🎢 エーワン株式会社

Rev 1.00.00

### Renesas R5F565NE(CK-RX65N)用サンプル

#### (e2studio RX65N\_gnu\_dhcp\_tcp\_aes\_gt202)の説明

(e2studio Version:2024-04 / FreeRTOS Version 10.4.3 -rx-1.0.8)

- 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
ワークスペース名 FreeRTOS_sample_gnu_wifi_CK	概要 無線 WiFi-module(GT202- QCA4002) を使用した DHCP と TCP 通信のサ ンプル セキュリティ API Crypto ライブラリ(Mbed TLS)AES-CBC を使用し たサンプル (暗号・復 号)	プロジェクト名 RX65N_gnu_dhcp_tcp_aes_gt202 Free RTOS モードで動作 FreeRTOS-Plus-TCP (dhcp_client) TCP 通信(Client) (FreeRTOS_send) (FreeRTOS_recv)
		暗号・復号(AES-CBC) (MBEDTLS_AES_ENCRYPT) (MBEDTLS_AES_DECRYPT)

統合開発環境
Renesas e2studio(Version 2024-04)
FreeRTOS (Version 10.4.3 rx 1.0.8)
FreeRTOS-Plus-TCP(Version4.0.0)
Mbed TLS(Version3.5.2)
GCC for Renesas RX(Version 8.3.0.202405)
テキストファイル・エンコード(sjis)

ハード環境	
CK-RX65N(ルネサス製)	



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

	接続		×	COM番号は、 DC 側でいいアル通信E
ポート(P):	COM5	~	現在の接続を直線定(N)	□□ PC 側Cンリアル通信中 ↓ か釆号を指定する
スピード(E):	115200		AUT AND CHARACIA	
データ(D)	8 bit	~	キャンセル	115200BPS
パリティ(A):	none	~	11. 	8bit
ストップビット(s):	1 bit	~	ヘルプ(H)	none
フロー制御(F):	none	~		Ibit
0	_ ] ミリ秒/字((	) [0]	ミリ利沙/行(L)	の仕様にする。

#### ④端末の設定

端末サイズ(T) 90 X 50 図=ウィンドウサイズ 自動的に調整(W	改行コ 受信(R ズ(S): 送信(M	-F CR ~ CR+LF ~	<del>0</del> *	ZIL	<ul> <li>ータ使用時に CR</li> <li>ードがカットされ</li> <li>設定の場合は、</li> <li>信:LF</li> <li>にして下</li> </ul>
端末ID(I): VT10 応答(A):		ーカルエコー(L): 動切り替え(VT<->TE	ヘルプ =K)(U):	(H)	
漢字-受信(K)	漢字-送信(J)				
SJIS ~	SJIS ~	漢字-(~)(N)	^[\$B	4	
	11/ 12 day 1 (m)	attended (a)	of/ m		

赤枠の設定にする。



### 4. 動作構成



# <TCP\_IP\_AES\_OpenSSL.exe>

[Plain mode]

- 1. 平文を受信する。
- 2. 受信した平文をそのまま送信する。

### [AES-CBC mode]

- 1. 暗号文を受信する。
- 2. 受信した暗号文を復号して表示する。
- 3. 復号文を暗号化した文章を送信する。

### <RX65N\_gnu\_dhcp\_tcp\_aes\_gt202>

[Plain mode]

- 1. 平文を送信する。
- 2. 受信した平文を TeraTerm に表示する。

### [AES-CBC mode]

- 1. AES(CBC)暗号化した文章を送信する。
- 2. 受信した暗号文を復号して TeraTerm に表 示する。





ジャンパ		備考
J2	ショート	Current Measurement point for MCU
J15	オープン	Select debugger comms mode
J16	1-2ショート	DEBUG
J21	ショート	Enable USB boot mode
J22	オープン	Select USB boot mode power supply method
J11	オープン	Configures the MCU for normal boot m ode

# 5. 「RX65N\_gnu\_dhcp\_tcp\_aes\_gt202」サンプルの説明

<pre><freertos_sample_gnu_wifi_c< pre=""></freertos_sample_gnu_wifi_c<></pre>	<freertos_sample_gnu_wifi_ck></freertos_sample_gnu_wifi_ck>					
<rx65n_gnu_dhcp_tcp_aes< td=""><td colspan="3">DHCP / TCP 通信 / AES-CBC(暗号・復号)</td></rx65n_gnu_dhcp_tcp_aes<>	DHCP / TCP 通信 / AES-CBC(暗号・復号)					
_gt202>	サンプルプロジェクト					
<hardwaredebug></hardwaredebug>	RX65N_gnu_dhcp_tcp_aes_gt 202.elf	ELFファイル、デバッガで使用				
	RX65N_gnu_dhcp_tcp_aes_gt	MAPファイル、アドレス情報				
	RX65N_gnu_dhcp_tcp_aes_gt	モトローラーHEX ファイル				
	その他					
<pre>sm&gt;</pre>	<freertos></freertos>	BTOS カーネル				
	<pre><freertos-plus-tcp></freertos-plus-tcp></pre>	TCP/IP Хタック				
	<pre><mbedtls decelopment=""></mbedtls></pre>	暗号・復号ライブラリ				
	<a href="https://www.autoparticity.com">AWS aes&gt;</a>	AES-CBC				
	aws aes c					
	aws_aesh	aws aesc のヘッダー				
	<pre></pre>	未使用				
	<pre></pre>					
	tcp aes task.c					
	task_function.h	tcp_aes_task.c のヘッダー				
	<pre></pre>	スタートアップ				
	freertos_object_int.c	tcp_aes_taskの作成				
	freertos.start.c	CMTOの初期化と MAIN_TASK の起動				
	freertos.start.h	reertos.start.cのヘッダー				
	<smc_gen></smc_gen>	スマートコンフィグレータによ り作成				
	<wifi_gt202></wifi_gt202>	GT202用ドライバソース一式				
	demo_printf.c	コンソール入出力処理のソース				
	demo_printf.h	demo_printf.c のヘッダー				
	hardware_setup.c	周辺IOデバイス初期化ソース				
	hardware_setup.h	hardware_setup.cのヘッダー				
	main_task.c	メインタスク				
	sf_wifi_nsal_api.h	wifi コントロールのヘッダ				
	FreeRTOSConfig.h	FreeRTOS のコンフィグレーシ				
		ョンファイル				
	FreeRTOSIPConfig.h	FreeRTOS+TCP のコンフィグ				
		レーションファイル				
	linker_script.ld	リンカ用スクリプトファイル				
RX65N_gnu_ dhcp_tcp_aes_gt202.scfg	スマートコンフィグレータの	管理ファイル				
その他	自動生成ファイル					

5-1.フォルダ構成とファイル名【<ホルダ名>を示す】



5-2. Macro Defines の説明

Macro Name	値	説明
ipconfigUSE_DHCP	0	DHCP Client Disable
		◎ヘッダーファイルに直接 IP アドレスを記述
		FreeRTOSConfig.h :
		/* IP address configuration. */
		#define configIP_ADDR0 192
		#define configIP_ADDR1 168
		#define configIP_ADDR2 21
		#define configIP_ADDR3 95
	1	DHCP Client Enable

5-3. サンプルの動作説明(基板側 CK-RX65N)

1) DHCP 無効時(ipconfigUSE\_DHCP = 0) Term 画面

< 1 > ["The FreeRTOS\_IPInit starting"]

<2> ["The network is up waiting"]

<成功画面>

IPアドレス確立により、

基板上のLED3(緑色)を100msec毎に点滅・LED4(青色)IRQ0割り込み毎に点滅



<失敗画面>

IPアドレス未確立により、基板上のLED6(赤色)を100msec毎に点滅



🎾 エーワン株式会社

```
2) DHCP 有効時(ipconfigUSE_DHCP = 1)
Term 画面
<1>「'The FreeRTOS_IPInit starting'」
<2>「'The network is up waiting'」
```

```
<成功画面>
```

IPアドレス確立により、

基板上のLED3(緑色)を100msec毎に点滅・LED4(青色)IRQ0割り込み毎に点滅



<失敗画面>

IPアドレス未確立により、基板上のLED6(赤色)を100msec毎に点滅





3) TCP/IP 送受信

<TCPAES Thread>

	5 - Tera Ter	m VT					×
ファイル(F)	搞集(E)	設定(S)	コントロール(0)	ウィンドウ(W)	ヘルブ(H	1)	
<the net<="" td=""><td>twork i</td><td>s up wa</td><th>aiting&gt;</th><td></td><td></td><td></td><td>^</td></the>	twork i	s up wa	aiting>				^
Successi [MAC ADI [MY IP.a [SUBNET [MTU siz [SERVER [TCP poi [UDP poi {The net {Start f	fully a DRESS]: adr ]: mask]: ze ]: port]: port]: rt ]: twork i FreeRTO	ssigned 84-5D 192.10 255.25 1500 50000 50000 50001 50002 s up ar S TCP 2	d by DHCP a -D7-4A-5B-B 38.21.54 55.255.0 nd running> AES RX65N[c	ddress(et 0 :lient]>	h0)		
AES_mode Press ar [Press	e[PLAIN ny key. 'C' key	] SERVE  to cha	ER_IP[192.1 ange AES mo	68.21.53] de/'L' ch	ange I	ensth]	
AES_mode Press ar {Press	e[CBC] ny key. 'C' key	KevLena  to cha	sth[128] SE ange AES mo	RVER_IP[1 de/'L' ch	92.168 ange	.21.53 ength]	3]

表示項目	表示内容	説明
AES_mode[x]	PLAIN	送受信モードの指定
	CBC	◎ "C"キー入力によりモード変更
		PLAIN -> CBC -> PLAIN
		◎変数のフラグにより指定
		aes.c: : int AES_crypto_mode;
		0 : PLAIN // 平文モード
		1:CBC //AES-CBC モード暗号・復号
KeyLength[x]	128	AES-CBC モード時の Key ビット時の指定
	192	◎ "L"キー入力により Key ビット変更
	256	128 > 192 > 256 > 128
		◎変数の数値により指定
		aes.c: : int AES_crypto_bit;
		128 : Key ビット長が 128bit
		192 : Key ビット長が 192bit
		256 : Key ビット長が 256bit
SERVER_IP[xxx.xxxx.xxxx]	固定	送信先(PC 側)IP アドレス
		◎define にて指定
		tcp_aes_thread_entryc:
		#define SERVER_IP IP_ADDRESS(192,168,xx,xx)

5-4. Windows PC 側のテストプログラムで動作確認(PLAIN(平文)モード)

1)「TCP\_IP\_AES\_OpenSSL.exe」を起動する。(各モード共通)

プログラム場所【¥\_PC\_Test\_OpenSSL】サンプルの解凍ホルダ



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。

TCP_IP_AES_OpenSSL(server) Ver1.00			×
Setting         PC IP.adr         IPC 側の IP アドレス自動検出         PC 側の port 固定	送信劳	ë port	
AES KEY File Name AES IV File Name	-1673		
接続 Dモニタ Plain マ ①Plain モード選択 0 Monitor	d	s	切胜斤
Wait Connect >>>>     ②クリック     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。

表示項目	説明
AES_mode[PLAIN]	送受信モードの指定 ◎変数のフラグにより指定
	aes.c: : int AES_crypto_mode = PLAIN; // 0=PLAIN
SERVER_IP[192.168.21.53]	送信先(PC 側)IP アドレス ◎define にて指定 tcp_aes_thread_entry.c: #define SERVER_IP IP_ADDRESS(192,168,21,53)

ファイル(F) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) ヘルプ(H) AES_mode[PLAIN] SERVER_IP[192.168.21.53]	COM	4 - Iera Iei					<u></u>
ES_mode[PLAIN] SERVER_IP[192.168.21.53]	ファイル(F)	編集(E)	設定(S)	コントロール(0)	ウィンドウ(W)	ヘルプ(H)	
	ES_mod	e[PLAIN	] SERVE	ER_IP[192.1	68.21.53]		^

4) 基板側から PC(server) 側へ平文テキストを送信する。

①disconection の場合は、PC(server)との Conection 処理を実行

77/ 1L/E)	/編集/E)	铅宗(S)	72b0=11(0)	ウィンドウ(W)	ALT/H		
(F1)V(F)	福来(C)	67. AE (3)	1/10-10(0)	2121-2(W)	(n) (viv)		/
ES_mode	e[PLAIN	] PC(se	erver)との接続	確認中			-
ress a	ny key.		ss 'C' key	to change	AES mo	de	}
Mait c	lient_s	ocket_	connect.>				
Mait c	lient s	ocket	connect.>				
Wait c	lient_s	ocket	connect.>	DC(aaraan)	しの技法	±7	
<u>~</u>	in a sum l	at ad s		PU(server)	とい接続	元J	

☆接続が失敗した場合は「Ctrl+C」Key-Pushで中断する。



```
②接続完了後、「基板」側から PC(server) 側へ平文テキストを送受信する
```

アイル(F) #	<b>冒集(E)</b>	設定	:(S)	עב	<u>+0-</u>	JL(0	) 5	ハント	・ウ(W	0	ヘルフ	(H)				-
Tabam 23	DI ATN		①何	らか	のF	Key Z	をPu	ısh								
ress any	key.	{F	res	s	'C'	key	/ to	b d	hana	se /	¥ES	mode	]			
Send Pla	inTex	t le	engt	th(1	28)	) >										
68 69	73 20	69	73	20	61	20	76	65	72	79	20	69	(1)P	Cへのì	送信デー	-タ
0 70 6F	72 74	61	6E	74	20	73	65	63	72	65	74	20	mpc	ortant	.secr	et.
1 65 73	13 61	6/	65	21	20	49	20	11	69	60	50	20	mes	sage	.1.wi	11.
74 60	20 69 65 65	74	20	00	01	20	20	00	05	03	70	79	USE	e. 11.1	or.en	cry
60 6E	0F 0E 6E 20	20	65	0E 70	74	20	04 6E	67	20	60	19	20		on.ar	nd.dec	ryp
62 65	20 66	75	71	75	72	65	25	20	62	70	20	11	1 the	fut	ire h	v A
6F 65	20 43	ĥF	72	70	6F	72	61	74	69	6F	6F	20		Corr	orati	00
ecv Pla	inTex	t le	eng	th()	28	)>	01	1.4	00	VI.	VL.	20	1 One		oraci	011.
68 69	73 20	69	73	20	61	20	76	65	72	79	20	69	(2) P	Cから	の受信う	データ
70 6F	72 74	61	6E	74	20	73	65	63	72	65	74	20	mpo	ortant	.secr	et.
65 73	73 61	67	65	21	20	49	20	77	69	6C	6C	20	mes	sage!	.I.wi	11.
73 65	20 69	74	20	66	6F	72	20	65	6E	63	72	79	use	e.it.f	or.en	сгу
74 69	6F 6E	20	61	6E	64	20	64	65	63	72	79	70	pti	on.ar	nd.dec	гур
69 6F	6E 20	74	65	73	74	69	6E	67	20	69	6E	20	tic	on tes	sting.	in.
68 65	20 66	/5	74	75	12	55	ZE	20	62	/9	20	41	the	e. futu	ireb	у.А
0E 00	20 43	٥F	12	70	οr	12	01	74	69	ЮF	0E	20	I one	e.lorp	orati	on.
S madal	DI ATNI	1 0	TOT	IDI	102	5 16	20 /	1.6	27							
O_HOUEL	LAIN		501-		0,		10.2		101			2	1			

☆受信処理が失敗した場合は「Ctrl+C」Key-Pushで中断する。

5) 基板側の原文保存場所(各モード共通)

モジュール名	変数名
tcp_aes_thread_entry.c	static UCHAR *src_str={ //テスト用原文
	"This is a very important secret message!"
	"I will use it for encryption and decryption testing"
	"in the future. by Aone Corporation"
	<i>}</i> ,

# 6)「TCP\_IP\_AES\_OpenSSL」側の送受信を確認する。

PC IP.adr 192.168.2	21.53 PC p	ort 50000	送信先 IP.adr		送信先 port	
[RxTime Ave(5)(5 - [TxTime Ave(42)(42	> 5)us access(1 ? -> 42)us acces	) 1ms<(0)10 xs(1) 1ms<(0	ms<(0)cnt] )10ms<(0)cnt]			
ES KEY File Name						老照
ES IV File Name						「製田
· 接続 ☑ モニタ	Plain v	~		2	Cls	切断
Monitor						
(1) Receive plain (1) Receive plain (4 68 69 73 20 68 (0 70 6F 72 74 61	data[1] >>> data from c 73 20 61 20 6E 74 20 73 65 21 20 49	1 ient 76 65 72 65 63 72 20 77 69	(1 79 20 69 T 65 74 20 m	)「基板」側から( his.is.a.very.i portant.secret.	の受信データ	





7)「TCP\_IP\_AES\_OpenSSL」その他の操作(各モード共通)

Setting									
PC IP.adr	192.168.21.53	PC port	50000	送信先 IP.	ədr		送信党	Eport	
[RxTime A [TxTime A	lve(4)(3 -> 5)us ac lve(33)(25 -> 42)us	cess(2) 1ms s access(2)	s<(0)10n 1ms<(0)	ns<(0)cnt] 10ms<(0)cn	tj				
AES KEY FIL	e Name								藍照
AES IV File	Name								100月18日
接続 Monitor		~	~			3	0	6	切助
6D 70 6 6D 65 7 75 73 65 70 74 69	✓ 「Monitor」 画 20 63 74 20 66 6F 6E 20 61 6E 6E 20 74 65 73	面に送受 6F 72 20 64 20 64 74 69 6E	信ダン 65 6E 6 65 63 7 67 20 6	プ表示 3 /2 /9 2 79 70 9 6E 20	mportant message! use.it.fo ption.and tion.test	Monitor .encry .decryp ing.in.	 :」表示	Fを消ま	11

8) **TCP\_IP-Port**を「切断」する。(各モード共通)

security							
PC IP.adr	192.168.21.53	PC port	50000	送信先 IP.adr		送信先 port	
[RxTime [TxTime	Ave(4)(3 -> 5)us a Ave(33)(25 -> 42)u	ccess(2) 1m us access(2)	s<(0)10 1ms<(0	ms<(0)cnt] )10ms<(0)cnt]			
AES KEY F	ile Name						些耻
AES IV File	Name						「新日本
接続 Monitor	→ ELA Plain	~	~		3	Cls	切断
6D 65 73 75 73 65 70 74 69	73 61 67 65 21 20 69 74 20 66 6F 6E 20 61 6E 6E 20 74 65 73	20 49 20 6F 72 20 64 20 64 74 69 6E	77 69 6 65 6E 6 65 63 7 67 20 6	C 6C 20 me 3 72 79 us 72 79 70 pt 3 8E 20 ti 4 20 41 th	ssage!.I.will. e.it.for.encry ion.and.decryp on.testing.in.	<u> </u>	<u>7</u>

5-5. 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.exe」を起動する。
- 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。

TCP_IP_AES_OpenSSL(server) Ver1.10	0.000	۵	×
Setting PC IP.adr 192.168.21.53 PC port 50000 送信先 IP.adr	送信	先port	
共通鍵の保存ファイル 初期ベクタ値の保存ファイル			
AES KEY File Name AES_KEY.txt AES IV File Name AES_IV.txt ①CBCモード選	択		参照
1865     ○モニタ     CBC     128       Monitor     128/192/256		Cls	切断
[KEY]:0x11 0x11 11 0x11 0x22 0x22 0x22 0x22 0x	44 0×44 54 0×82	0×44 ( 0×0A (	0×44 0×DB
③クリック     登録されている「共通鍵」と「       基板側からの接続を待つ	初期べ	クタ値 <u></u>	<u></u>
			1

3)「AES KEY.txt」「AES IV.txt」の説明

「AES_KEY.txt」 共通鍵テキストファイル	
// default aes_common_key 共通鍵 128bit   192bit   256bit	
//コメント行は、//のみの使用にして下さい。	
0x11,0x11,0x11,0x11,0x22,0x22,0x22,0x22,	,0x33,0x44,0x44,0x44,0x44, //128bit
0x55,0x55,0x55,0x55,0x66,0x66,0x66,0x66,	$// \uparrow + 192 \text{bit}$
0x77,0x77,0x77,0x77,0x88,0x88,0x88,0x88,	$//\uparrow + 256 bit$

☆共通鍵を変更する場合は、基板側と同等の鍵を適当なエディタで変更する。

「AES\_IV.txt」初期ベクタ値テキストファイル // default aes\_initial\_vect 初期化ベクタ // コメント行は、//のみの使用にして下さい。 0x01,0x3B,0x01,0x02,0xC7,0xD6,0xF2,0x1D,0x9E,0x18,0x6F,0x84,0x54,0x82,0x0A,0xDB

☆初期ベクタ値を変更する場合は、基板側と同等の初期ベクタ値を適当なエディタで変更する。

### 4) 基板側の各項目の確認と設定。

表示項目	説明
AES_mode[CBC]	送受信モードの指定
	◎変数のフラグにより指定
	aes.c: : int AES_crypto_mode = CBC; // 1=CBC
KeyLength[128]	AES-CBC モード時の Key ビット時の指定
	◎変数の数値により指定
	$aes.c: : int AES_crypto_bit = 128;$
SERVER_IP[192.168.21.53]	送信先(PC 側)IP アドレス
	◎define にて指定
	tcp_aes_thread_entry.c :
	#define SERVER_IP IP_ADDRESS(192,168,21,53)

共通鍵データの保存モジュールと変数 【aes.c】	
uint8_tAES_key[32] ={ // default aes_common_key {	共通鍵 128bit   192bit   256bit
0x11,0x11,0x11,0x11,0x22,0x22,0x22,0x22,	:33,0x33,0x33,0x44,0x44,0x44,0x44,// 128bit
0x55,0x55,0x55,0x55,0x66,0x66,0x66,0x66,	$//\uparrow$ +192bit
0x77,0x77,0x77,0x77,0x88,0x88,0x88,0x88,	$//\uparrow + 256 bit$
};	

初期ベクタ値データの保存モジュールと変数【aes.c】 uint8\_tAES\_iv[16] = {// default aes\_initial\_vect 初期化ベクタ 0x01,0x3B,0x01,0x02,0xC7,0xD6,0xF2,0x1D,0x9E,0x18,0x6F,0x84,0x54,0x82,0x0A,0xDB };

ファイル(F)	編集(E)	設定(S)	コントロール(0	) ウィント	ドウ(W)	ヘルプ(ト	<del>1</del> )		
ES mode	FCRC1	Keyl en	+b[128]	SERVER	TPF1	92 168	21	531	



<sup>5)</sup> 基板側から PC(server) 側へ CBC 暗号テキストを送信する。

🚾 COM5 - Tera Term VT	– 🗆 🗙
ファイル(F) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) ヘルブ(H)	
AES_mode[CBC] KeyLength[ <u>128</u> ] ①何らかの Key を Push	^
Press any key[Press 'C' key to change AES mode.	}
(Plainlext length(128) > 54 00 00 70 00 00 00 01 00 70 05 70 70 00 00	
	(1)基板側に保存してあ
6D 70 6F 72 74 61 6E 74 20 73 63 63 72 63 74 20 6D 65 72 72 61 67 65 21 20 49 20 77 69 66 66 20	る原文のタンフ表示
75 73 65 20 69 74 20 66 6F 72 20 65 6F 63 72 79	use it for encry
70 74 69 6F 6F 20 61 6F 64 20 64 65 63 72 79 70	ption and decryp
74 69 6F 6E 20 74 65 73 74 69 6E 67 20 69 6E 20	tion.testing.in.
74 68 65 20 66 75 74 75 72 65 2E 20 62 79 20 41	the.futureby.A
6F 6E 65 20 43 6F 72 70 6F 72 61 74 69 6F 6E 20	one.Corporation.
<send encrypttext="" length(128)=""></send>	
44 50 E8 F0 DA 72 DE 0D 8E D4 8F 1D B3 04 73 C0	(2)原文を暗号化して
29 30 C8 09 06 DE 48 C2 90 52 08 49 8B 69 63 E1	PC(server)側に送信した
64 32 IC ID CI 80 80 18 17 45 61 87 38 82 25 II 29 64 24 60 26 26 60 42 07 66 45 21 02 06 74 26	暗号文のダンプ表示
30 0A ZA UD 30 Z3 FU 4Z 07 00 43 31 UZ UF 74 3E   Ex xx Eq R5 xq 72 00 D5 20 ED 12 ED 9D X2 0R X2	
17 E0 26 1E 9E 05 24 ED 88 18 69 04 27 89 47 78	2 +\$ i ' 7
30 8D 5D 80 0F C9 A1 82 68 81 40 A8 8C 27 F2 80	0 ]- / /h @r ' -
13 5D 17 A6 EE 18 DA 13 8F 91 7D 06 53 3A 33 BA	.1.₹↓].S:3⊐
<recv encrypttext="" length(128)=""></recv>	
44 50 E8 F0 DA 72 DE 0D 8E D4 8F 1D B3 04 73 C0	(3)PC(server)が受信した
29 30 C8 09 06 DE 48 C2 90 52 08 49 8B 69 63 E1	暗号文を復号した文章
64 32 1C 1D C1 8D 80 18 17 45 61 B7 3B B2 25 11	を PC(server)側で暗
36 6A ZA CB 36 25 FU 42 07 66 45 31 UZ UF 74 3E	号化した暗号文を受信
E4 44 F8 60 48 72 88 60 30 ED 13 FD 86 A3 06 43	したダンプ表示
20 20 50 50 0F 52 03 24 FD 88 10 03 08 27 85 87 78	0 ]~ / /h @/ '~
13 5D 17 A6 FF 18 DA 13 8F 91 7D 06 53 3A 33 BA	1 7 L 1 S:31
<pre><decrypttext length(128)=""></decrypttext></pre>	111/11/11/11/10/04
54 68 69 73 20 69 73 20 61 20 76 65 72 79 20 69 💊	(4)受信した PC(server)か
6D 70 6F 72 74 61 6E 74 20 73 65 63 72 65 74 20	らの暗号文を復号化し
6D 65 73 73 61 67 65 21 20 49 20 77 69 6C 6C 20	た複合文のダンプ表示
75 73 65 20 69 74 20 66 6F 72 20 65 6E 63 72 79	about cutor cutory
70 74 69 6F 6E 20 61 6E 64 20 64 65 63 72 79 70 1	ption.and.decryp
74 69 6F 6E 20 74 65 73 74 69 6E 67 20 69 6E 20 1	tion.testing.in.
AF AF A5 20 00 73 74 73 72 03 2E 20 02 73 20 41 1	one Corporation
	one.corporation.

☆受信処理が失敗した場合は「Ctrl+C」Key-Pushで中断する。

 ②「(2)Send EncrptText」と「(3)Recv EncrptText」が同等の場合、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る。

 <sup>「(1)</sup>PlainText」と「(4)DecryptText」が同等の場合、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る。



6)「TCP\_IP\_AES\_OpenSSL」側の送受信を確認する。

Setting			
PC IP.adr 192.168.21.53	PC port 50000 送信先 IP.ad	ir 📃	送信先 port
[RxTime Ave(3)(3 -> 3)us acc [TxTime Ave(21)(21 -> 21)us	cess(1) 1ms<(0)10ms<(0)cnt] access(1) 1ms<(0)10ms<(0)cnt]		
AES KEY File Name AES_KEY.t	bxt		参照
AES IV File Name AES_IV.tx	t		参照
接続 ☑ モニタ ⊂ВС	✓ 128 ✓	2	Cls 切断
Ionitor			
K         Wait Connect         >>>           接続しました。         K         Wait Receive data[1]           (44 50 E8 F0 DA 72 DE 0D 8         20 30 C8 03 06 DE 48 C2 8           23 30 C8 03 06 DE 48 C2 8         24 32 1C 1D C1 8D 80 18           36 6A 2A CB 36 25 FD 42 (           E4 44 F8 65 43 72 88 B5 8           30 8D 5D B0 0F C3 A1 B2 8           30 8D 5D B0 0F C3 A1 B2 8           31 5D 17 A6 EE 18 DA 13 8           -(2)Create decode data           54 68 63 73 20 63 73 20 6           50 73 65 20 63 74 20 66 8           74 63 6F 6E 20 74 65 73 73           65 65 20 43 6F 72 70 6           74 63 6F 6E 20 74 65 73           75 73 65 20 66 75 74 75 73           74 68 65 20 66 75 74 75 73           75 6E 65 20 43 6F 72 70 6           (3)Send encode data to           (44 50 E8 F0 DA 72 DE 0D 8           29 30 C8 03 06 DE 48 C2 8           34 32 1C 1D C1 8D 80 18           36 6A 2A CB 36 54 37 28 8 B5           37 60 26 F 72 72 88 B5           38 65 43 72 88 B5           39 0 C8 05 06 0F C3 41 B2	>>>           from client           BE D4 8F 1D B3 04 73 C0           90 52 08 49 8B 69 63 E1           17 45 61 B7 3B B2 25 11           07 66 45 31 D2 DF 74 3E           30 ED 13 FD 8B A3 06 43           88 18 69 0A 27 89 A7 78           88 14 00 A8 8C 27 F2 B0           BF 91 7D 06 53 3A 33 BA           61 20 76 65 72 79 20 69           20 49 20 77 69 6C 6C 20           6F 72 20 65 6E 63 72 79 70           74 69 6E 67 20 89 6E 20           72 65 2E 20 62 79 20 41           6F 72 61 74 69 6F 6E 20           74 63 6E 172 79 80 64 30 64 33           80 64 85 10 B3 04 73 C0           90 52 08 49 8B 69 63 63 E1           17 45 61 B7 3B B2 25 11           107 66 45 31 D2 DF 74 3E           30 ED 13 FD 88 A3 06 43           81 86 90 A 27 89 A7 78           88 81 86 9 A2 78 84 778           88 81 40 A8 8C 27 F2 B0	1)基板側から受信 ンプ表示 0.1.1 2)受信した暗号文 号文のダンプ表示 5 5 3)復号化した復号 載板側に送信した暇 表示	した暗号文のダ を復号化した復 文を暗号化して 音号文のダンプ

- 「(1)Receive encode\_data from client」と「(3)Send encode\_data to client」が同等の場合、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る。
- ②「(2)Create decode\_data」と「基板」側の「(4)DecryptText 」が同等の場合、基板側と PC 側 が同等な復号処理(デュード)であることの実証になる。

- 6. 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。
- 7. 商標
  - ・e2studio・RX65N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。
  - ・CK-RX65Nは、ルネサス エレクトロニクス株式会社の商品名です。
  - ・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。
- 8. 参考文献
  - ・「RX65N 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
  - ・「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
  - $\cdot$  [FreeRTOS]

Amazon.com, Inc.

2011 エーワン株式会社

- ・ルネサス エレクトロニクス株式会社提供のサンプル集
- ・その他

 $\mp 486-0852$ 

愛知県春日井市下市場町6-9-20 エーワン株式会社 https://www.aone.co.ip

