Rev 1.10.00

Renesas S5D9 用サンプル(e2studio WIRE_DHCP_TCP_AES_COMMAND)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。

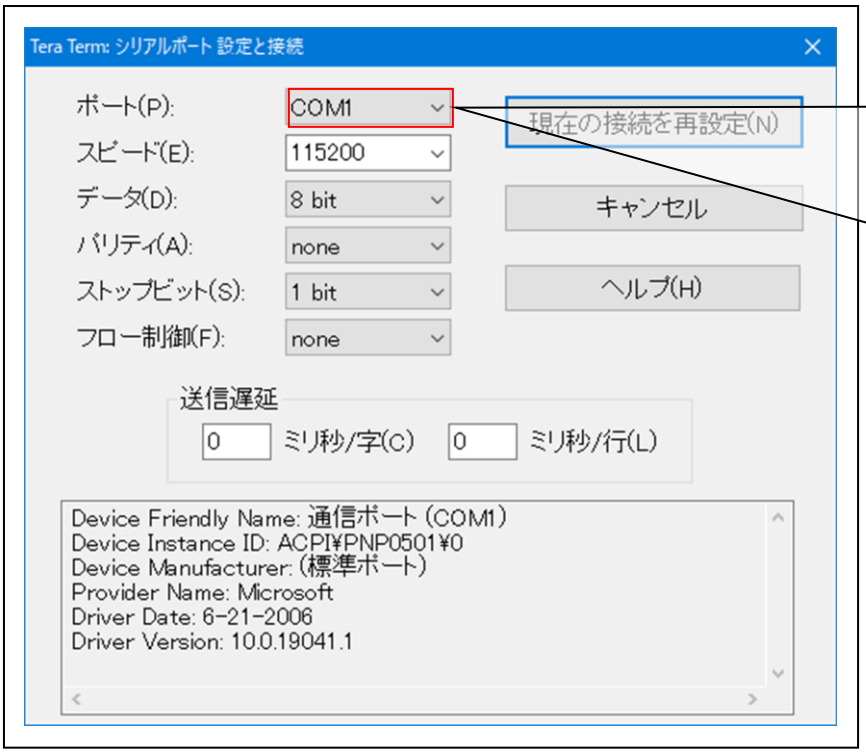
2．サンプルのプロジェクト名

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 AES を使用したサンプル（暗号・復号)	WIRE_DHCP_TCP_AES_COMMAND 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(AES) (g_sce_aes_r.p_api->encrypt) (g_sce_aes_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

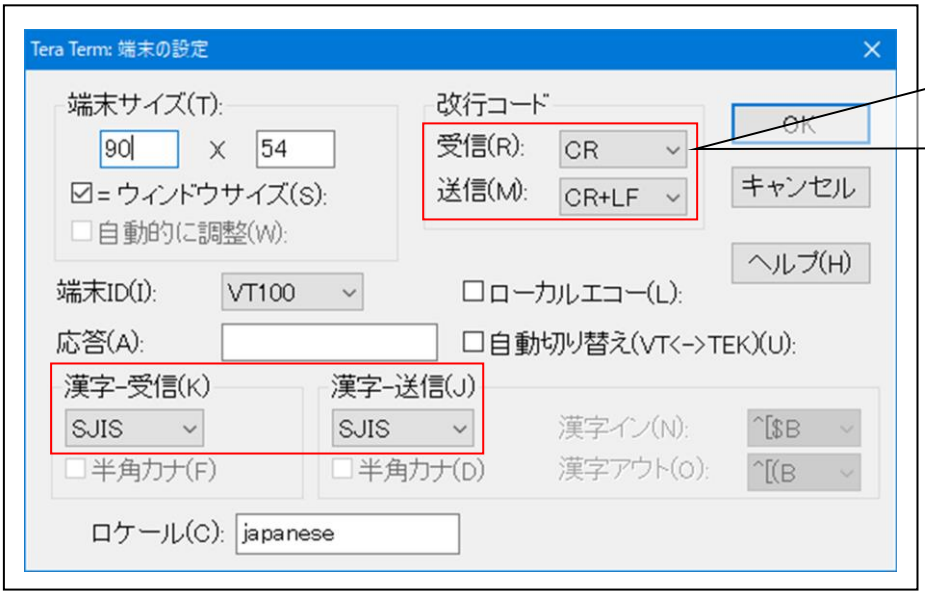


COM 番号は、
PC 側でシリアル通信可能な番号を指定する。

115200BPS
8bit
none
1bit
none

の仕様にする。

④端末の設定

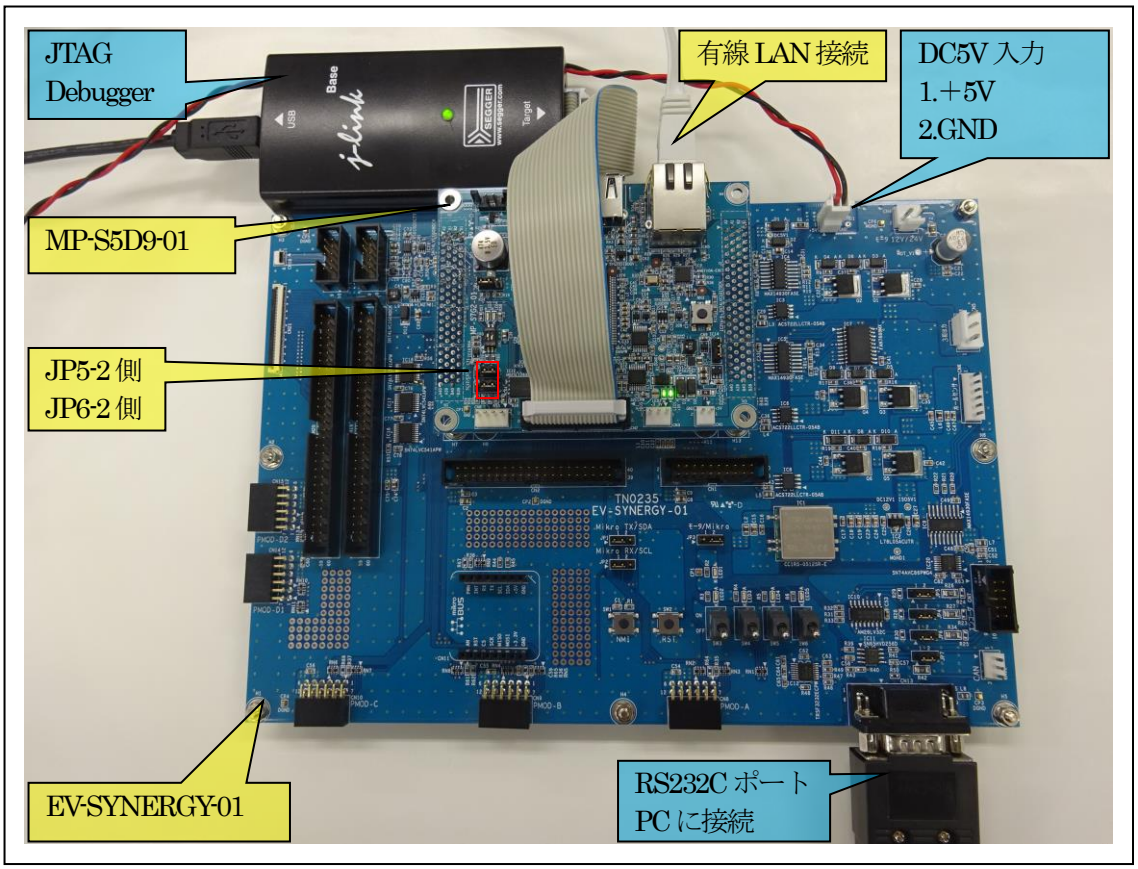


USB シリアルコンバータ使用時に CR コードがカットされる設定の場合は、受信：LF にして下さい。

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。

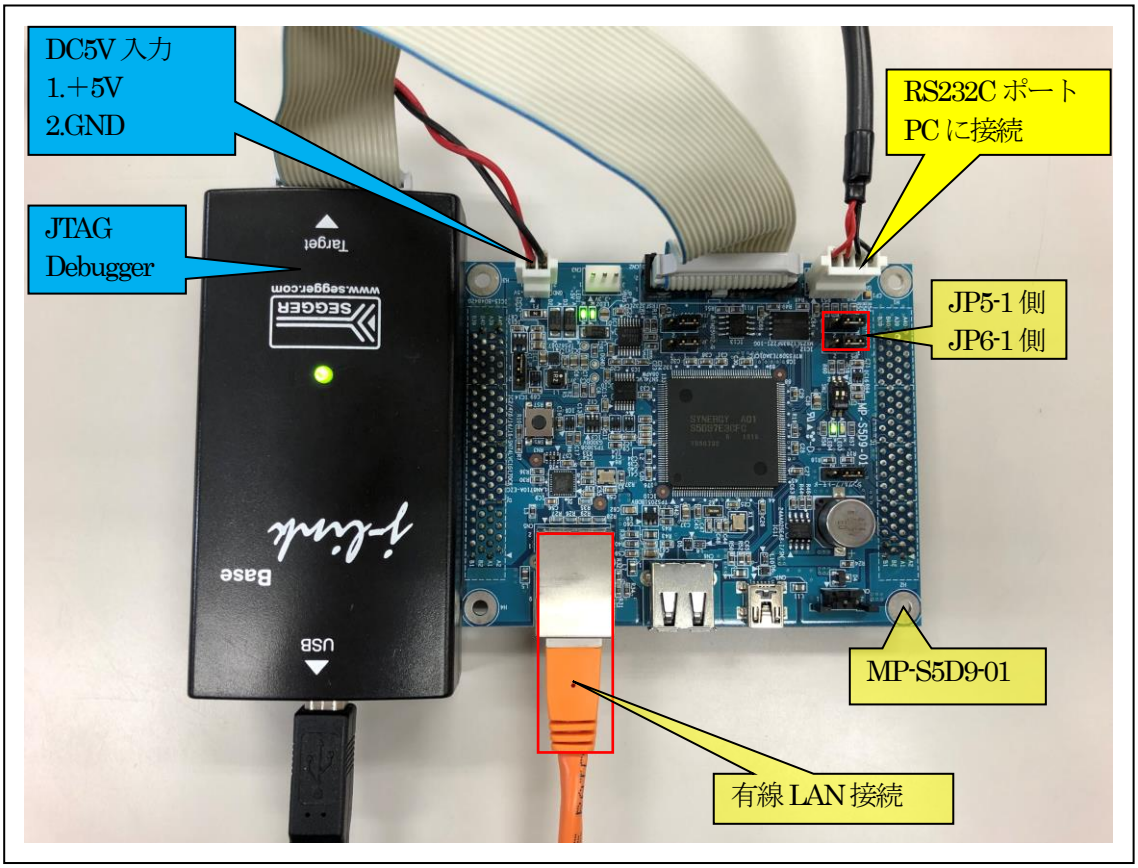
EV-SYNERGY-xx

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_AES_COMMAND」フォルダ構成とファイル名

S5D9_e2std_WIRE_NetX_1¥WIRE_DHCP_TCP_AES_COMMAND			
Debug		WIRE_DHCP_TCP_AES_COMMAND.elf	ELF ファイル、JTAG で使用
		WIRE_DHCP_TCP_AES_COMMAND.map	MAP ファイル、アドレス情報管理
		WIRE_DHCP_TCP_AES_COMMAND.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_aes	sce7_aes.c	AES の暗号・復号化サンプルファイル
		sce7_aes.h	sce7_aes.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_AES_COMMAND.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>"」

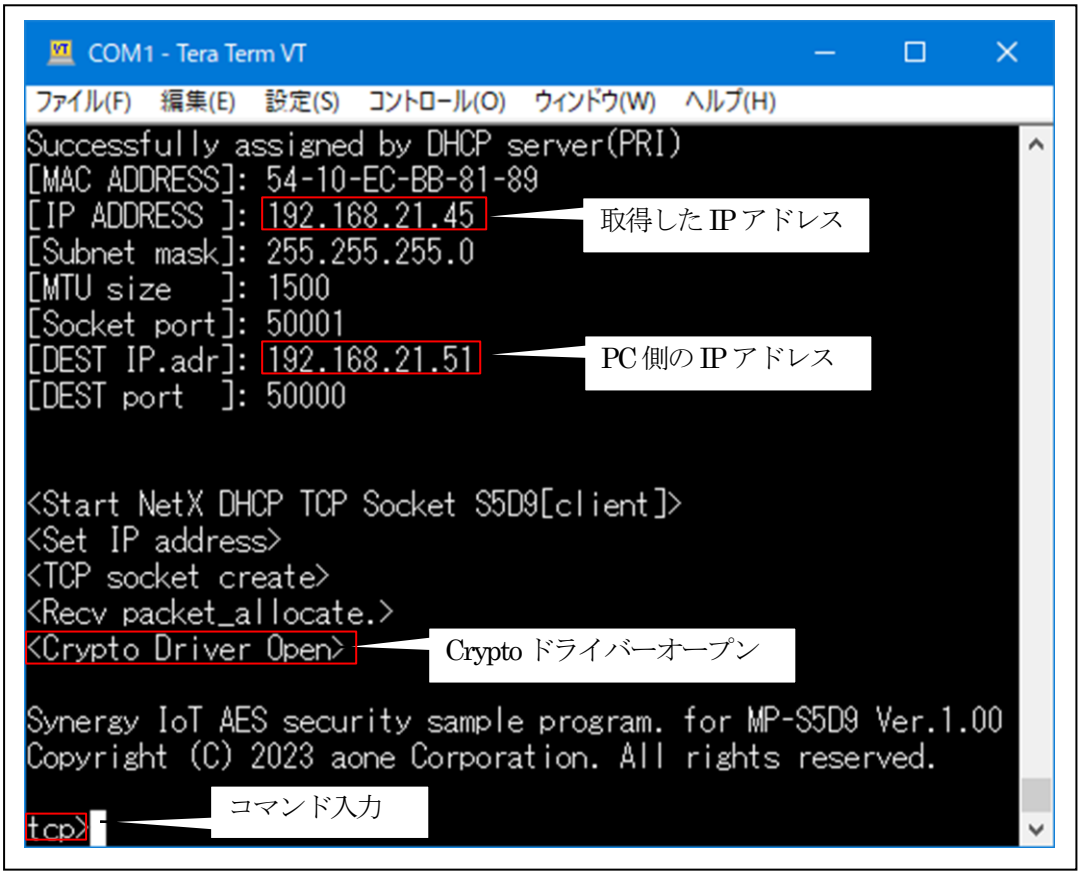
＜４＞「"<Wait DHCP State Change>"」

＜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 LED４を led blink thread で 200msec 毎に点滅

２）IP ドレス取得情報を Term 画面に表示する。



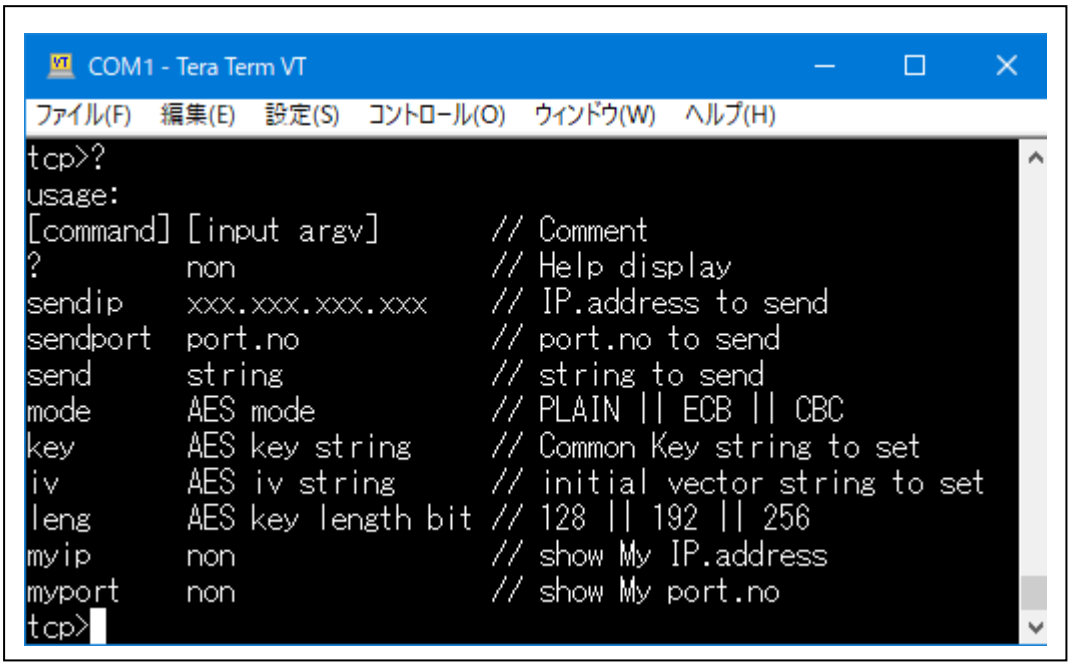
＜TCP Cmd Thread＞

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<ECB>文字列を「AES-ECB モード」で暗号化して送信する。
③<CBC>文字列を「AES-CBC モード」で暗号化して送信する。
④<CTR>文字列を「AES-CTR モード」で暗号化して送信する。
⑤<GCM>文字列を「AES-GCM モード」で暗号化して送信する。
⑥<XTS>文字列を「AES-XTS モード」で暗号化して送信する。

＜例＞

xxx> mode PLAIN⏎
xxx> mode ECB⏎
xxx> mode CBC⏎
xxx> mode CTR⏎
xxx> mode GCM⏎
xxx> mode XTS⏎
xxx> mode⏎
[Now mode]: ECB

（6）ｋｅｙ（共通鍵の設定と参照）

＜機能＞

・暗号・復号化するための共通鍵を設定する。（16 進数設定）

＜AES-XTS 以外＞

// key 長 128bit
xxx> key 0x11111111 0x22222222 0x33333333 0x44444444⏎
// key 長 196bit
xxx> key 0x11111111 0x22222222 0x33333333 0x44444444 0x55555555 0x66666666 ⏎
// key 長 256bit
xxx> key 0x11111111 0x22222222 0x33333333 0x44444444 0x55555555 0x66666666 0x77777777 0x88888888⏎
xxx> key⏎
[Now Key]: 11111111 22222222 33333333 44444444

＜AES-XTS は Key 長が倍になる＞

// key 長 128bit
xxx> key 0x11111111 0x22222222 0x33333333 0x44444444 0x55555555 0x66666666 0x77777777 0x88888888⏎
// key 長 256bit
xxx> key 0x11111111 0x22222222 0x33333333 0x44444444 0x55555555 0x66666666 0x77777777 0x88888888
0x99999999 0xaaaaaaaa 0xbbbbbbbb 0xcccccccc 0xdddddddd 0xeeeeeeee 0xffffffff 0x11111111⏎
xxx> key⏎
[Now Key]: 0x11111111 0x22222222 0x33333333 0x44444444
:0x55555555 0x66666666 0x77777777 0x88888888
:0x99999999 0xAAAAAAAA 0xBBBBBBBB 0xCCCCCCCC
:0xDDDDDDDD 0xEEEEEEEE 0xFFFFFFFF 0x11111111

（7）ｉｖ（初期ベクタ値の設定と参照）
＜機能＞
・暗号・復号化（ＡＥＳ－ＣＢＣ）モード時の初期ベクタ値を設定する。（16 進数設定）

＜AES-GCM 以外＞
xxx>iv 0x02013B01 0x1DF2D6C7 0x846F189E 0xDB0A8254↵
xxx>iv↵[Now IV]: 0x02013B01 0x1DF2D6C7 0x846F189E 0xDB0A8254

＜AES-GCM は、128bit に Format された 96bitIV にする必要あり＞
xxx>iv 0x02013B01 0x1DF2D6C7 0x846F189E 0x01000000↵
xxx>iv↵[Now IV]: 0x02013B01 0x1DF2D6C7 0x846F189E 0x01000000

96bitIV: 0x02013B01 0x1DF2D6C7 0x846F189E
128bit に Format された 96bitIV:バイト列で表記
0x01,0x3B,0x01,0x02,0xC7,0xD6,0xF2,0x1D,0x9E,0x18,0x6F,0x84,0x00,0x00,0x00,0x01
AES-GCM の IV の 96bit 以降は、必ず、バイト表記で 0x00,0x00,0x00,0x01 にする必要があります。

（8）ｌｅｎｇ（共通鍵長の設定と参照）
＜機能＞
・共通鍵の長さ(Key length 128/192/256)を設定する。（10 進数設定）

＜例＞
xxx>leng 128↵
xxx>leng 192↵
xxx>leng 256↵
xxx>leng↵[
[Now Key Length]: 128

（9）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。

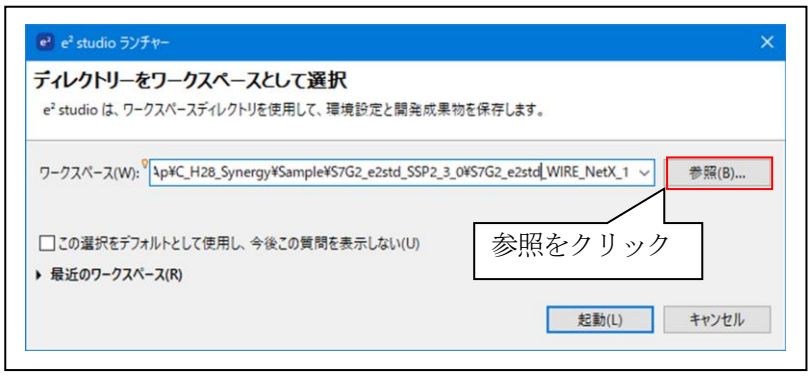
＜例＞
xxx>myip↵
[My IP]: 192.168.21.9

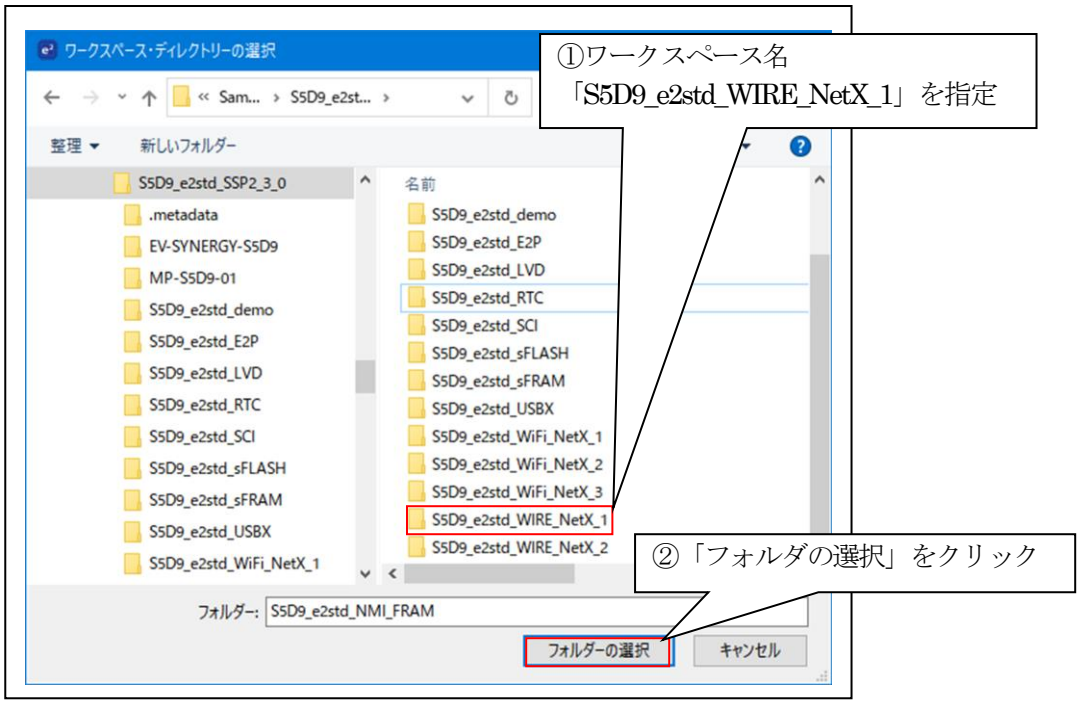
（１０）ｍｙｐｏｒｔ（基板側で登録したポート番号の参照）
＜機能＞
・基板側で登録したポート番号を表示する。

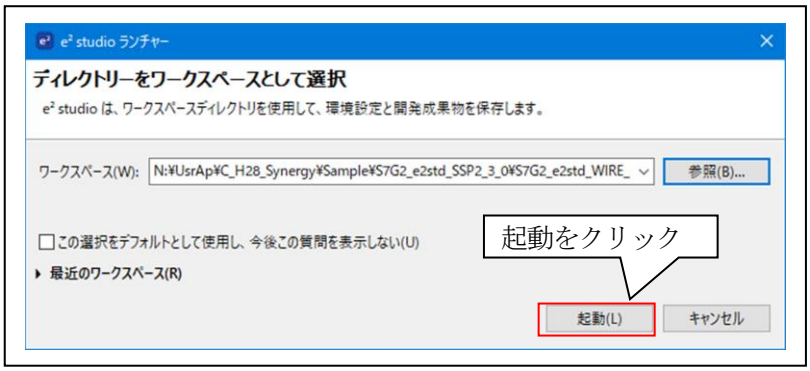
＜例＞
xxx>myport↵
[My port]: 50001

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

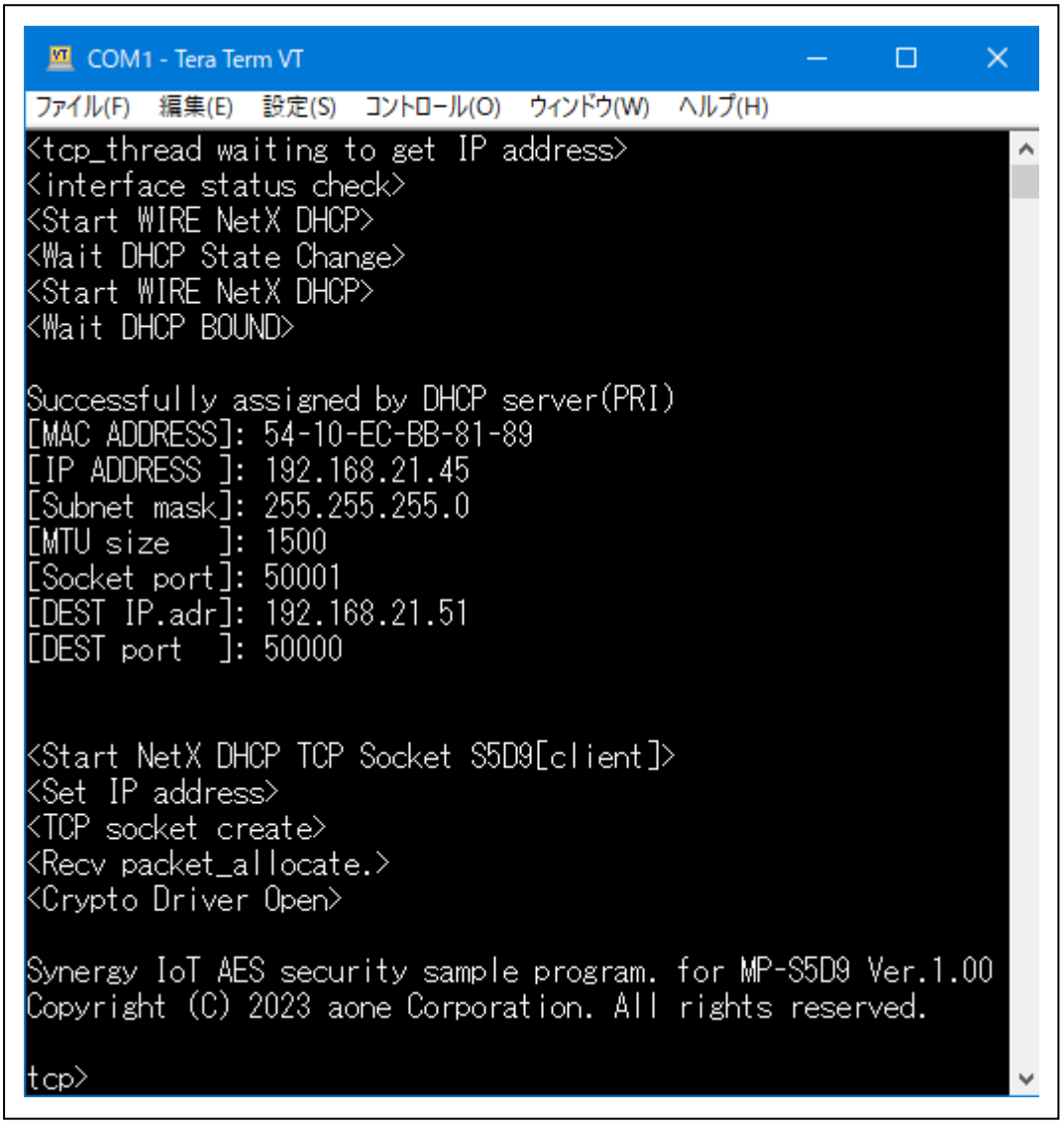
☆詳細操作は「e2studio_synergy_Import.pdf」の２項を参照して下さい。

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。

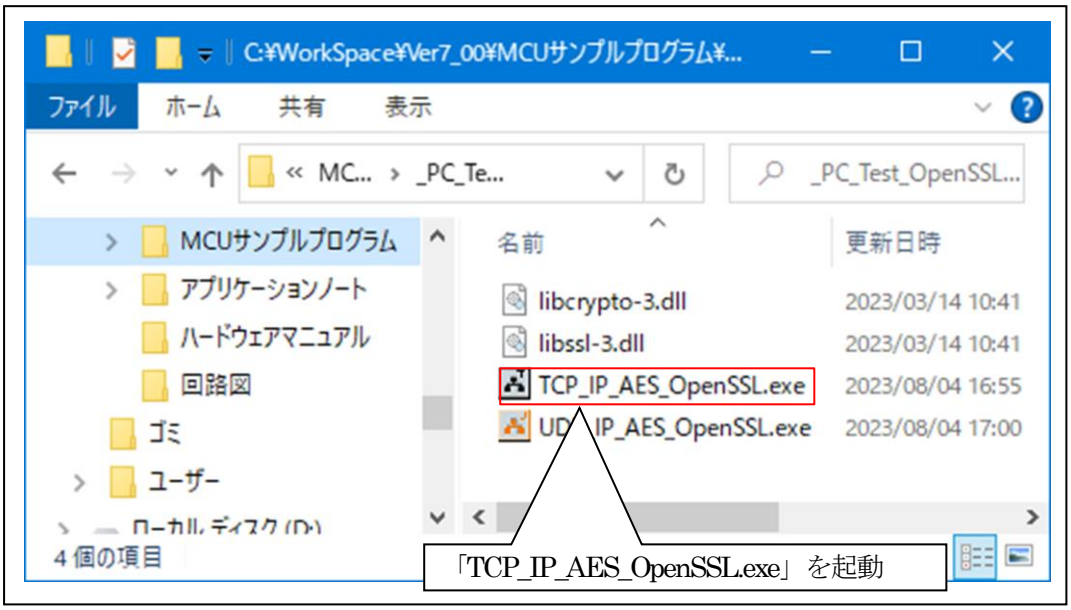
＜WIRE_DHCP_TCP 実行画面＞



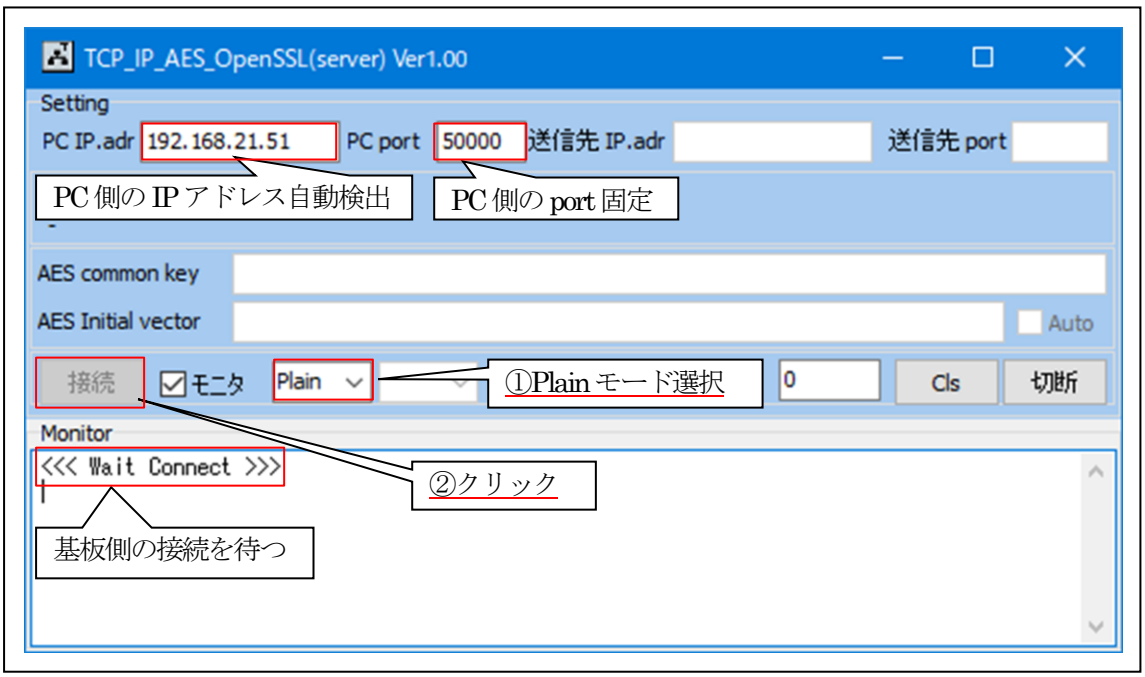
7－3．Windows PC 側のテスト用プログラムで動作確認（PLAIN（平文）モード）

1）「TCP_IP_AES_OpenSSL」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_OpenSSL】を Read/Write 可能な適当なフォルダに Copy してからお使い下さい。



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



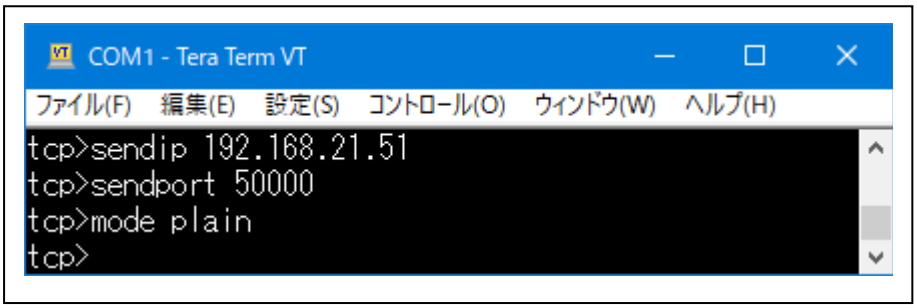
3）「基板」側の各項目の確認と設定。

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⏎

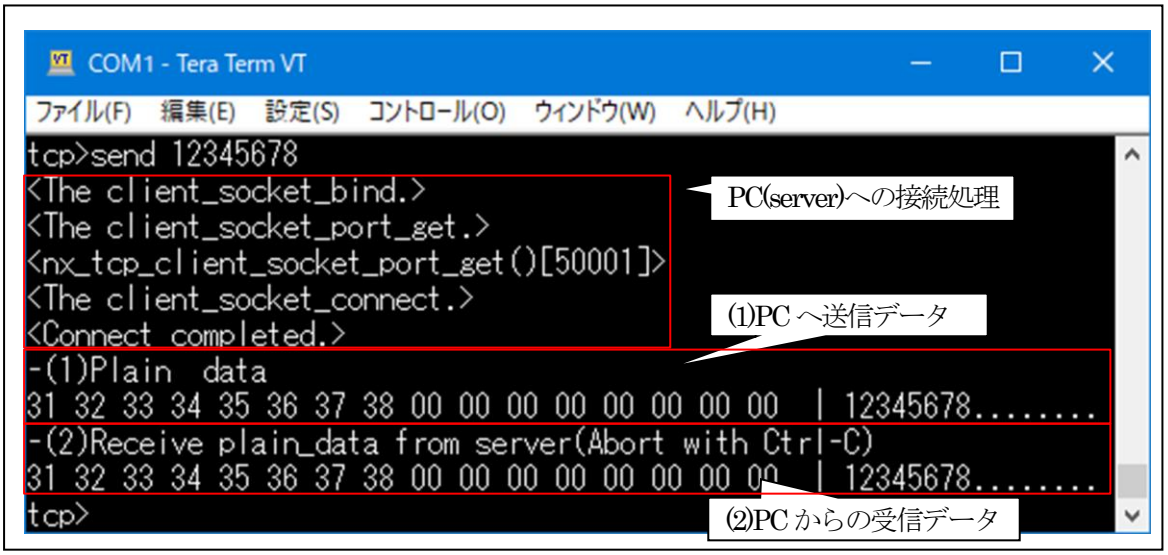
③「plain」（平文）モードを指定する。
ex) mode plain⏎

④実行画面

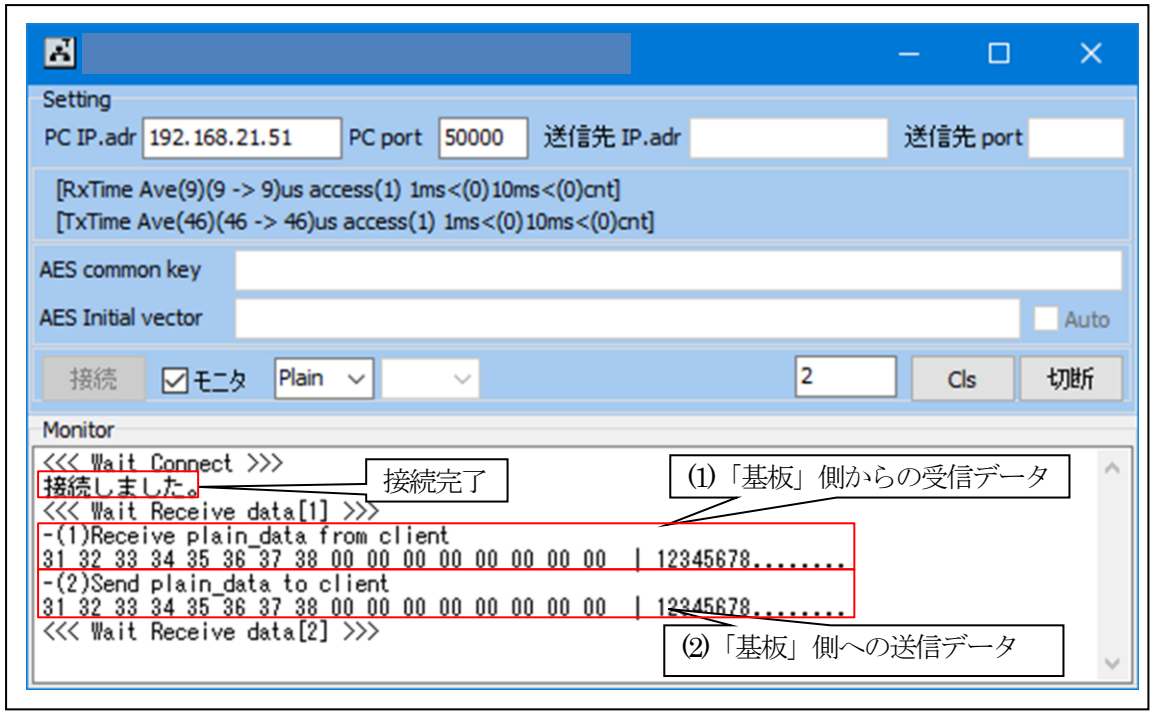


4）「基板」側から「PC」(server) 側へ平文テキストを送信する。
①テキスト文を送信する。（未接続の場合は、接続を実行）

ex) send 12345678

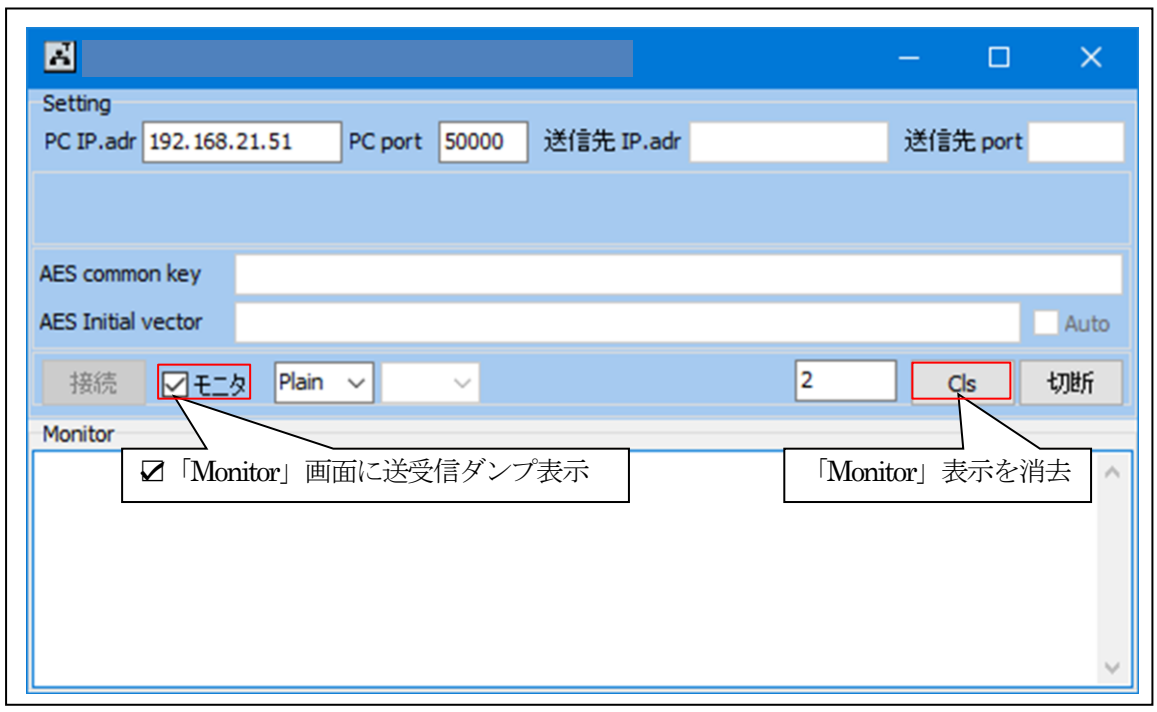


5）「TCP_IP_AES_OpenSSL」側の送受信を確認する。

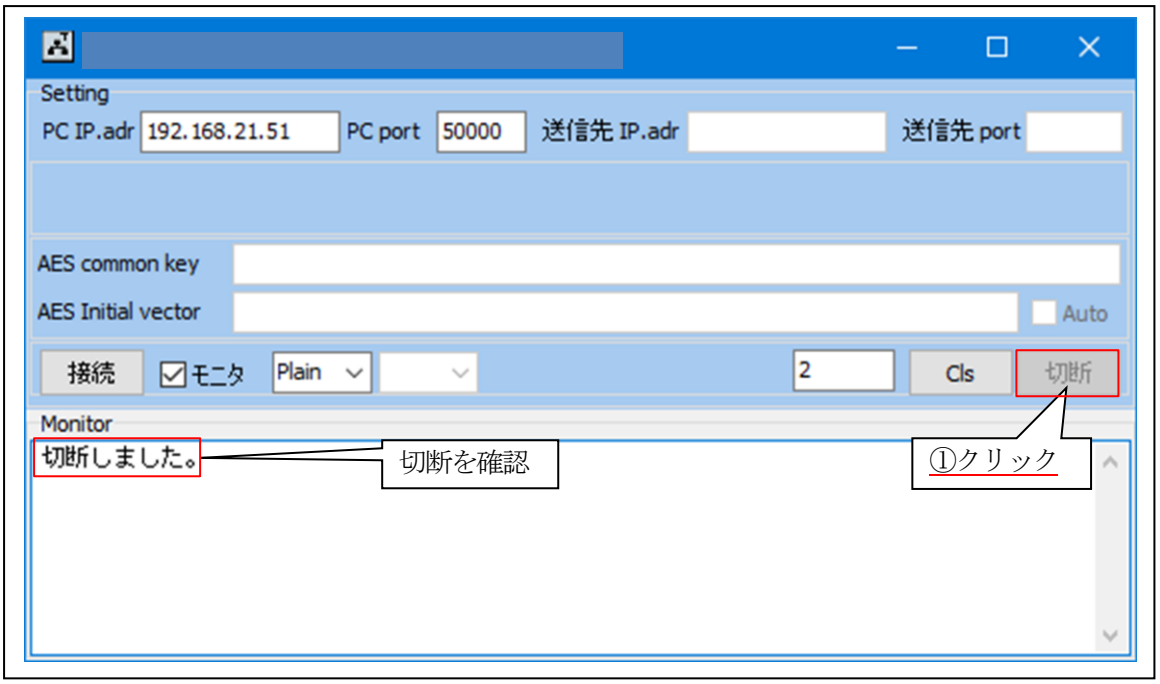


【Error 表示】
・受信時の接続失敗「”error!! client_socket_connect()[error code]”」
・受信異常「” error!! recvfrom()[error code]”」
・送信異常「”error!! sendto()[error code]”」

6）「TCP_IP_AES_OpenSSL」その他の操作（各モード共通）



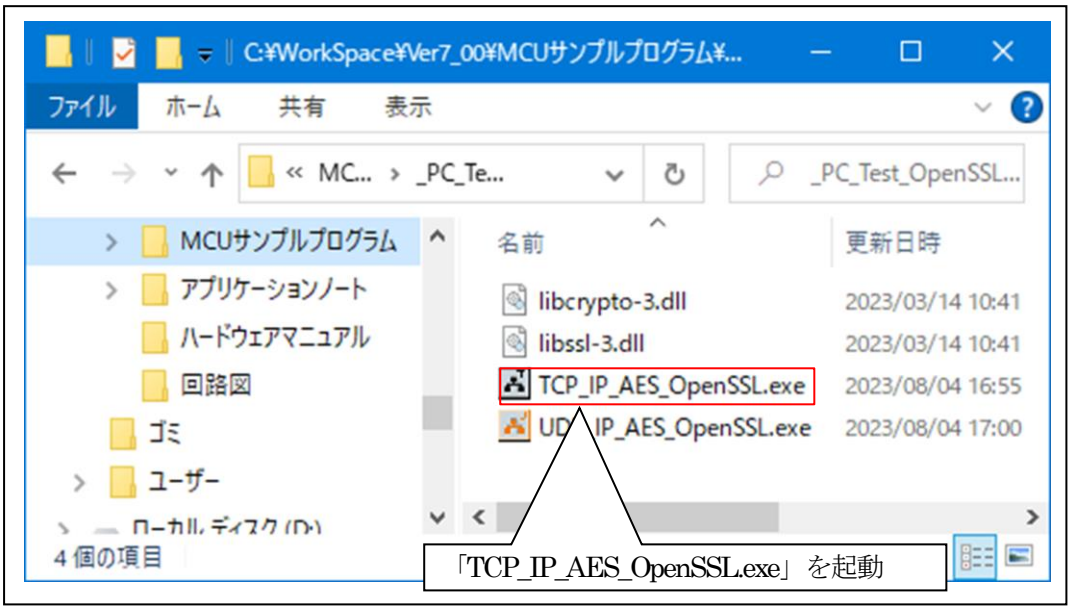
7）TCP_IP-Port を「切断」する。（各モード共通）



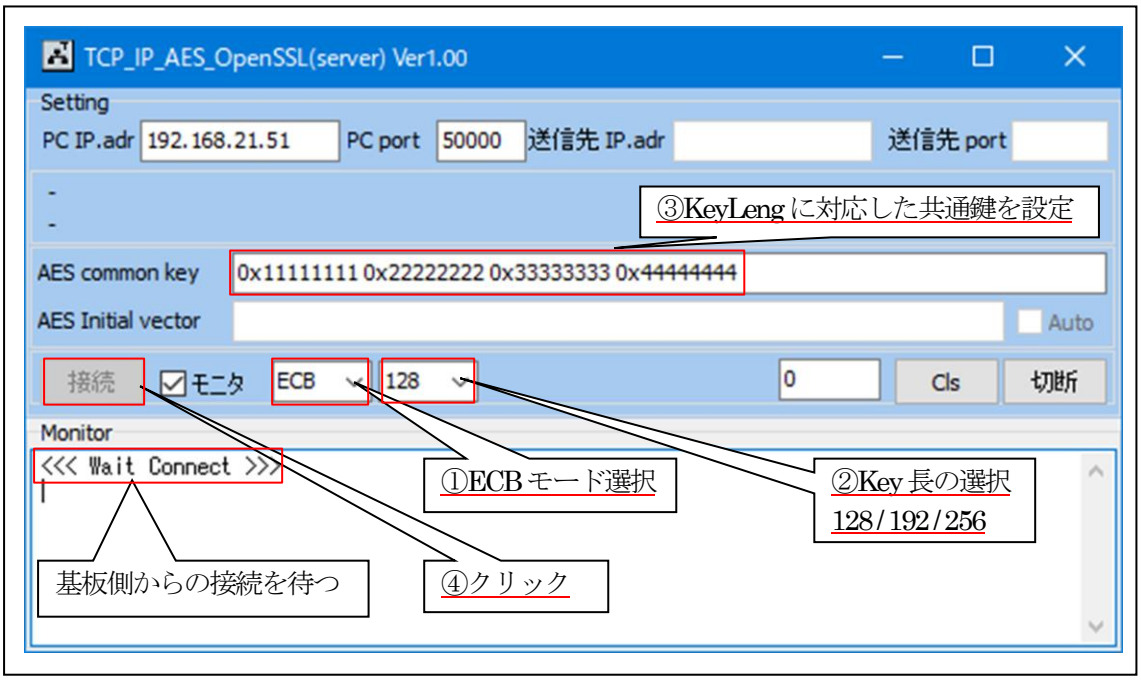
７－４．Windows PC 側のテスト用プログラムで動作確認（AES-ECB モード）

１）「TCP_IP_AES_OpenSSL」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_OpenSSL】を Read/Write 可能な適当なフォルダに Copy してからお使い下さい。



２）「TCP_IP_AES_OpenSSL」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⏎

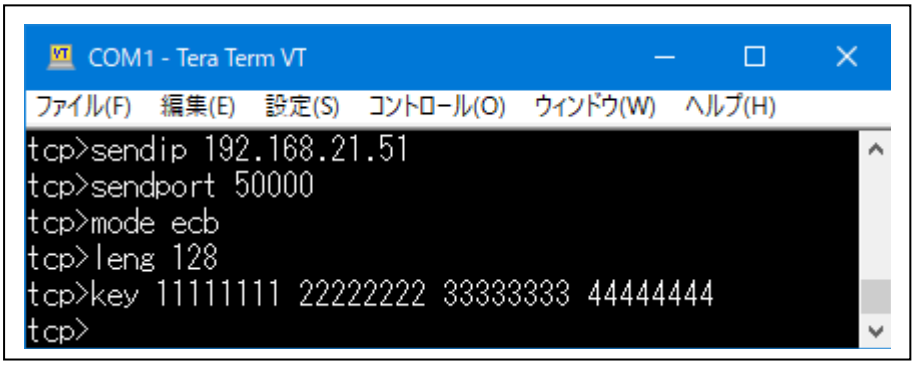
②PC 側のポート番号を設定する。
ex) sendport 50000⏎

③「ECB」（AES-ECB）モードを指定する。
ex) mode ecb⏎

④「Key Leng」を指定する。(128 / 192 / 256)
PC 側の「TCP_IP_AES_OpenSSL」と同じ Key 長にする。
ex) leng 128⏎

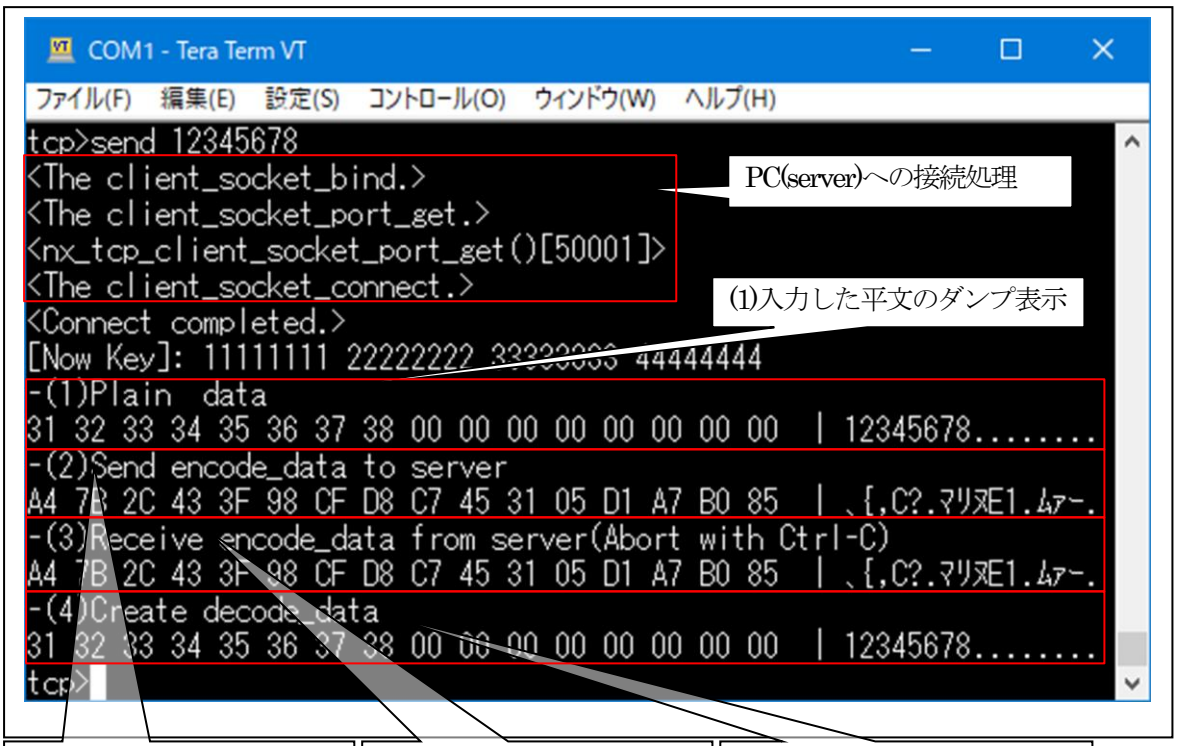
⑤「共通鍵」を指定する。（HEX　数値）
PC 側の「TCP_IP_AES_OpenSSL」と同等の共通鍵(AES common key)にする。
ex) key 11111111 22222222 33333333 44444444⏎

⑥実行画面



4）「基板」側から「PC」(server)側へ ECB 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）

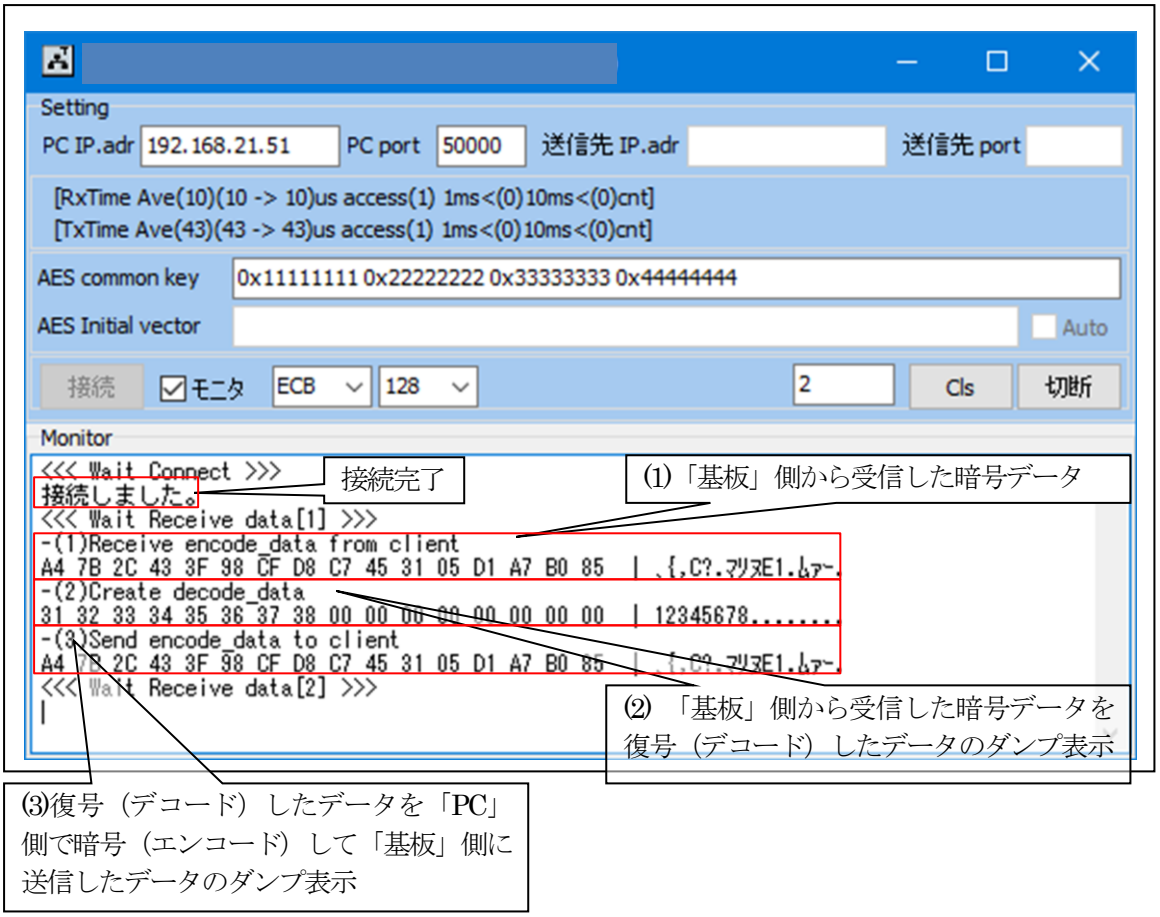
ex) send 12345678



②「(1)Plain data」と「(4)Create decode_data」が同等のダンプ表示の場合、基板側と PC 側が同等な復号処理（デコード）であることの証明になります。

③「(2)Send encode_data to server」と「(3)Receive encode_data from server(Abort with Ctrl-C)」が同等のダンプ表示の場合、基板側と PC 側が同等な暗号処理（エンコード）であることの証明になります。

5）「TCP_IP_AES_OpenSSL」側の送受信を確認する。



①「(1)Receive encode_data from client」と「(3)Send encode_data to client」が同等のダンプ表示の場合、基板側と PC 側が同等な暗号処理（エンコード）であることの実証になります。

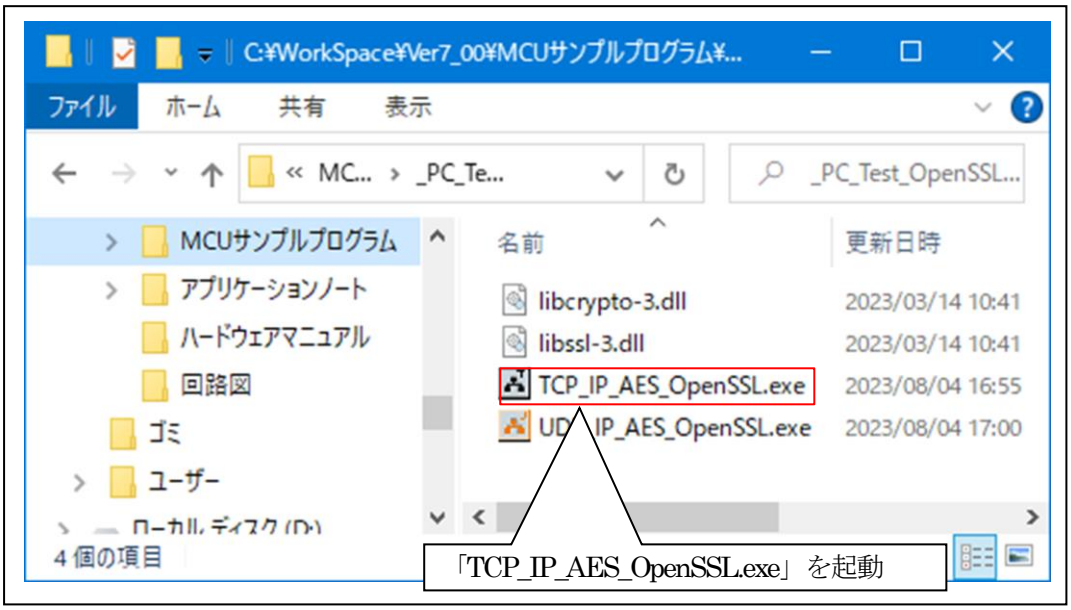
②「(2)Create decode_data」と「基板」側 TeraTerm 画面の「(4)Create decode_data」が同等のダンプ表示の場合、基板側と PC 側が同等な復号処理（デコード）であることの実証になります。

【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]”」

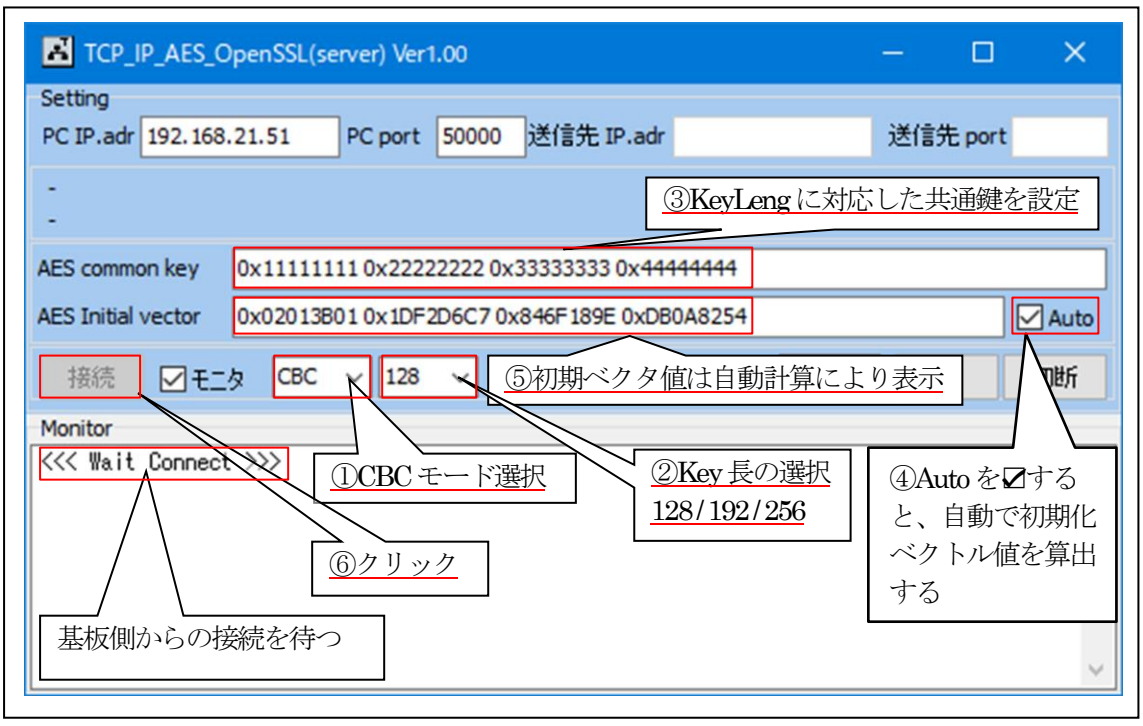
7－5．Windows PC 側のテスト用プログラムで動作確認（AES-CBC モード）

1）「TCP_IP_AES_OpenSSL」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_OpenSSL】を Read/Write 可能な適当なフォルダに Copy してからお使い下さい。



2）「TCP_IP_AES_OpenSSL」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⏎

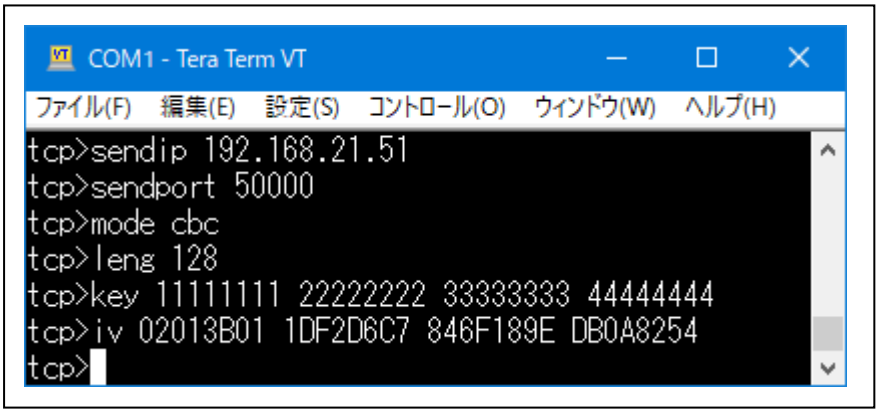
③「CBC」（AES-CBC）モードを指定する。
ex) mode cbc⏎

④「Key Leng」を指定する。(128 / 192 / 256)
PC 側の「TCP_IP_AES_OpenSSL」と同じ Key 長にする。
ex) leng 128⏎

⑤「共通鍵」を指定する。 （HEX　数値）
PC 側の「TCP_IP_AES_OpenSSL」と同等の共通鍵(AES common key)にする。
ex) key 11111111 22222222 33333333 44444444⏎

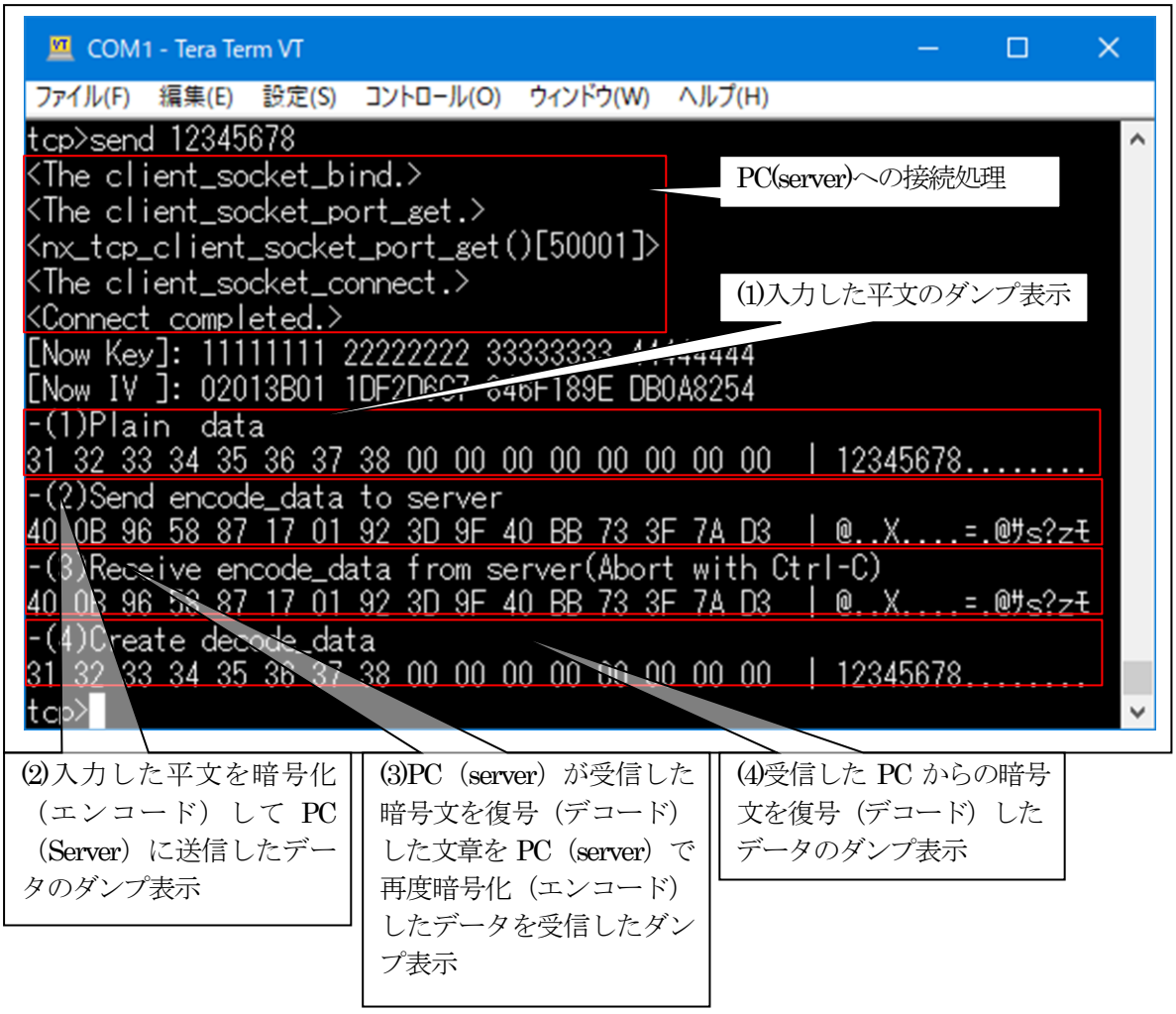
⑥「初期ベクタ値」を指定する。（HEX　数値）
PC 側の「TCP_IP_AES_OpenSSL」と同等の初期ベクタ値(AES Initial vector)にする。
ex) iv 02013B01 1DF2D6C7 846F189E DB0A8254⏎

⑦実行画面



4）「基板」側から「PC」(server)側へ CBC 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）

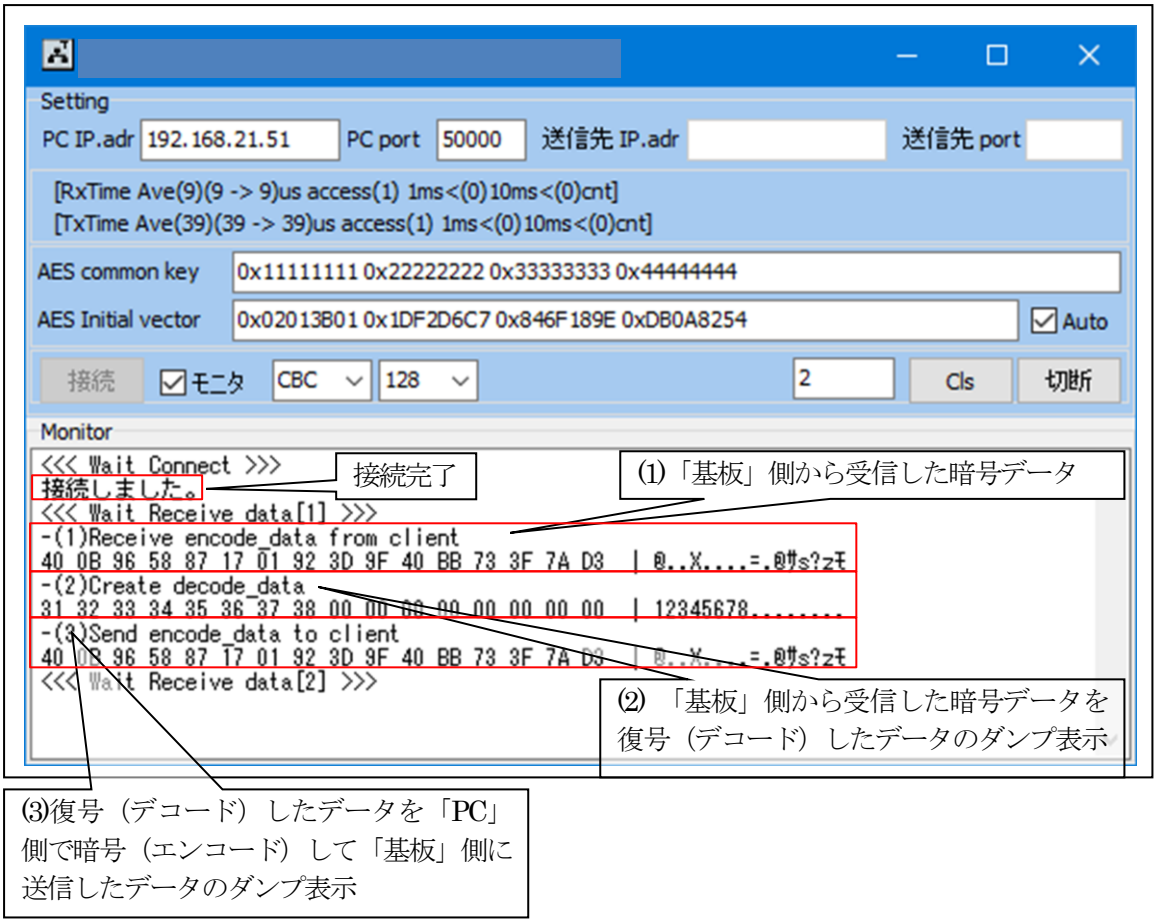
ex) send 12345678



②「(1)Plain data」と「(4)Create decode_data」が同等のダンプ表示の場合、基板側と PC 側が同等な復号処理（デコード）であることの実証になります。

③「(2)Send encode_data to server」と「(3)Receive encode_data from server(Abort with Ctrl-C)」が同等のダンプ表示の場合、基板側と PC 側が同等な暗号処理（エンコード）であることの実証になります。

5）「TCP_IP_AES_OpenSSL」側の送受信を確認する。



①「(1)Receive encode_data from client」と「(3)Send encode_data to client」が同等のダンプ表示の場合、基板側と PC 側が同等な暗号処理（エンコード）であることの実証になります。

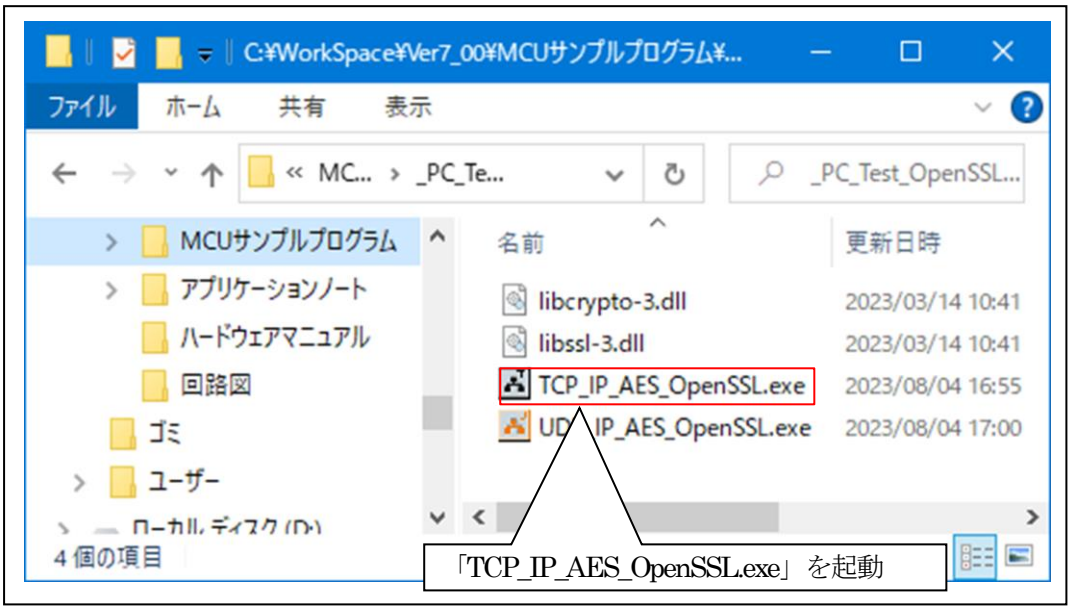
②「(2)Create decode_data」と「基板」側 TeraTerm 画面の「(4)Create decode_data」が同等のダンプ表示の場合、基板側と PC 側が同等な復号処理（デコード）であることの実証になります。

【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]”」

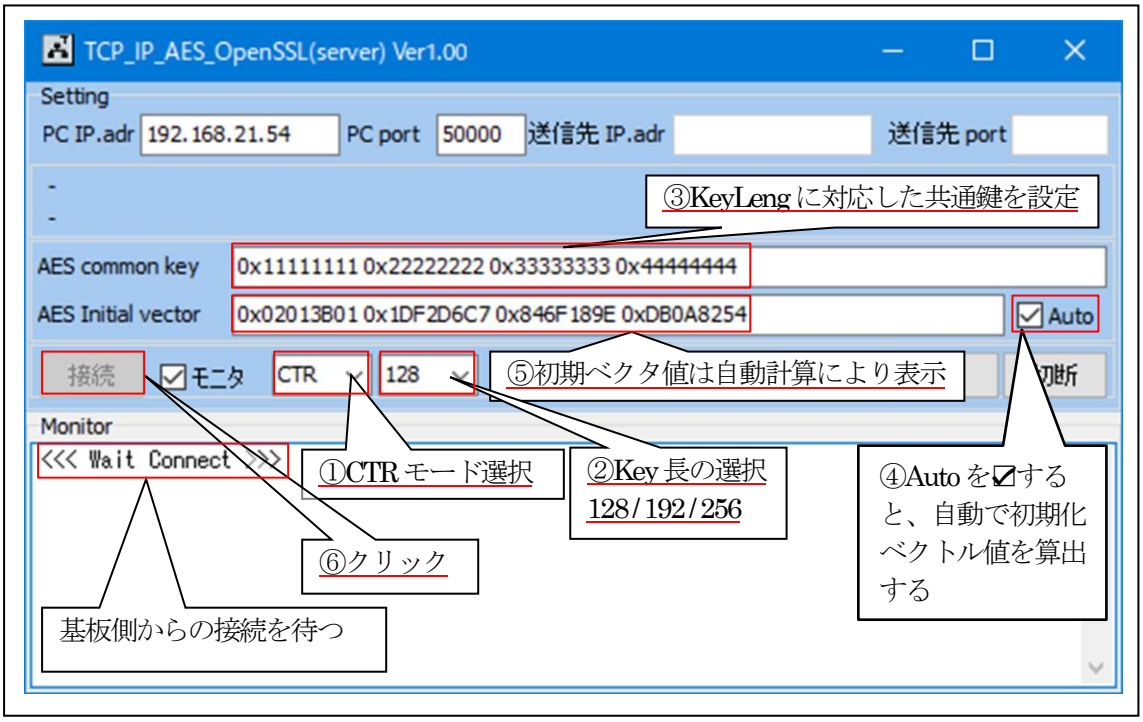
7－6．Windows PC 側のテスト用プログラムで動作確認（AES-CTR モード）

1）「TCP_IP_AES_OpenSSL」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_OpenSSL】を Read/Write 可能な適当なフォルダに Copy してからお使い下さい。



2）「TCP_IP_AES_OpenSSL」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 192.168.21.54↵

②PC 側のポート番号を設定する。
ex) sendport 50000↵

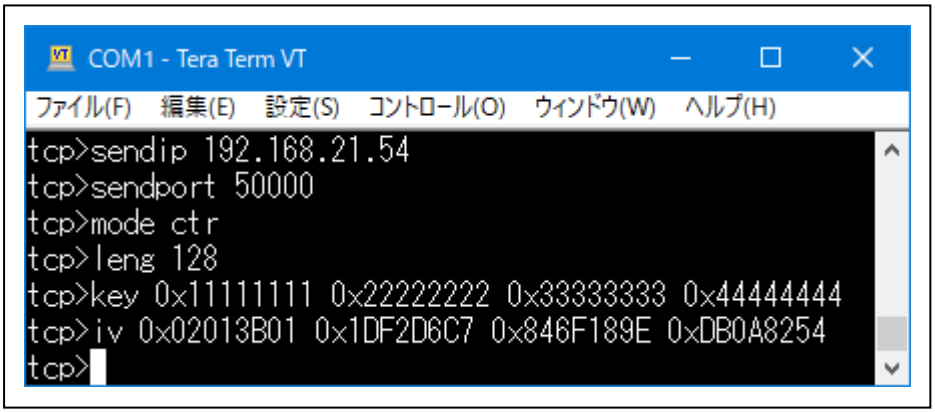
③「CTR」（AES-CTR）モードを指定する。
ex) mode ctr↵

④「Key Leng」を指定する。(128 / 192 / 256)
PC 側の「TCP_IP_AES_OpenSSL」と同じ Key 長にする。
ex) leng 128↵

⑤「共通鍵」を指定する。 （HEX　数値）
PC 側の「TCP_IP_AES_OpenSSL」と同等の共通鍵(AES common key)にする。
ex) key 11111111 22222222 33333333 44444444↵

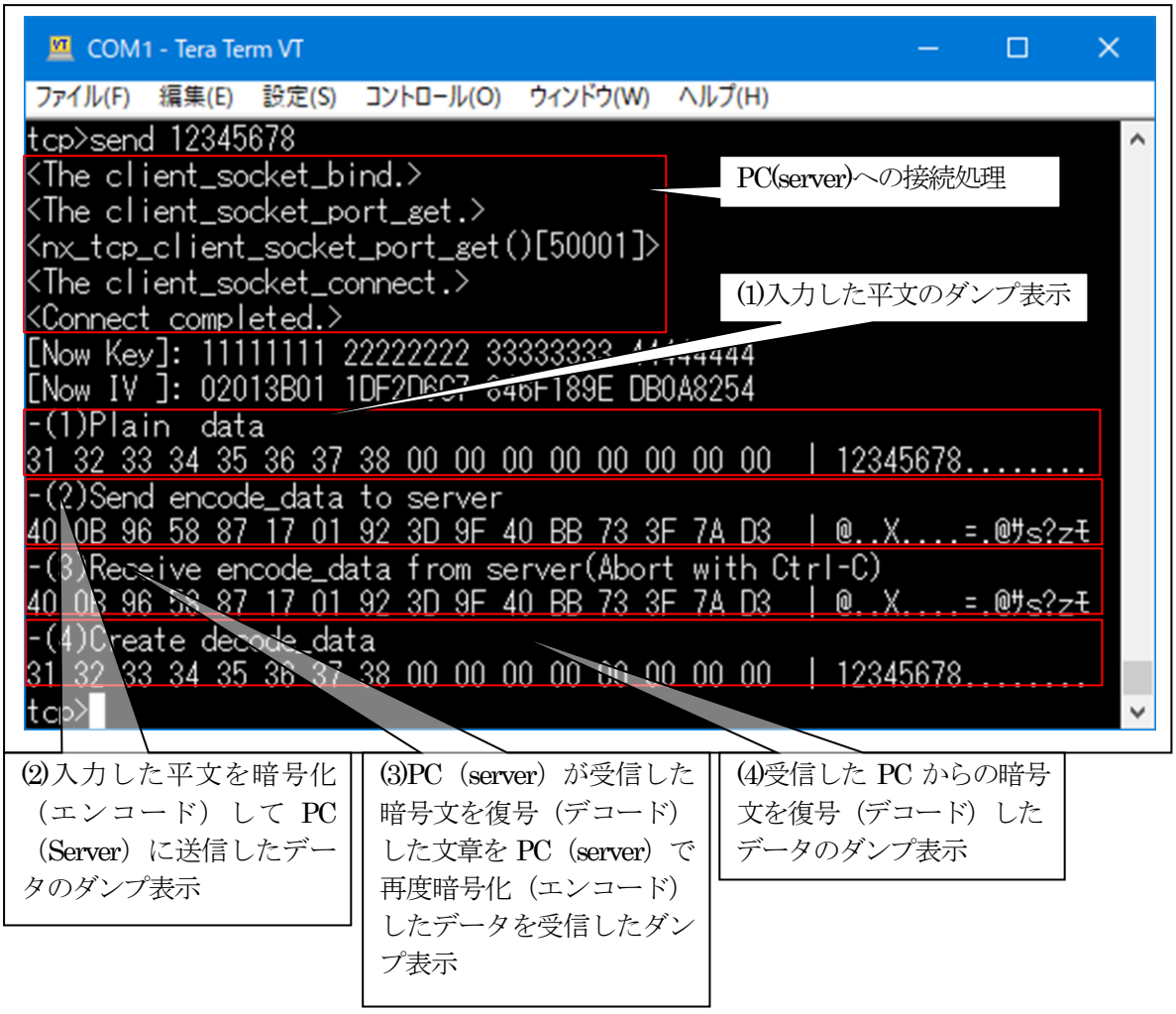
⑥「初期ベクタ値」を指定する。（HEX　数値）
PC 側の「TCP_IP_AES_OpenSSL」と同等の初期ベクタ値(AES Initial vector)にする。
ex) iv 02013B01 1DF2D6C7 846F189E DB0A8254↵

⑦実行画面



4）「基板」側から「PC」(server)側へCTR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）

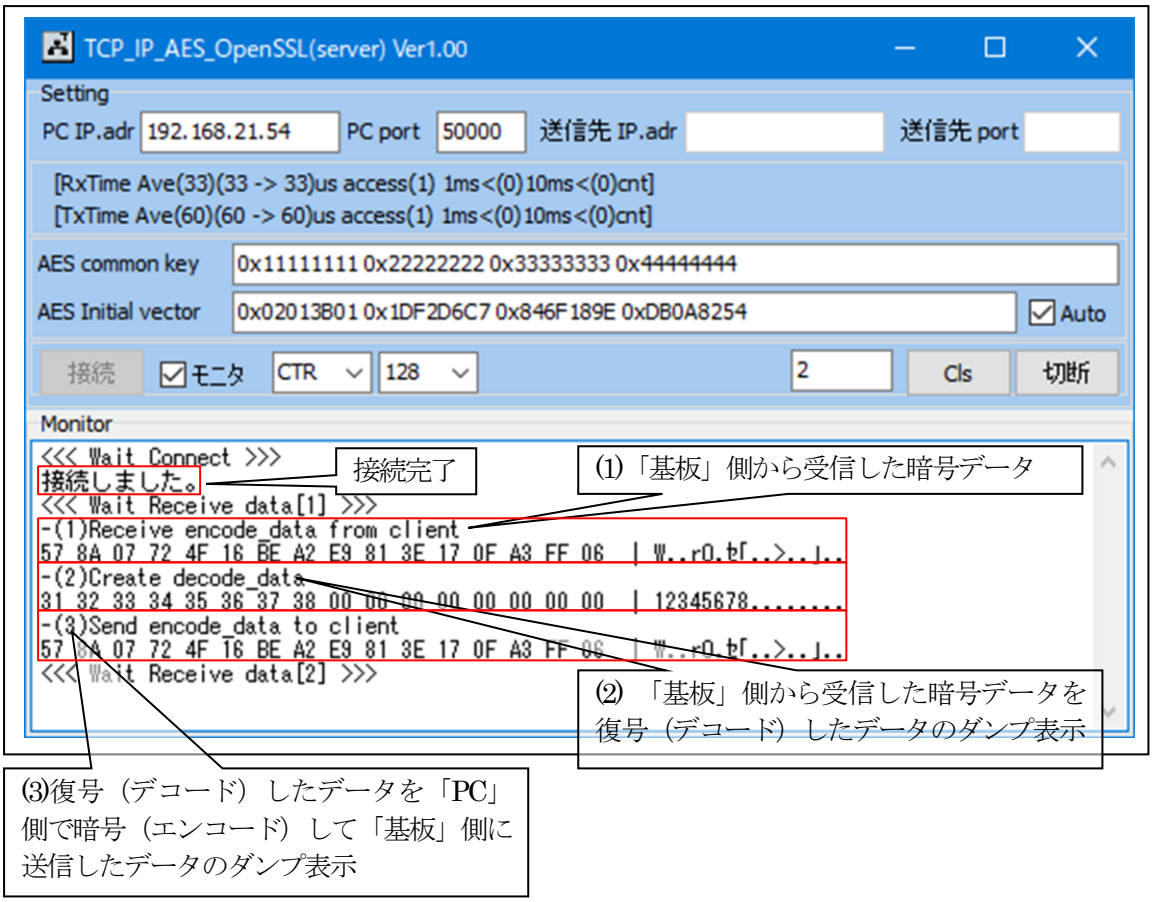
ex) send 12345678



②「(1)Plain data」と「(4)Create decode_data」が同等のダンプ表示の場合、基板側と PC 側が同等な復号処理（デコード）であることの実証になります。

③「(2)Send encode_data to server」と「(3)Receive encode_data from server(Abort with Ctrl-C)」が同等のダンプ表示の場合、基板側と PC 側が同等な暗号処理（エンコード）であることの実証になります。

5）「TCP_IP_AES_OpenSSL」側の送受信を確認する。



①「(1)Receive encode_data from client」と「(3)Send encode_data to client」が同等のダンプ表示の場合、基板側と PC 側が同等な暗号処理（エンコード）であることの実証になります。

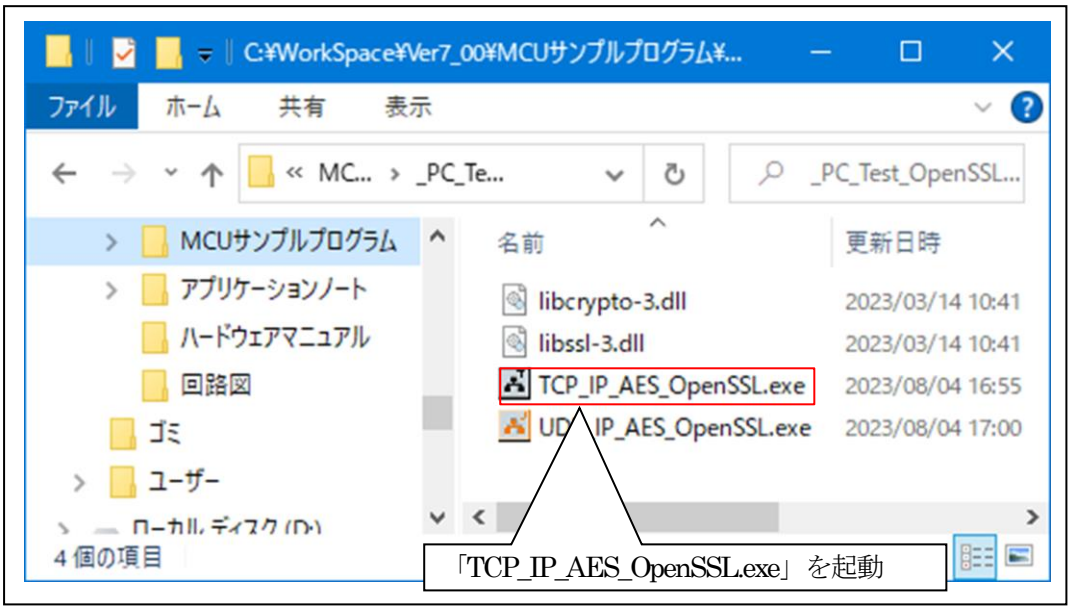
②「(2)Create decode_data」と「基板」側 TeraTerm 画面の「(4)Create decode_data」が同等のダンプ表示の場合、基板側と PC 側が同等な復号処理（デコード）であることの実証になります。

【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]”」

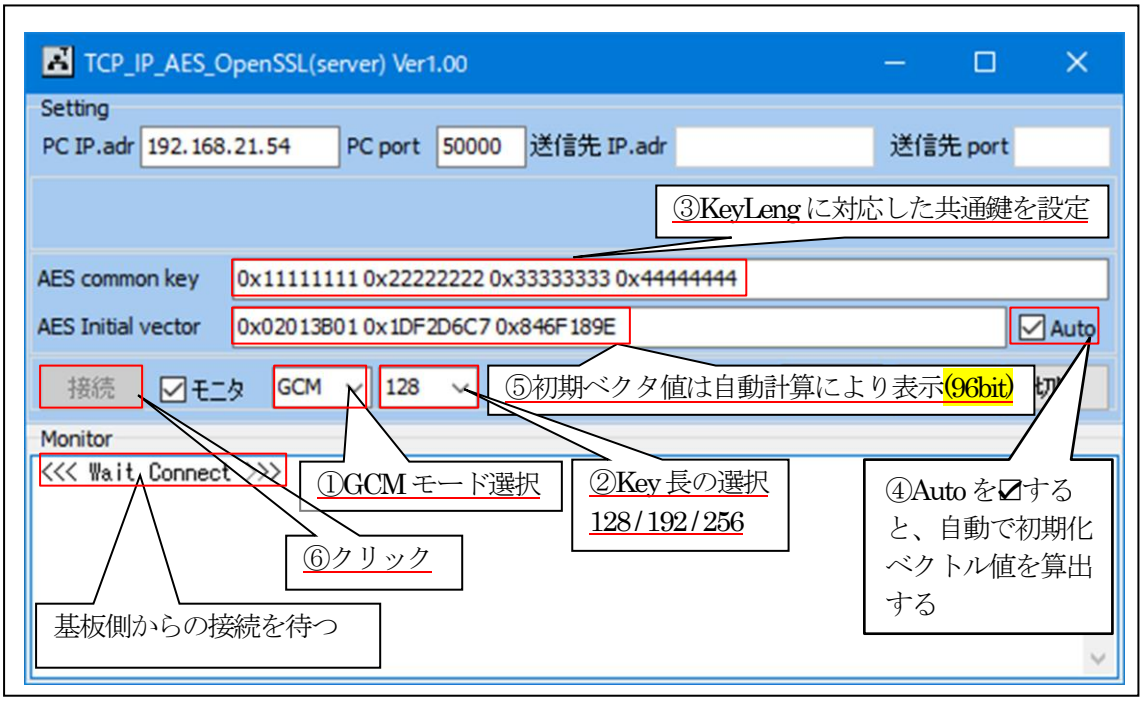
7－7．Windows PC 側のテスト用プログラムで動作確認（AES-GCM モード）

1）「TCP_IP_AES_OpenSSL」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_OpenSSL】を Read/Write 可能な適当なフォルダに Copy してからお使い下さい。



2）「TCP_IP_AES_OpenSSL」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 192.168.21.54⏎

②PC 側のポート番号を設定する。
ex) sendport 50000⏎

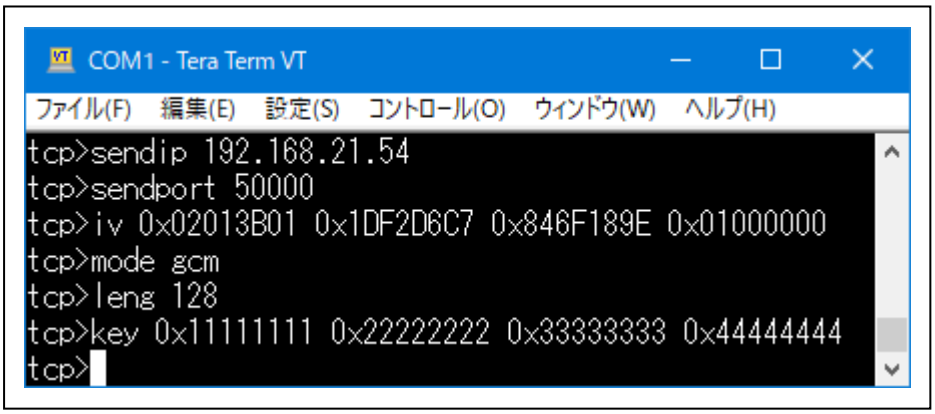
③「初期ベクタ値」を指定する。
PC 側の「UDP_IP_AES_OpenSSL」と同等の初期ベクタ値(AES Initial vector)にする。
ex) iv 0x02013B01 0x1DF2D6C7 0x846F189E 0x01000000⏎
AES-GCM の IV の 96bit 以降は、必ず、バイト表記で 0x00,0x00,0x00,0x01 にする必要があります。

④「GCM」（AES-GCM）モードを指定する。
ex) mode gcm⏎

⑤「Key Leng」を指定する。(128 / 192 / 256)
PC 側の「UDP_IP_AES_OpenSSL」と同じ Key 長にする。
ex) leng 128⏎

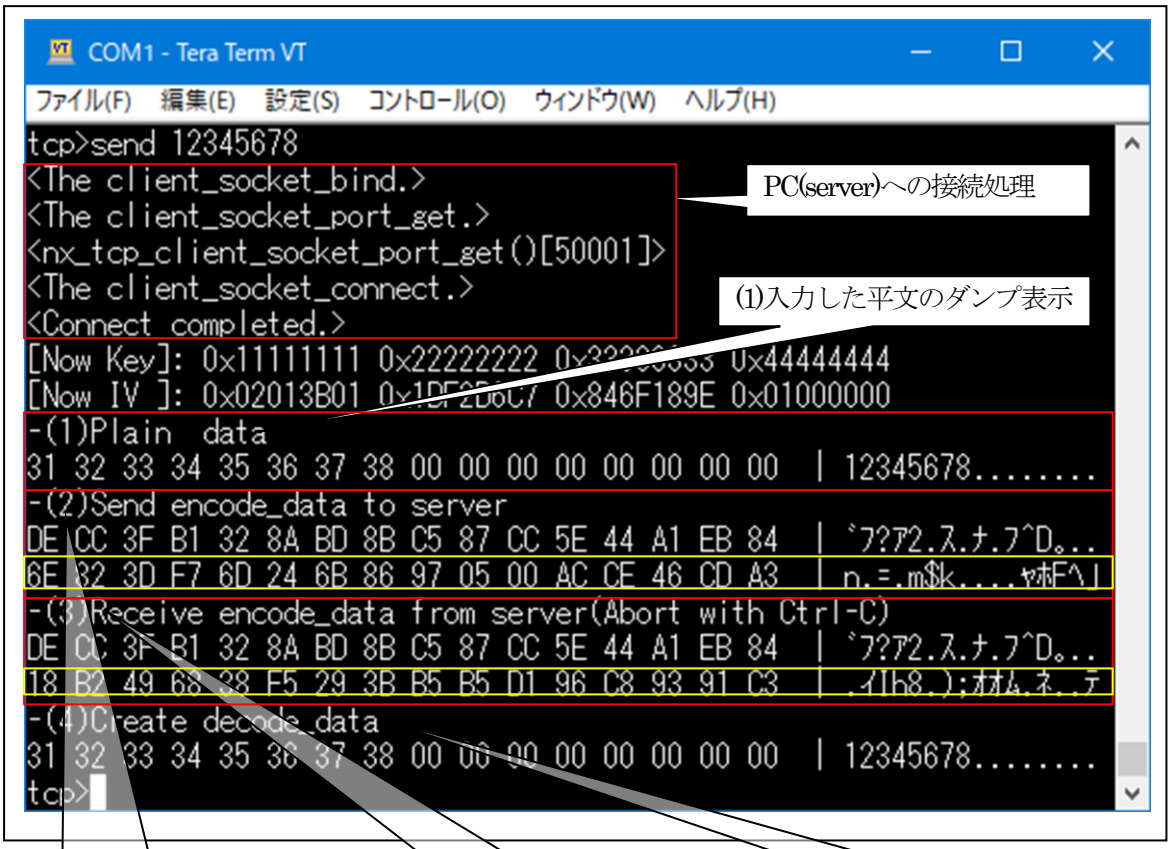
⑥「共通鍵」を指定する。
PC 側の「UDP_IP_AES_OpenSSL」と同等の共通鍵(AES common key)にする。
ex) key 0x11111111 0x22222222 0x33333333 0x44444444⏎

⑦実行画面



4）「基板」側から「PC」(server)側へ GCM 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



(2)入力した平文を暗号化（エンコード）したデータと認証用 TAG データ(16byte)を PC（Server）に送信したデータのダンプ表示

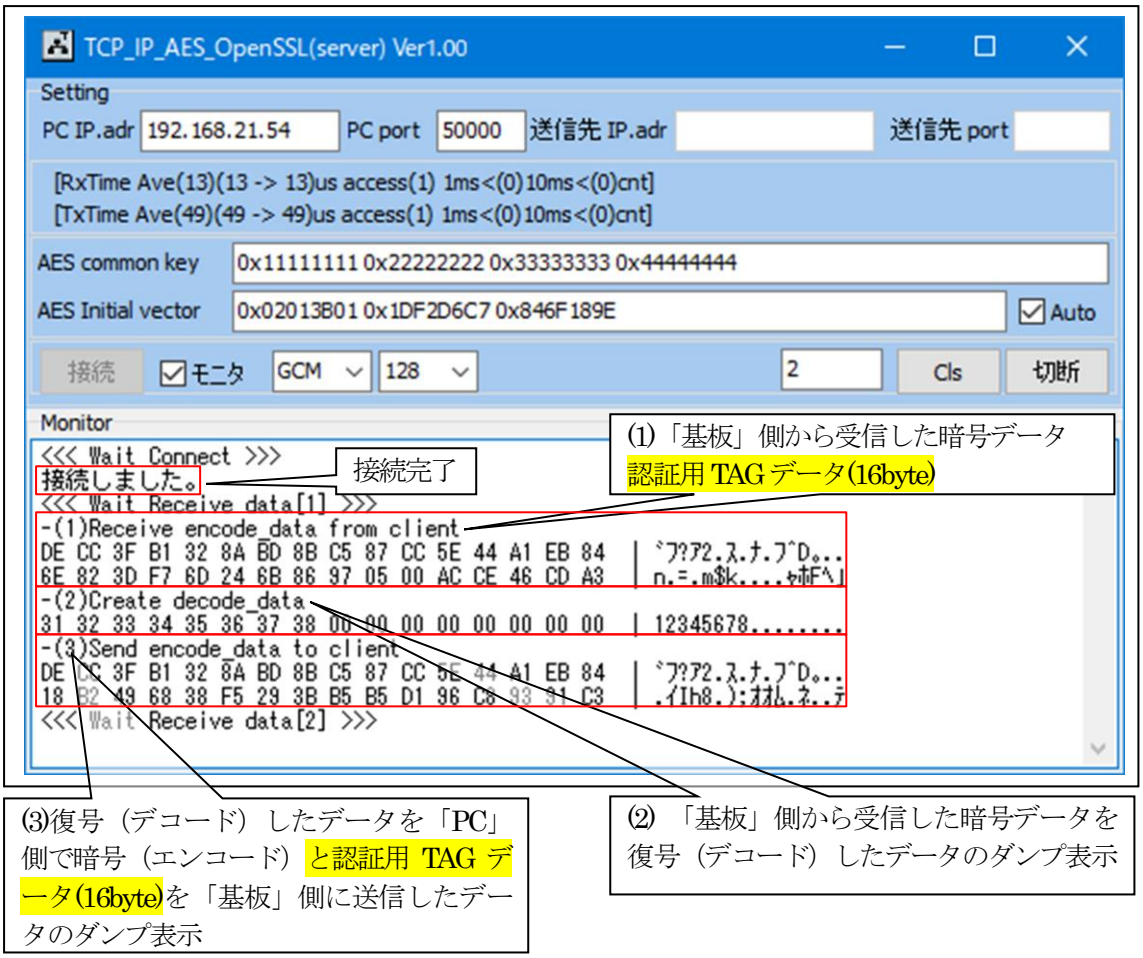
(3)PC（server）が受信した暗号文を復号（デコード）したデータと認証用 TAG データ(16byte)を PC（server）で再度暗号化（エンコード）したデータを受信したダンプ表示

(4)受信した PC からの暗号文を復号（デコード）したデータのダンプ表示

②「(1)Plain data」と「(4)Create decode_data」が同等のダンプ表示の場合、基板側と PC 側が同等な復号処理（デコード）であることの実証になります。

③「(2)Send encode_data to server」と「(3)Receive encode_data from server(Abort with Ctrl-C)」が同等のダンプ表示の場合、基板側と PC 側が同等な暗号処理（エンコード）であることの実証になります。

5）「TCP_IP_AES_OpenSSL」側の送受信を確認する。



①「(1)Receive encode_data from client」と「(3)Send encode_data to client」が同等のダンプ表示の場合、基板側と PC 側が同等な暗号処理（エンコード）であることの実証になります。

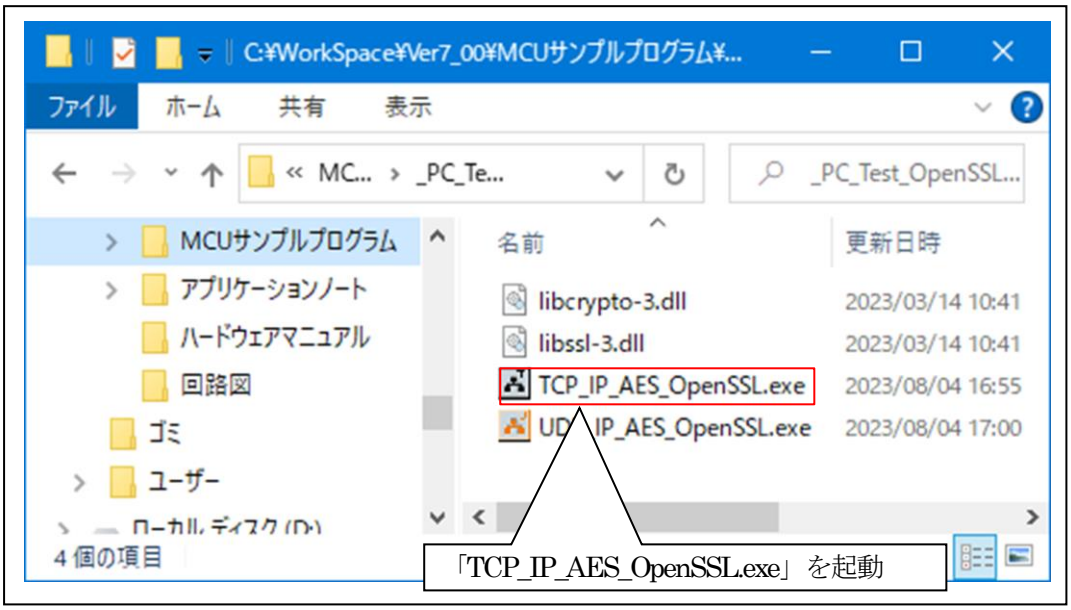
②「(2)Create decode_data」と「基板」側 TeraTerm 画面の「(4)Create decode_data」が同等のダンプ表示の場合、基板側と PC 側が同等な復号処理（デコード）であることの実証になります。

【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]"」

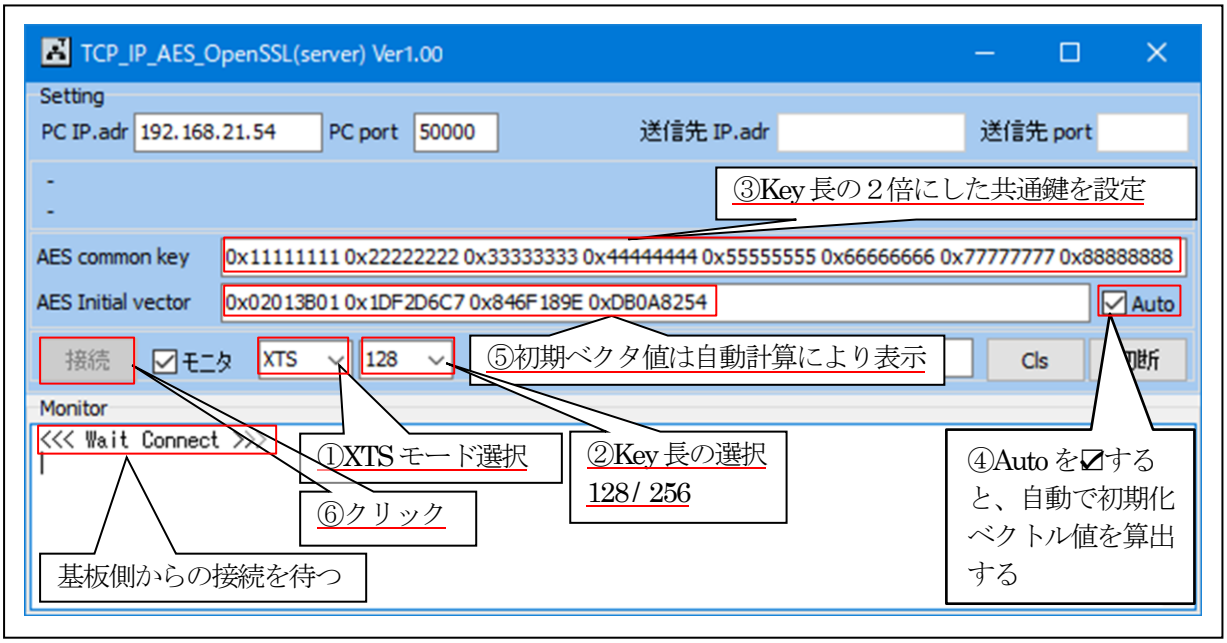
7－8．Windows PC 側のテスト用プログラムで動作確認（AES-XTS モード）

1）「TCP_IP_AES_OpenSSL」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_OpenSSL】を Read/Write 可能な適当なフォルダに Copy してからお使い下さい。



2）「TCP_IP_AES_OpenSSL」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 192.168.21.54↵

②PC 側のポート番号を設定する。
ex) sendport 50000↵

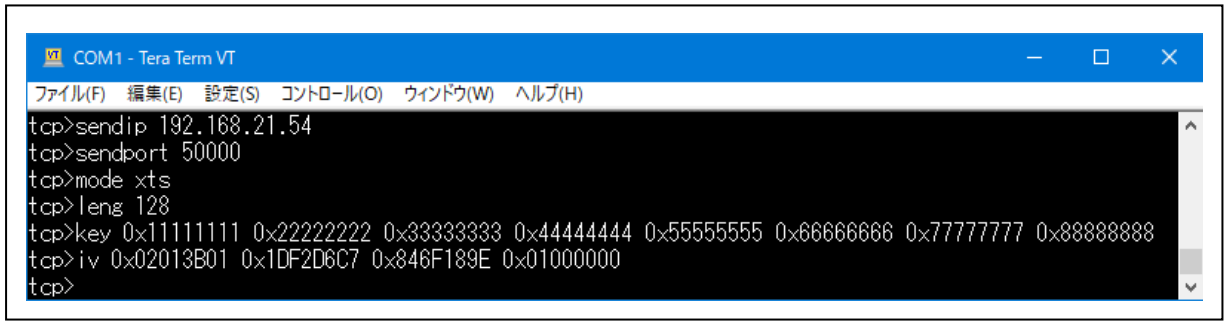
③「XTS」（AES-XTS）モードを指定する。
ex) mode ctr↵

④「Key Leng」を指定する。(128 / 256)
PC 側の「TCP_IP_AES_OpenSSL」と同じ Key 長にする。
ex) leng 128↵

⑤「共通鍵」を指定する。
PC 側の「UDP_IP_AES_OpenSSL」と同等の共通鍵(AES common key)にする。
ex) key 0x11111111 0x22222222 0x33333333 0x44444444 0x55555555 0x66666666 0x77777777 0x88888888↵

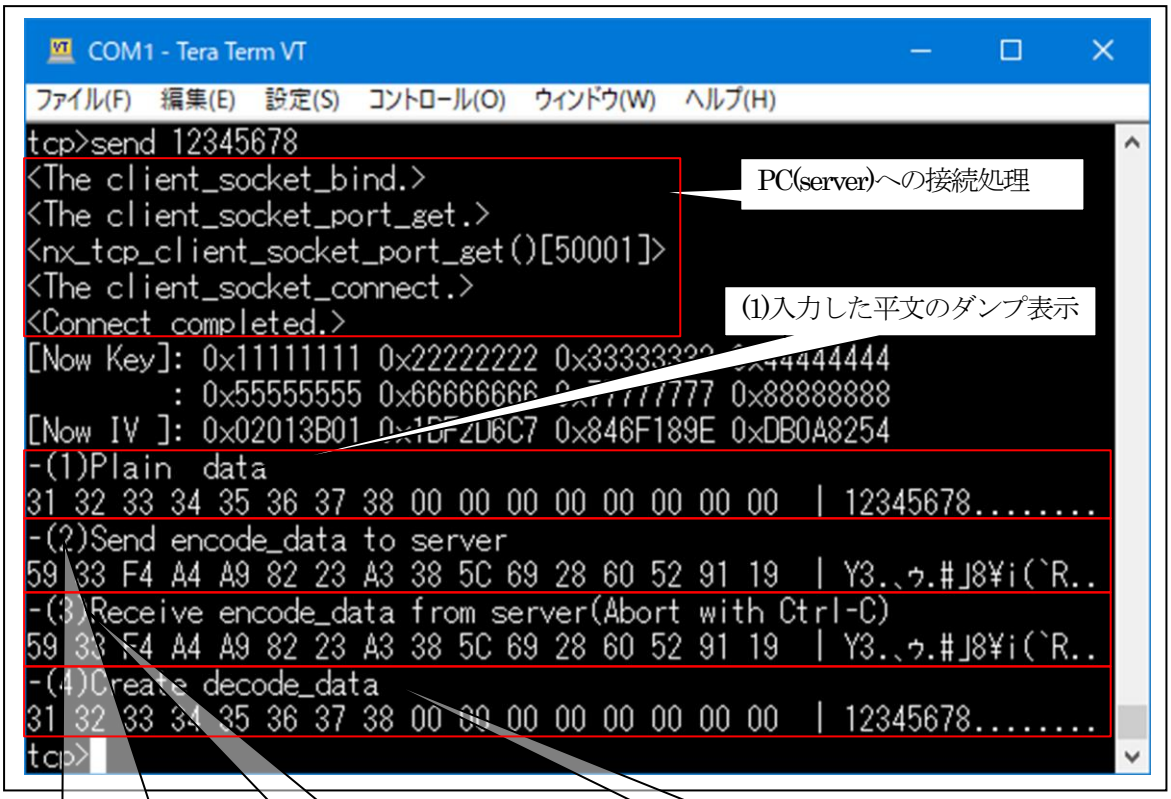
⑥「初期ベクタ値」を指定する。（HEX　数値）
PC 側の「TCP_IP_AES_OpenSSL」と同等の初期ベクタ値(AES Initial vector)にする。
ex) iv 02013B01 1DF2D6C7 846F189E DB0A8254↵

⑦実行画面



4）「基板」側から「PC」(server)側へ XTS 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）

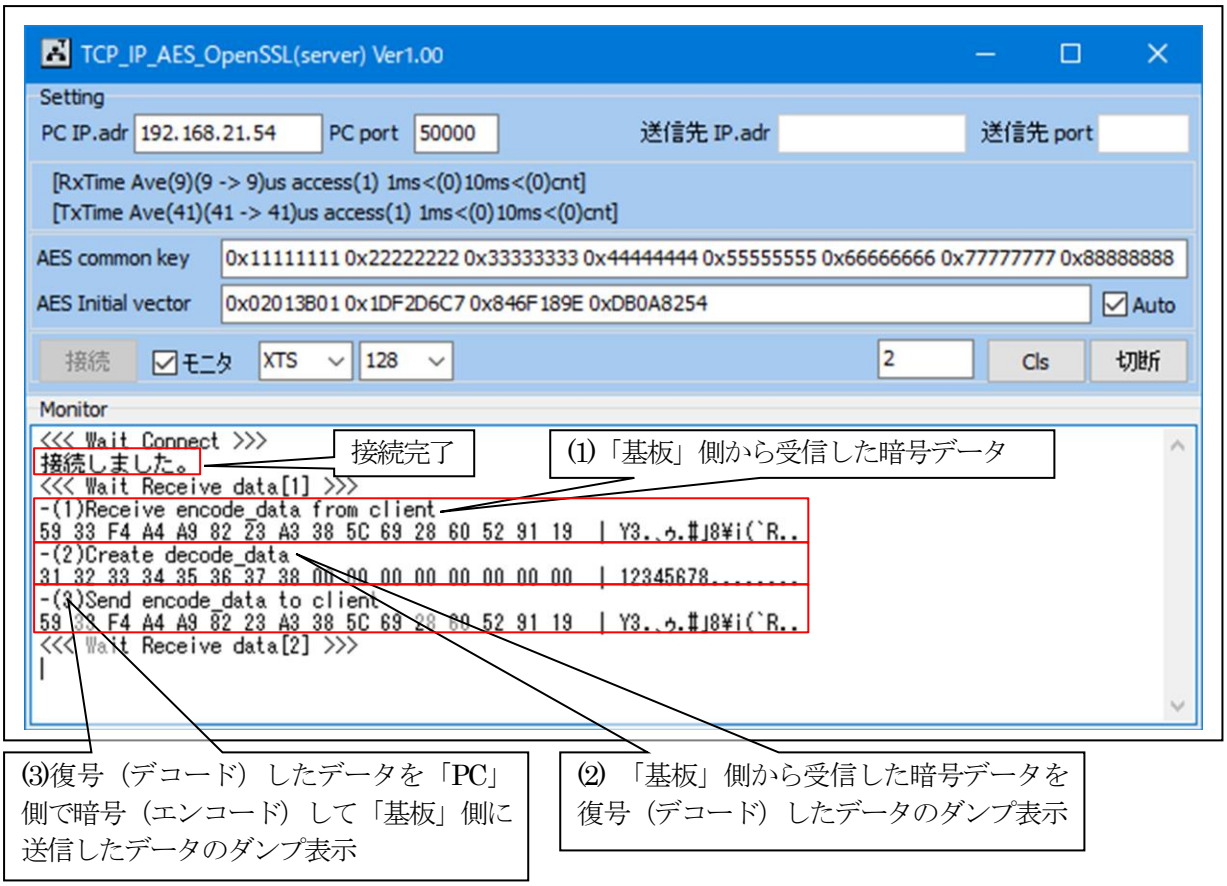
ex) send 12345678



②「(1)Plain data」と「(4)Create decode_data」が同等のダンプ表示の場合、基板側と PC 側が同等な復号処理（デコード）であることの実証になります。

③「(2)Send encode_data to server」と「(3)Receive encode_data from server(Abort with Ctrl-C)」が同等のダンプ表示の場合、基板側と PC 側が同等な暗号処理（エンコード）であることの実証になります。

5）「TCP_IP_AES_OpenSSL」側の送受信を確認する。



①「(1)Receive encode_data from client」と「(3)Send encode_data to client」が同等のダンプ表示の場合、基板側と PC 側が同等な暗号処理（エンコード）であることの実証になります。

②「(2)Create decode_data」と「基板」側 TeraTerm 画面の「(4)Create decode_data」が同等のダンプ表示の場合、基板側と PC 側が同等な復号処理（デコード）であることの実証になります。

【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]"」

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．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。

９．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