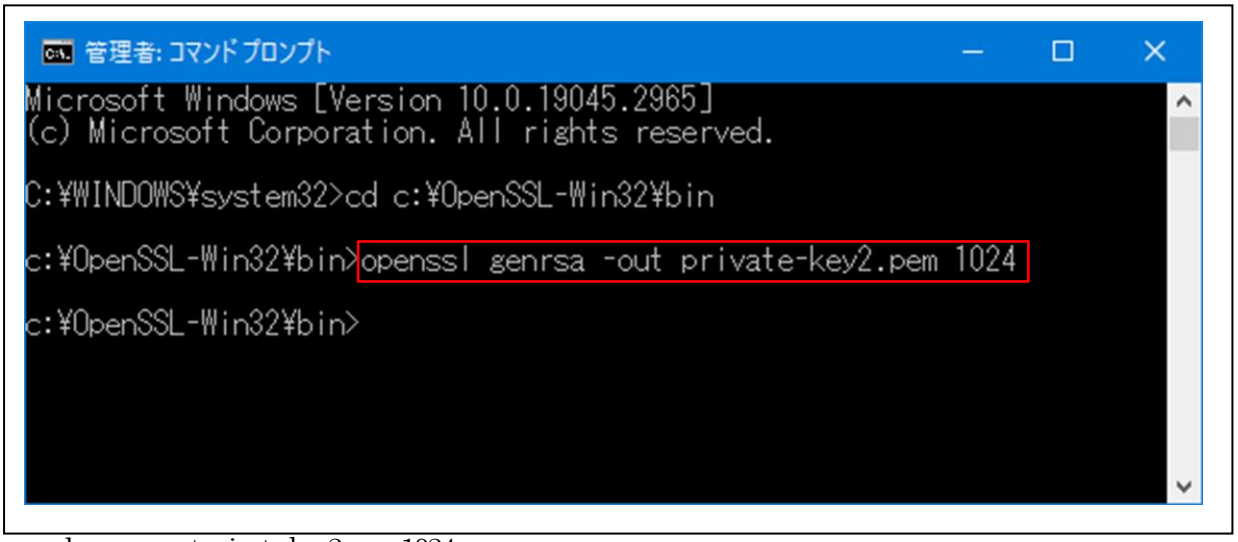
＜OpenSSL のインストール終了＞

２）OpenSSL を起動して秘密鍵と公開鍵を生成する。
２－１）コマンドプロンプトを起動して CD コマンドにて「OpenSSL-Win32¥bin」に移動する。

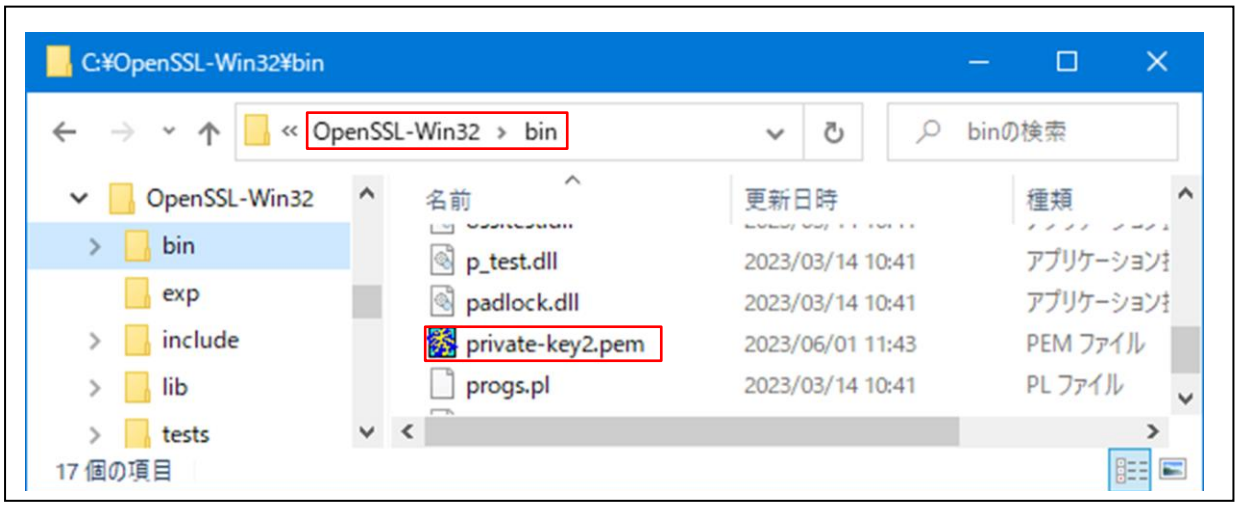


cd c:¥OpenSSL-Win32¥bin

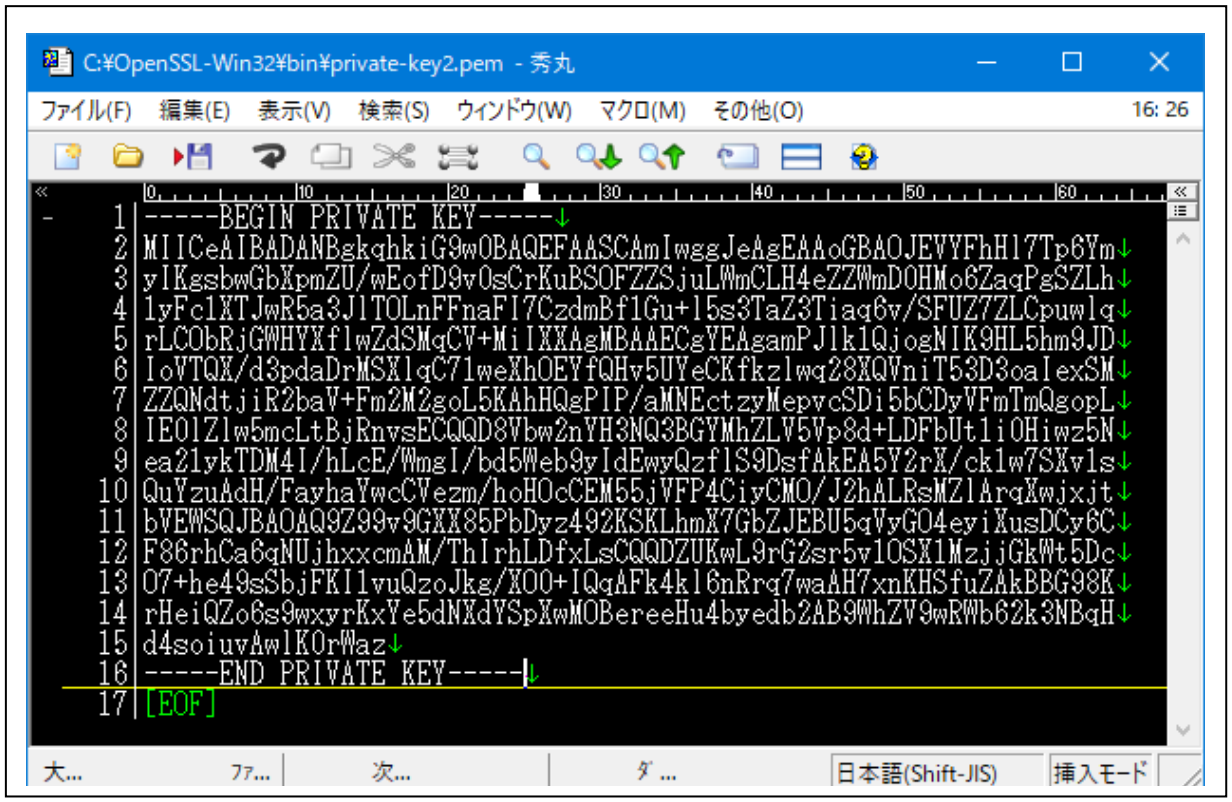
２－２）RSA 秘密鍵の生成(private-key2.pem)



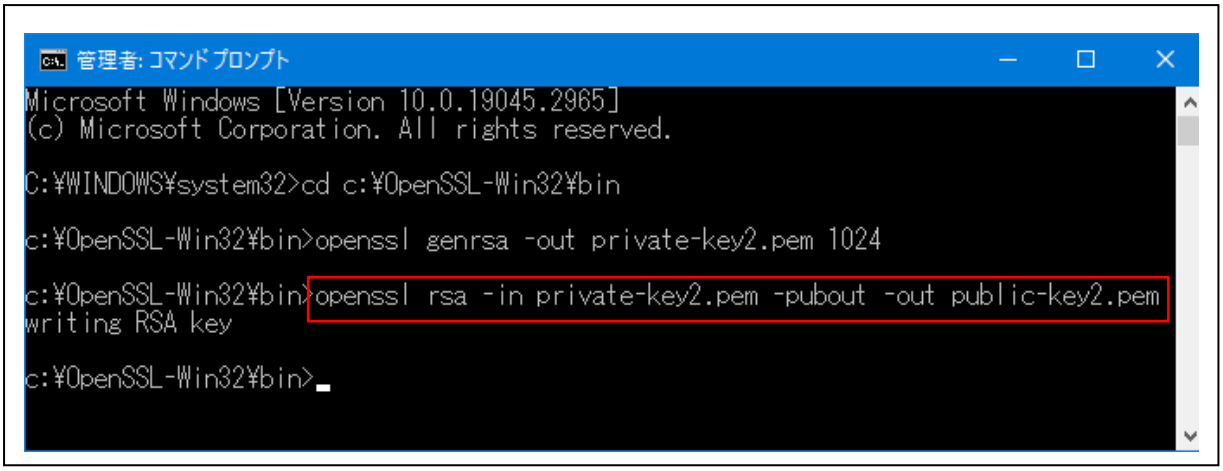
openssl genrsa -out private-key2.pem 1024



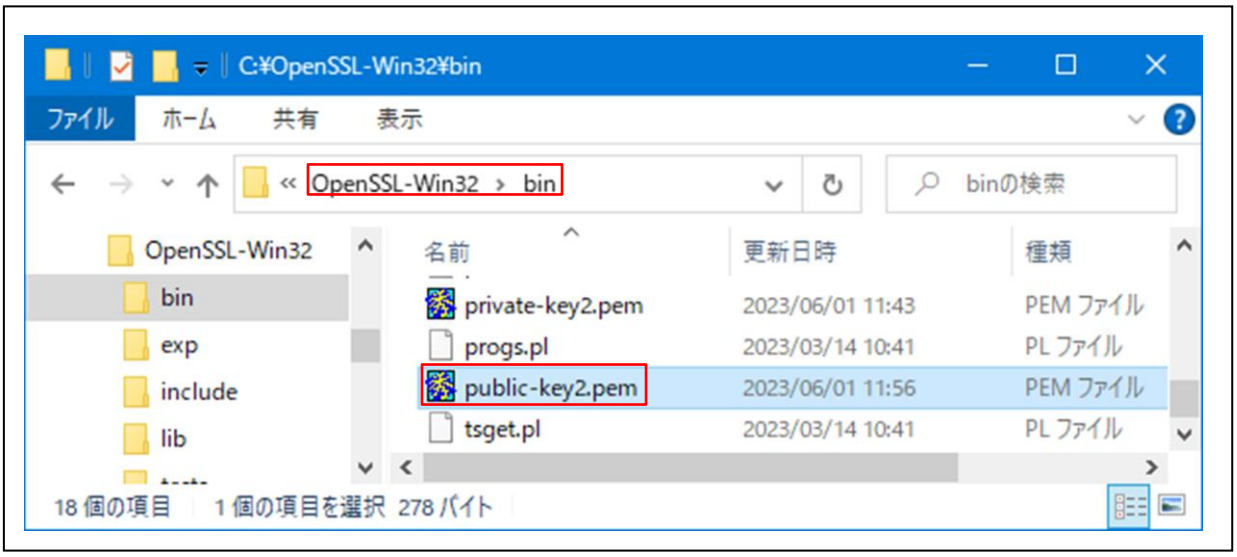
２－３）生成した秘密鍵「private-key2.pem」をメモ帳等のエディタで内容確認



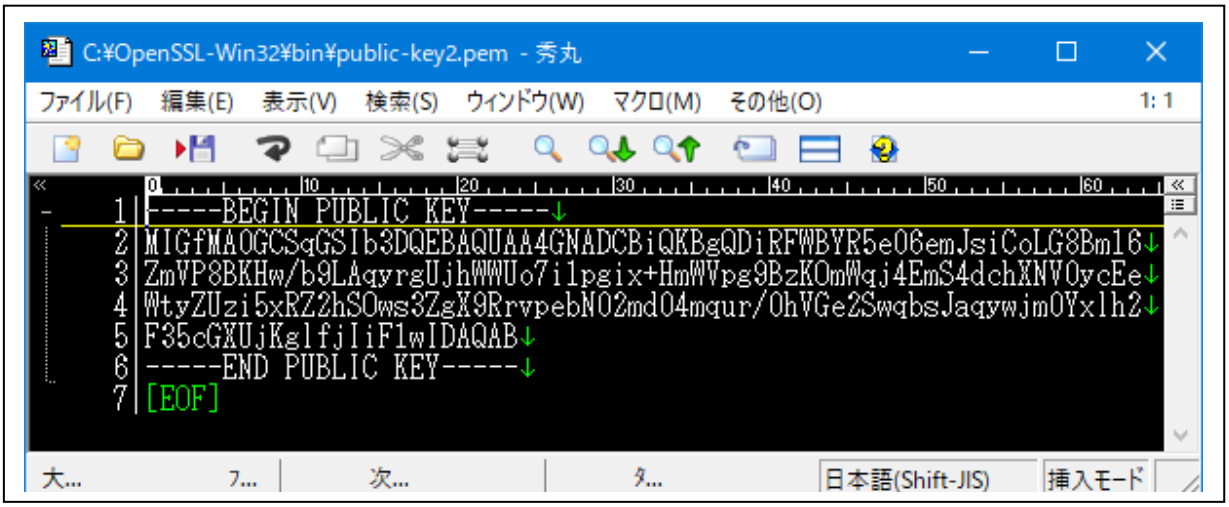
２－４）RSA 公開鍵の生成(public-key2.pem)



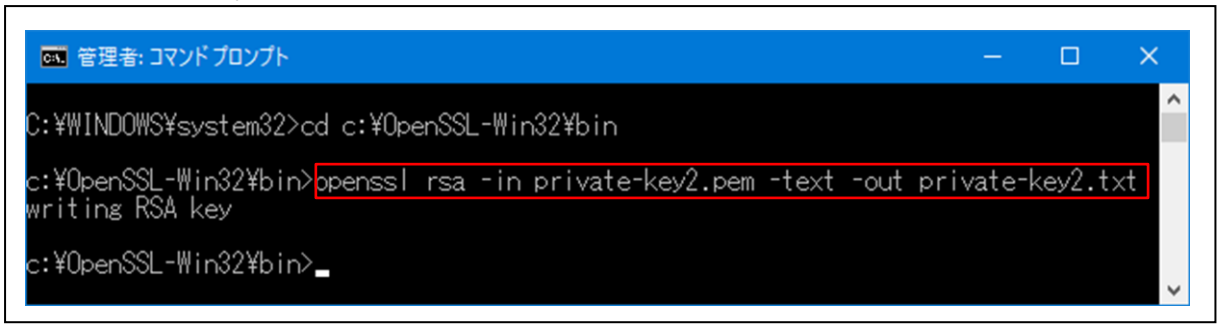
openssl rsa -in private-key2.pem -pubout -out public-key2.pem



２－５）生成した秘密鍵「public-key2.pem」をメモ帳等のエディタで内容確認

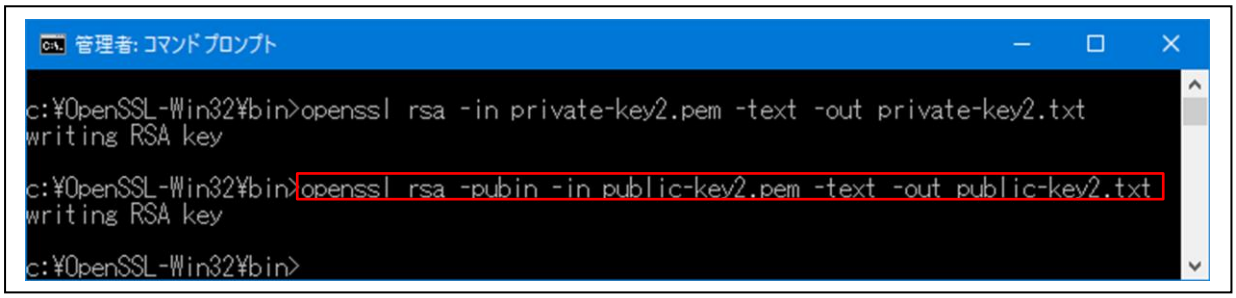


２－６）RSA 秘密鍵(private-key2.pem)を DER 形式のテキストファイルを生成



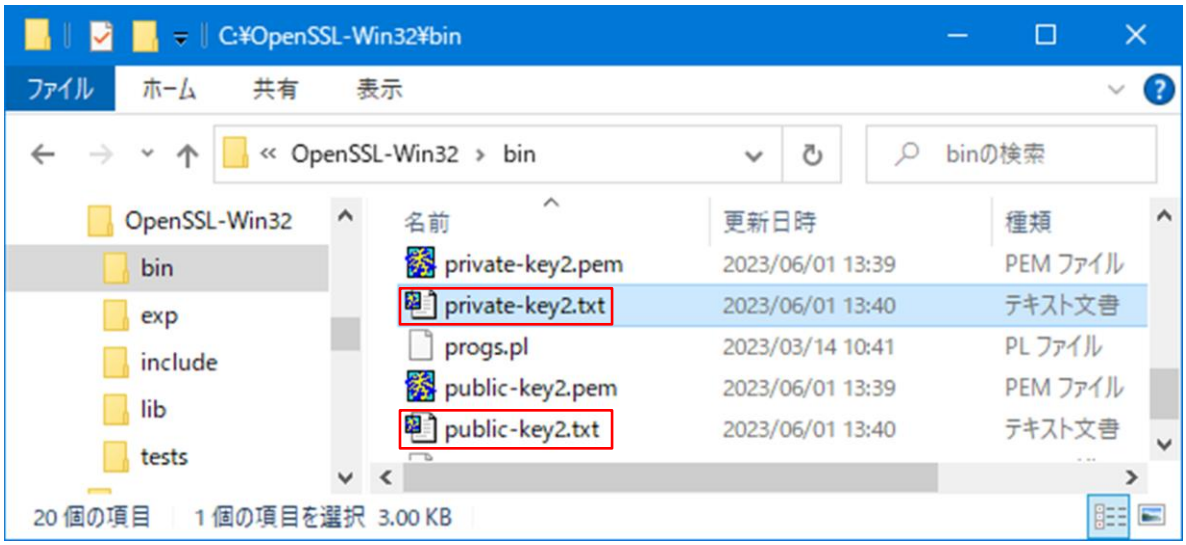
openssl rsa -in private-key2.pem -text -out private-key2.txt

２－７）RSA 公開鍵(public-key2.pem)を DER 形式のテキストファイルを生成



openssl rsa -pubin -in public-key2.pem -text -out public-key2.txt

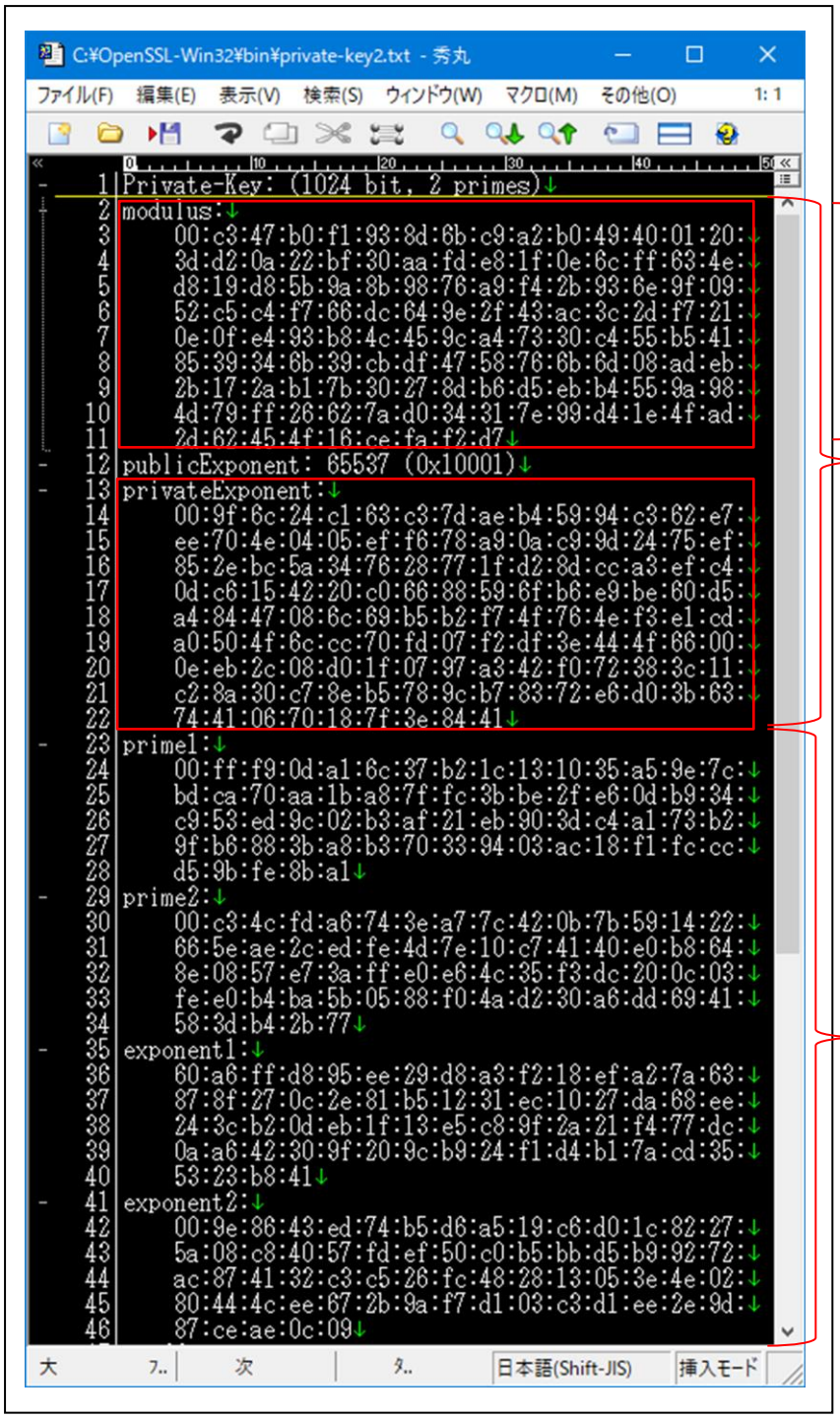
2－8）生成したDER形式のテキストファイルをメモ帳等のエディタで内容確認



<private-key2.txt>

<public-key2.txt>

２－９生成した DER 形式の秘密鍵(private-key2.txt)の説明



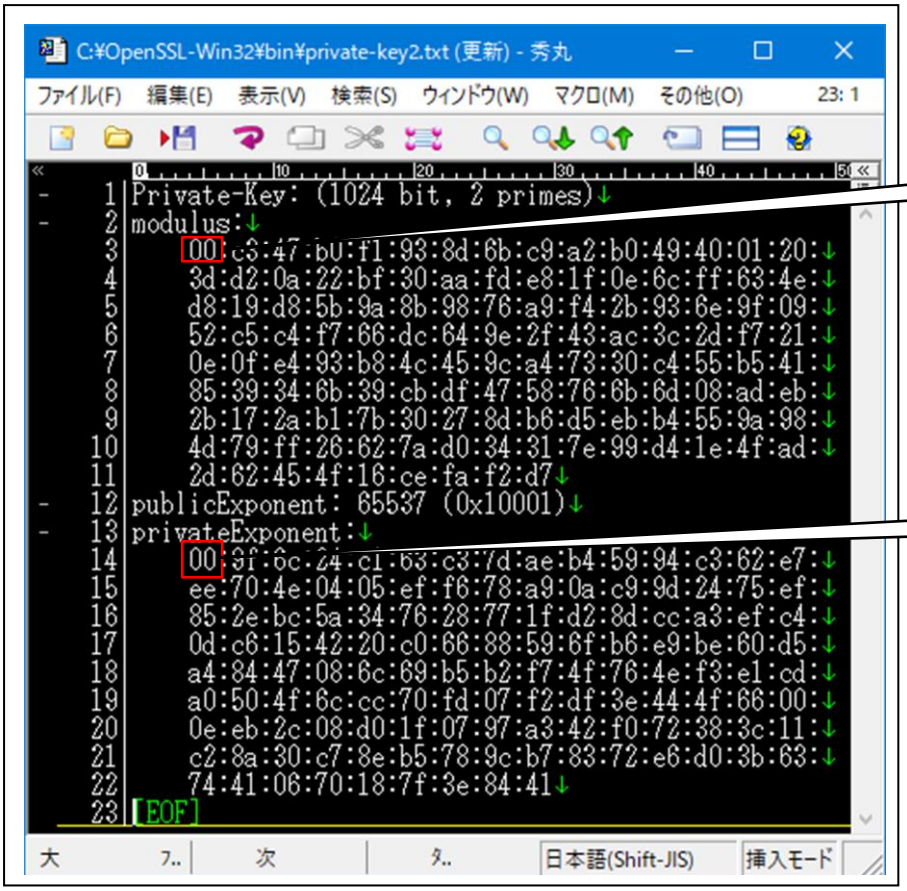
[modulus]
公開鍵として使用する。

[modulus]と
[privateExponent]
を組み合わせて秘密鍵に使用する。

[prime1]以下の数値は、秘密鍵を生成するために使用した素数等の数値です。無視しても構いません。

3）公開鍵よ秘密鍵を S5D9 側に登録するため DER 形式のデータを C 言語用に編集する。

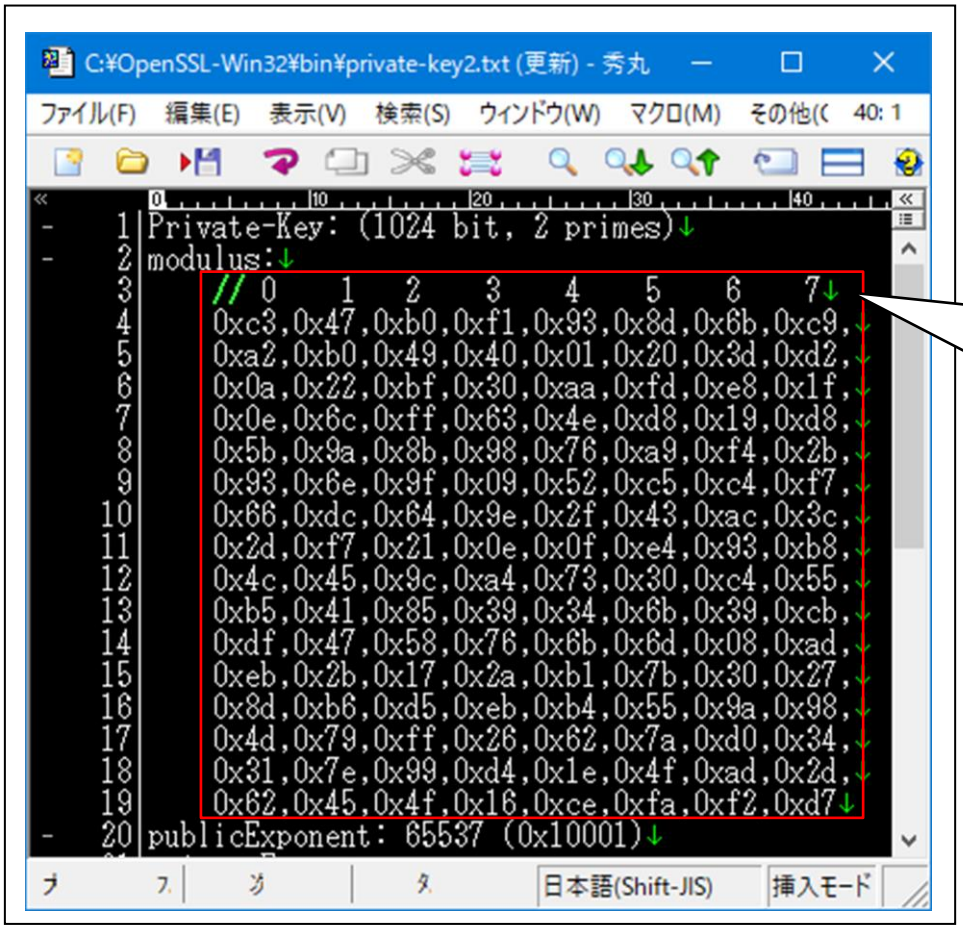
<説明>



先頭の[00]は無視する

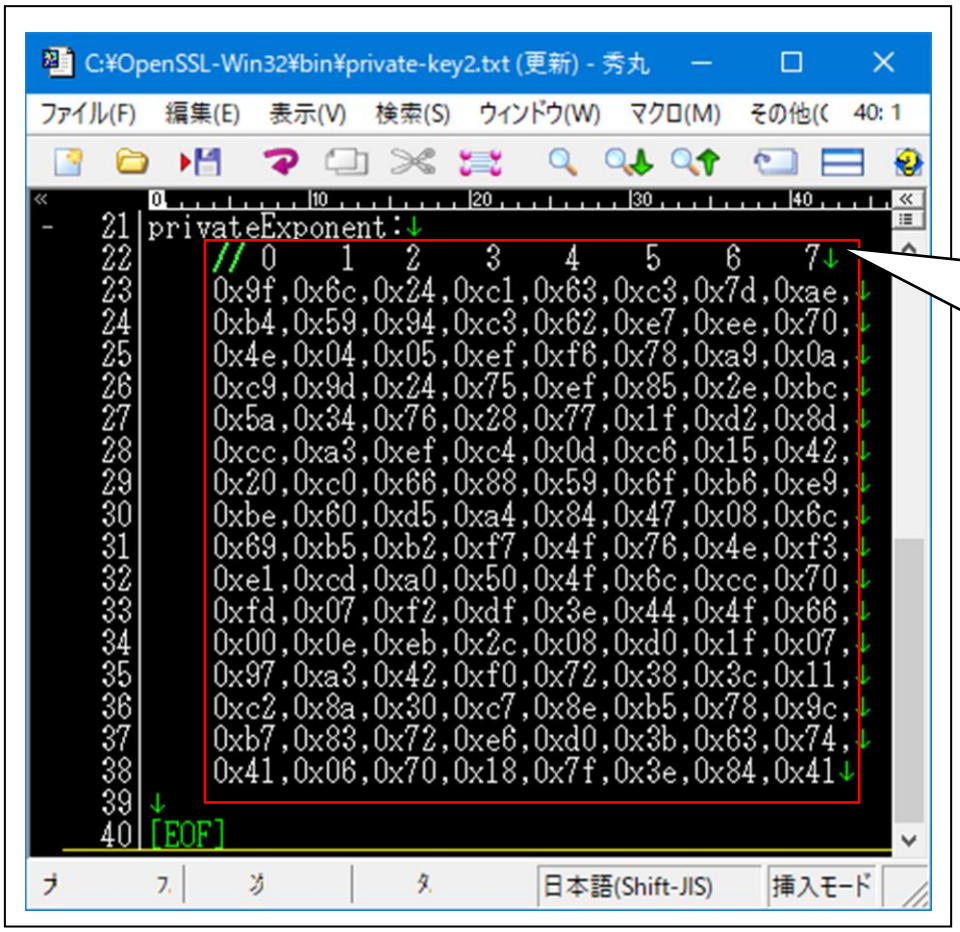
先頭の[00]は無視する

<modulus の編集>



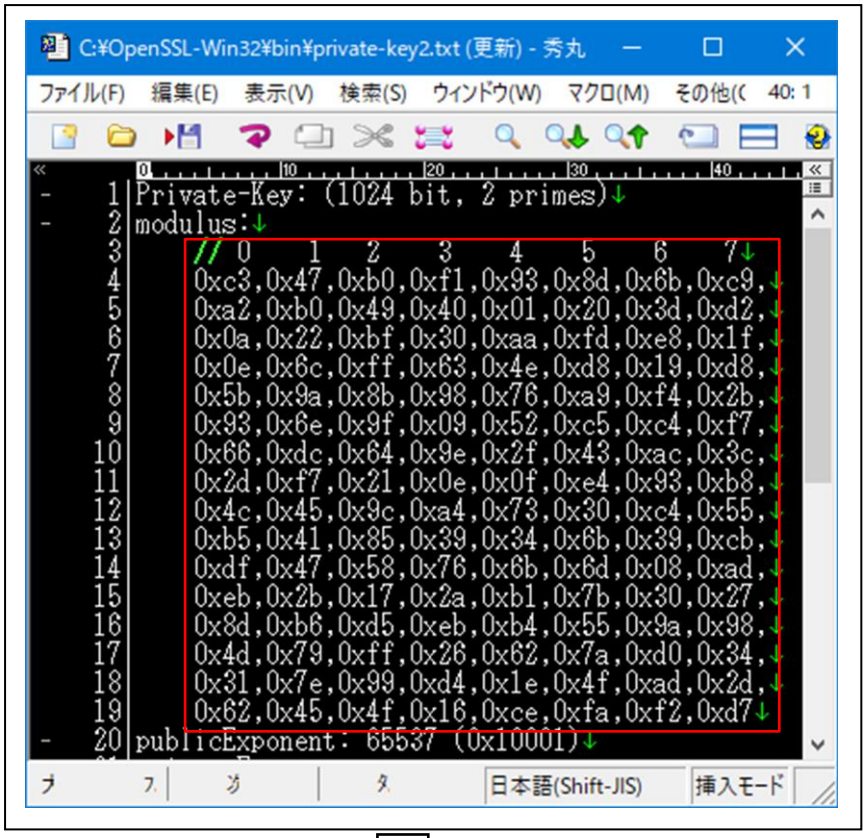
e2studio でビルドが通るように C ベースの Hex 記述に編集する。

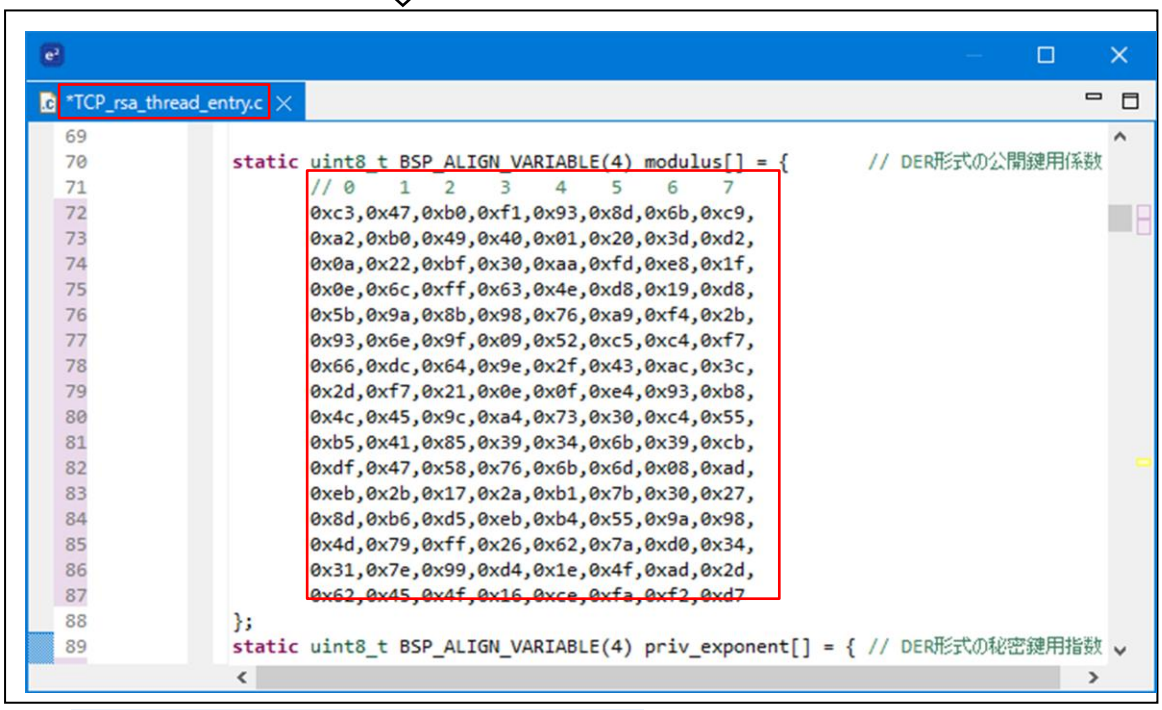
<privateExponent の編集>



e2studio でビルドが通るように C ベースの Hex 記述に編集する。

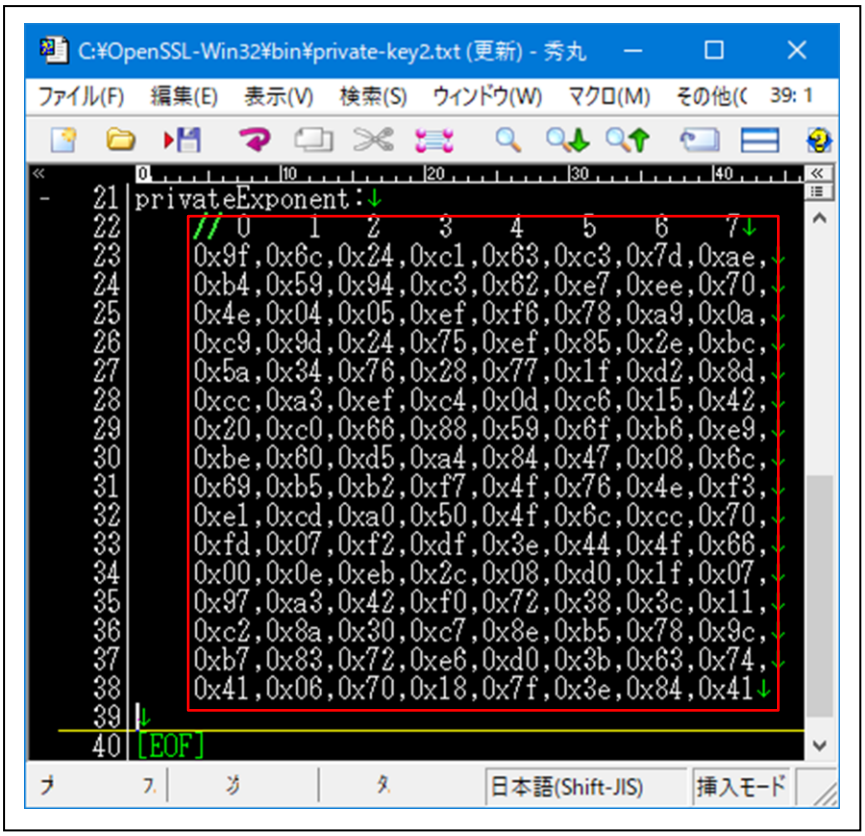
4）前項の3）で編集したDER形式の「modules」を置き換える

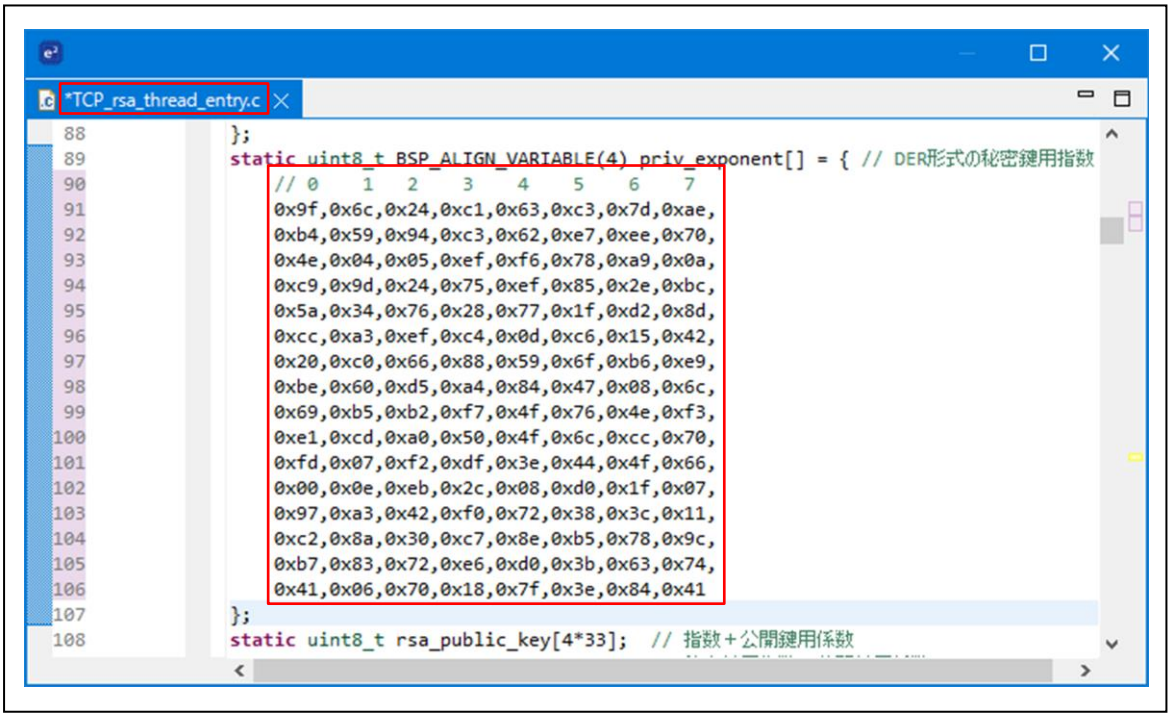




変数「`static uint8_t BSP_ALIGN_VARIABLE(4) modulus[]`」

5）前項の３）で編集した DER 形式の「privateExponent」を置き換える





変数「`static uint8_t BSP_ALIGN_VARIABLE(4) priv_exponent[]`」

6）クリーン・ビルドする。

９．注意事項

・本文書の著作権は、エーワン（株）が保有します。
・本文書を無断での転載は一切禁止します。
・本文書に記載されている内容についての質問やサポートはお受けすることが出来ません。
・本文章に関して、ルネサス　エレクトロニクス社への問い合わせは御遠慮願います。
・本文書の内容に従い、使用した結果、損害が発生しても、弊社では一切の責任は負わないものとします。
・本文書の内容に関して、万全を期して作成しましたが、ご不審な点、誤りなどの点がありましたら弊社までご連絡くだされば幸いです。
・本文書の内容は、予告なしに変更されることがあります。

１０．商標

・e2studio は、ルネサス　エレクトロニクス株式会社の登録商標、または商品名称です。
・Renesas Synergy™および S3A7/S5D9/S7G2 は、ルネサス　エレクトロニクス株式会社の登録商標、または商品名です。
・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。

１０．参考文献

・「S3A7　ユーザーズマニュアル　ハードウェア編」　ルネサス　エレクトロニクス株式会社
・「S5D9　ユーザーズマニュアル　ハードウェア編」　ルネサス　エレクトロニクス株式会社
・「S7G2　ユーザーズマニュアル　ハードウェア編」　ルネサス　エレクトロニクス株式会社
・ルネサス　エレクトロニクス株式会社提供のサンプル集
・「e2studio ユーザーズマニュアル　入門ガイド」　ルネサス　エレクトロニクス株式会社
・「SSP vx.x.x User's Manual」　ルネサス　エレクトロニクス株式会社
・「X-Ware Component Documents for Renesas Synergy™」ルネサス　エレクトロニクス株式会社
・その他

〒486-0852
愛知県春日井市下市場町 6-9-20
エーワン株式会社
https://www.robin-w.com