

## Renesas R5F565NE(CK-RX65N)用サンプル

### (e2studio RX65N\_ccrx SMTP\_TLS\_uselib)の説明

#### (e2studio Version:2024-04 / Azure RTOS Version 6.2.1 rel-rx-2.0.0)

#### 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ください。
- 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

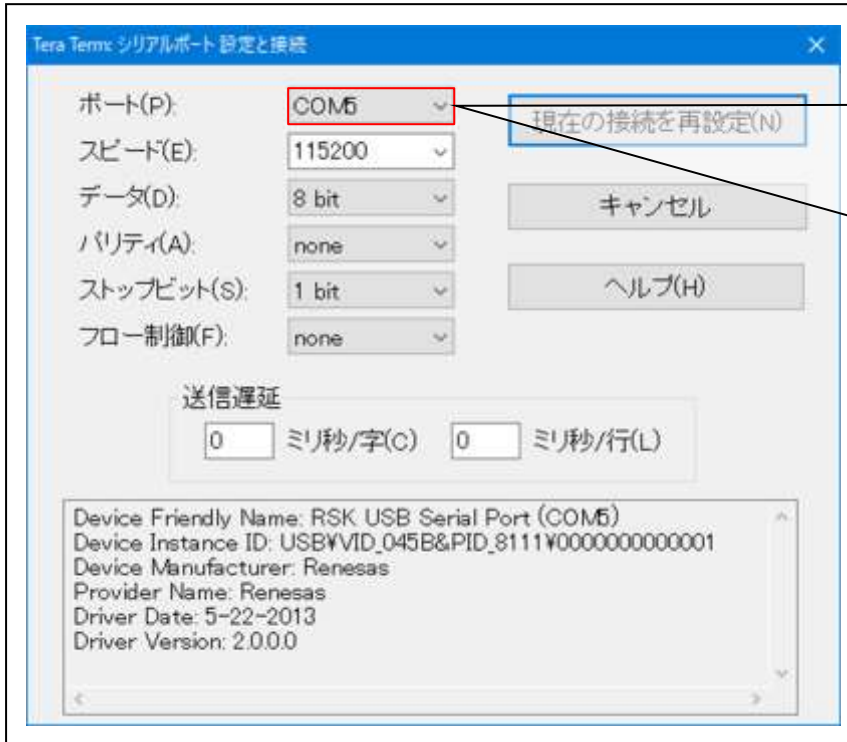
ワークスペース名	概要	プロジェクト名
Azure_sample_ccrx_CK	<p>有線 LAN 接続した DHCP と SMTP クライアントのサンプル</p> <p>Yahoo メールの SMTP サーバーに接続して TLS セッションでクライアント認証後、メール送信をする。</p>	<p>RX65N_ccrx SMTP_TLS_uselib</p> <p>Azure RTOS モードで動作</p> <p>NetX Duo DHCP Client (dhcp_client)</p> <p>NetX Duo DNS Client (nx_dns_.....)</p> <p>NetX Duo SMTP Client (nx_smtp_.....)</p> <p>NetX Duo TLS/DTLS Session (nx_secure_.....)</p>

統合開発環境
Renesas e2studio(Version 2024-04)
Azure RTOS (Version 6.2.1 rel-rx 2.0.0)
Renesas CCRX(Version 2.08.01)
テキストファイル・エンコード(UTF-8)

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。
- ② PC にインストールし実行する
- ③ シリアルポートの設定

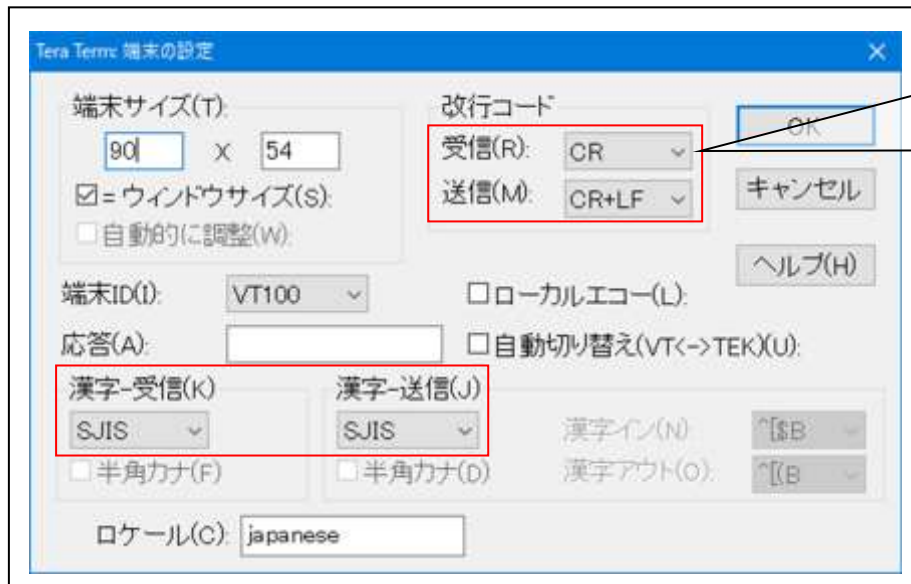


COM 番号は、  
PC 側でシリアル通信可能な  
番号を指定する。

115200BPS  
8bit  
none  
1bit  
none

の仕様にする。

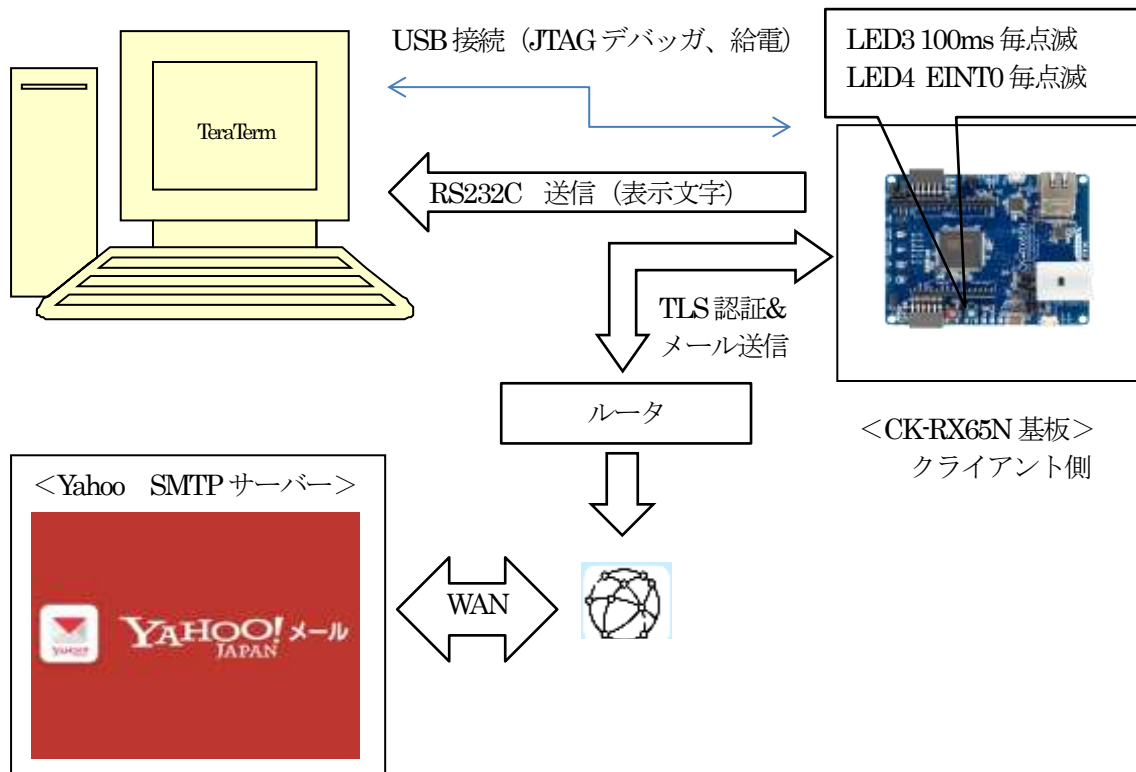
#### ④ 端末の設定



USB シリアルコンバータ  
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。

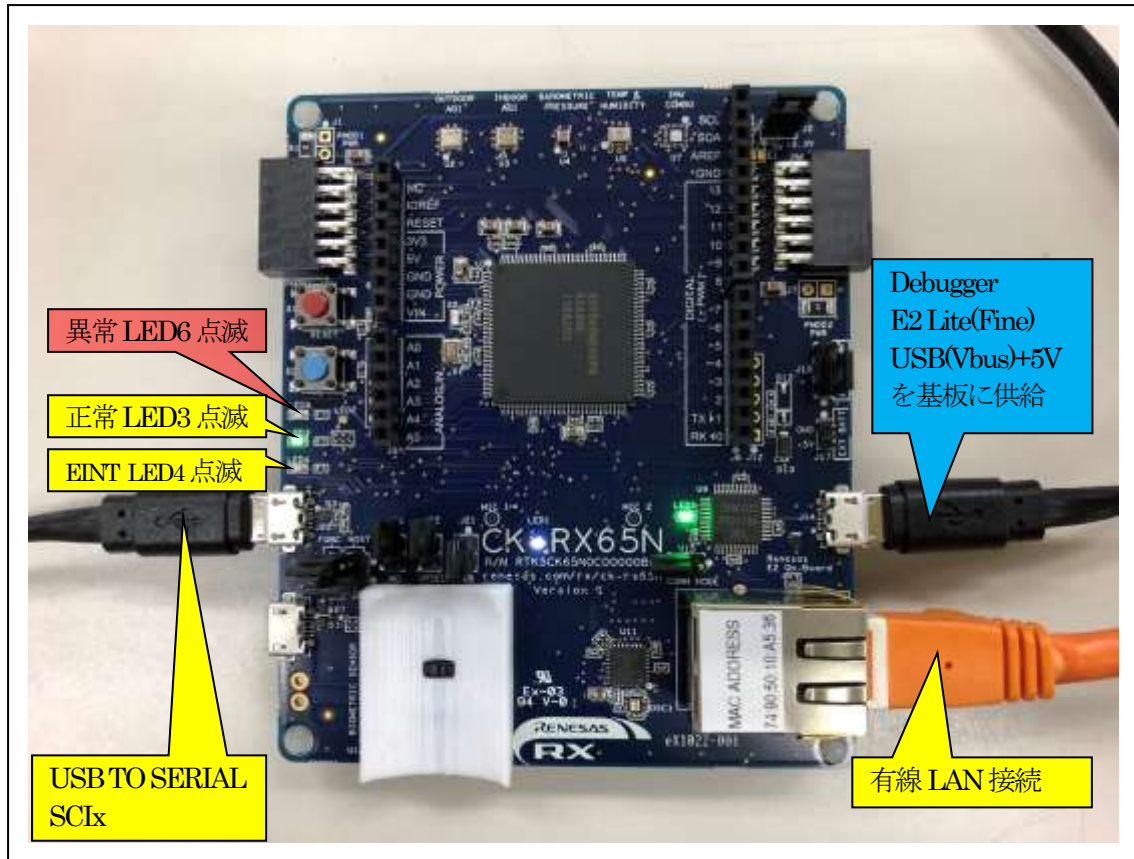
赤枠の設定にする。

4. 動作構成



<RX65N\_ccrx\_SMTP\_TLS\_uselib>

1. OpenSSL を使用して取得した Yahoo メール の SMTP サーバ証明書は登録済み。
2. SSL/TLS 対応の API を実装する。(NetX Duo TLS/DTLS Session)
3. 現バージョンの AzureRTOS NetX Duo SMTP Client は、SSL/TLS に対応していないため SSL/TLS 対応に追加した SMTP クライアントを実装する。
4. 定義済みのメール文を「送信先メールアドレス」に送信する。



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 mode

## 5. 「RX65N\_ccrx\_SMTP\_TLS\_uselib」 サンプルの説明

## 5-1. フォルダ構成とファイル名【&lt;ホルダ名&gt;を示す】

<Azure_sample_ccrx_CK>		
<rx_ccrx_filex_lib>	AzureRTOS FileX ライブラリ作成用ホルダ	
<rx_ccrx_netxduo_addons_lib >	AzureRTOS NetX DuoAddons ライブラリ作成用ホルダ ©SMTP クライアントソースはビルド除外にする。	
<rx_ccrx_netxduo_lib>	AzureRTOS NetX Duo ライブラリ作成用ホルダ	
<rx_ccrx_threadx_lib>	AzureRTOS ThredX ライブラリ作成用ホルダ	
<RX65N_ccrx_SMTP_TLS_uselib>	DHCP / TLS 認証 / メール送信 サンプルプロジェクト	
<HardwareDebug>	RX65N_ccrx_SMTP_TLS_u selib.abs	ABS ファイル、デバッガで使用
	RX65N_ccrx_SMTP_TLS_u selib.map	MAP ファイル、アドレス情報
	RX65N_ccrx_SMTP_TLS_u selib.mot	モトローラーHEX ファイル
	その他	自動生成ファイル
<lib>	<filex>	FileX (全C ソースはビルド除外)
	<netxduo>	NetX Duo (全C ソースはビルド除外)
	<netxduo_addons>	NetX Duo Addons (全C ソースはビルド除外)
	<thredx>	ThreadX (全C ソースはビルド除外)
	librx_ccrx_filex_lib.lib	FileX ライブラリ
	librx_ccrx_netxduo_addons_lib.lib	NetX DuoAddons ライブラリ
	librx_ccrx_netxduo_lib.lib	NetX ライブラリ
	ibrx_ccrx_threadx_lib.lib	ThredX ライブラリ
<src>	<rtos_config>	スマートコンフィグレータにより作成
	<smc_gen>	スマートコンフィグレータにより作成
	<rtos_skeleton>	
	dhcp_fixed_entry.c	DHCP スレッド処理のソース
	SMTP_thread_entry.c	SMTP スレッド処理のソース
	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 I/O デバイス初期化ソース
	hardware_setup.h	hardware_setup.c のヘッダー
	nx_driver_rx_fit.c	Renesas RX FIT driver: Control
	nx_driver_rx_fit.h	nx_driver_rx_fit.c のヘッダー
	nxd_smtp_client.c	TLS 通信を追加した SMTP クライアントソース
	nxd_smtp_client.h	nxd_smtp_client.c のヘッダー
	sample_netx_duo_ping.c	NetX 等初期化サンプルソース
	sample_netx_duo_ping.h	sample_netx_duo_ping.c のヘッダー

	RX65N_ccrx_SMTP_TLS_uselib.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

1) MACアドレスの定義場所

	nx_driver_rx_fit.c
84行	static UCHAR_netx_driver_rx_fit_mac_address[] = {0x74,0x90,0x50,0x10,0xa5,0x36}; //MAC アドレス

2) ライブラリの変更

nx_ccrx_netxduo_addons_lib	¥src¥netxduo_addons¥crypto_libraries¥src¥ nx_crypto_generic_ciphersuites.c
Ciphersuite table の変更	NX_SECURE_TLS_CIPHERSUITE_INFO _nx_crypto_ciphersuite_lookup_table[] = { ..... // 省略 ソースを参照 };

3) DNSサーバとゲートウェイのIPアドレス変更

	¥src¥rtos_skeleton¥SMTP_thread_entry.c
310行	#define DNS_SERVER_ADDRESS IP_ADDRESS(192,168,xx,xxx)
311行	#define GATEWAY_ADDRESS IP_ADDRESS(192,168,xx,xxx)

## 5-2. Macro Defines の説明

Macro Name	値	説明
NX_DNS_CACHE_ENABLE		DNS キャッシュを有効にする。
SecureTLS	0	証明書の検証が不要な SMTP サーバーの場合
	1	証明書の検証が必要な SMTP サーバーの場合
NX_ENABLE_DHCP	0	DHCP Client Disable ◎ソースコードに直接 IP アドレスを記述 sample_netx_duo_ping.c : <pre> status = nx_ip_create( &amp;ip_0, "NetX IP Instance 0", #ifdef NX_ENABLE_DHCP == 1) IP_ADDRESS(0,0,0,0), IP_ADDRESS(255,255,255,0), #else IP_ADDRESS(192,168,21,19), // 固定 IP アドレス IP_ADDRESS(255,255,255,0), // サブネットマスク #endif &amp;pool_0, nx_driver_rx_fit, (UCHAR*)ip_thread_stack, sizeof(ip_thread_stack), 1);           </pre>
	1	DHCP Client Enable
TX_INCLUDE_USER_DEFINE_FILE		「tx_user.h」を有効にする
NX_INCLUDE_USER_DEFINE_FILE		「nx_user.h」を有効にする
FX_INCLUDE_USER_DEFINE_FILE		「fx_user.h」を有効にする
NXD_MQTT_CLOUD_ENABLE		MQTT メッセージングプロトコルを有効にする
NX_SECURE_ENABLE		MQTT クライアントは TLS サポート付きで構築される
NX_ENABLE_EXTENDED_NOTIFY_SUPPORT		多くのコールバックフックを有効にする
NX_ENABLE_IP_PACKET_FILTER		IP パケットを有効にする
FLATCC_NO_ASSERT		FLATCC をアサートしない
NX_AZURE_IOT_LOG_LEVEL	0	NX_AZURE ログ関数を使用しない
	1	LogError(...)を使用する
	2	LogError(...)/LogInfo(...)を使用する
	3	LogError(...)/LogInfo(...)/LogDebug(...)を使用する

## 5-3. サンプルの動作説明 (基板側 CK-RX65N)

## 1) DHCP 無効時 (NX\_ENABLE\_DHCP=0)

&lt;DHCP FIXED Thread&gt;

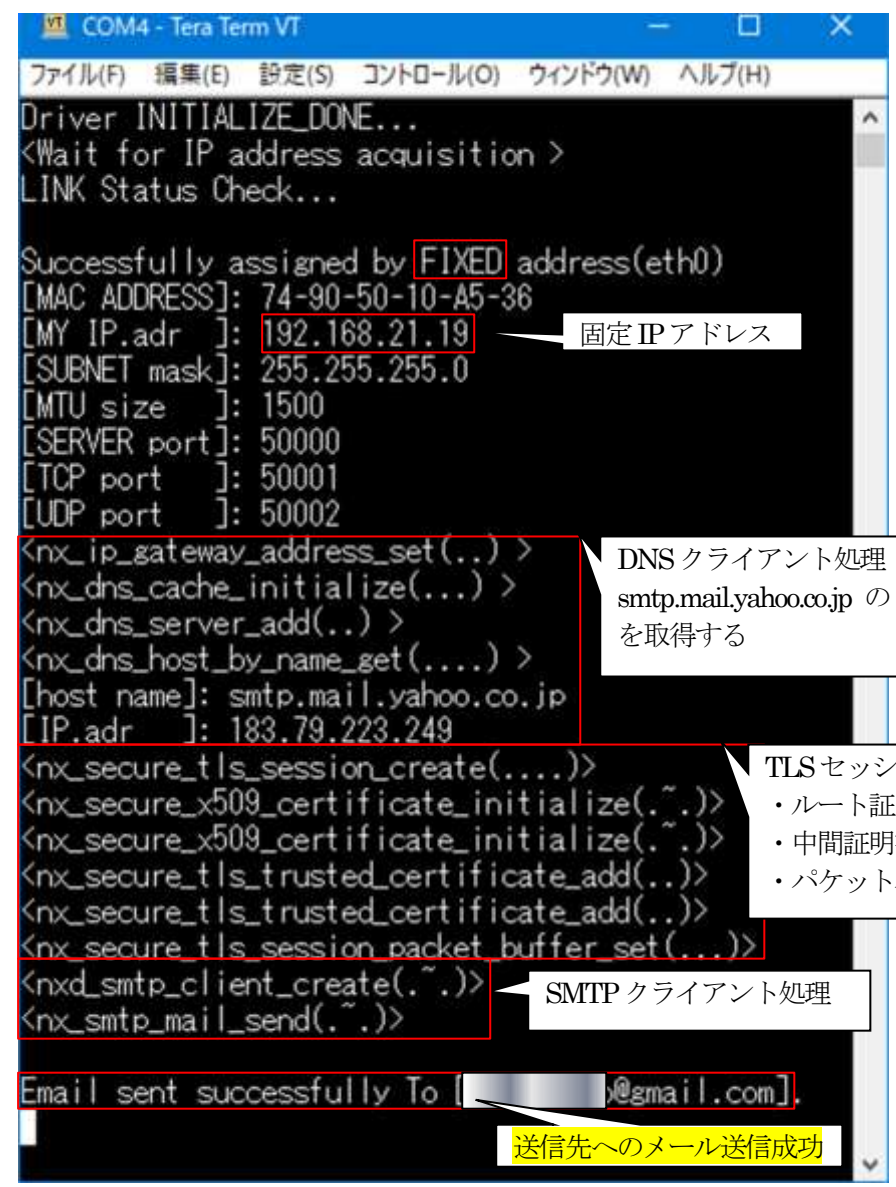
Term 画面

- < 1 > ["Driver INITIALIZE\_DONE.."]
- < 2 > ["<Wait for IP address acquisition>.."]
- < 3 > ["LINK Status Check.."]

&lt;成功画面&gt;

IP アドレス確立により、

基板上的 LED3 (緑色) を 100msec 毎に点滅・LED4 (青色) EINT0 割り込み毎に点滅



COM4 - 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) ヘルプ(H)

```

Driver INITIALIZE_DONE..
<Wait for IP address acquisition >
LINK Status Check...

Successfully assigned by FIXED address(eth0)
[MAC ADDRESS]: 74-90-50-10-A5-36
[MY IP.adr ]: 192.168.21.19
[SUBNET mask]: 255.255.255.0
[MTU size ]: 1500
[SERVER port]: 50000
[TCP port ]: 50001
[UDP port ]: 50002

<nxd_ip_gateway_address_set(..) >
<nxd_dns_cache_initialize(..) >
<nxd_dns_server_add(..) >
<nxd_dns_host_by_name_get(..) >
[host name]: smtp.mail.yahoo.co.jp
[IP.adr ]: 183.79.223.249

<nxd_secure_tls_session_create(..) >
<nxd_secure_x509_certificate_initialize(..) >
<nxd_secure_x509_certificate_initialize(..) >
<nxd_secure_tls_trusted_certificate_add(..) >
<nxd_secure_tls_trusted_certificate_add(..) >
<nxd_secure_tls_session_packet_buffer_set(..) >

<nxd_smtp_client_create(..) >
<nxd_smtp_mail_send(..) >

Email sent successfully To [redacted@gmail.com].
  
```

固定 IP アドレス

DNS クライアント処理  
smtp.mail.yahoo.co.jp の IP アドレス  
を取得する

TLS セッションの初期化  
・ルート証明書の追加  
・中間証明書の追加  
・パケットバッファの登録

SMTP クライアント処理

送信先へのメール送信成功



## 2) DHCP 有効時 (NX\_ENABLE\_DHCP=1)

&lt;DHCP FIXED Thread&gt;

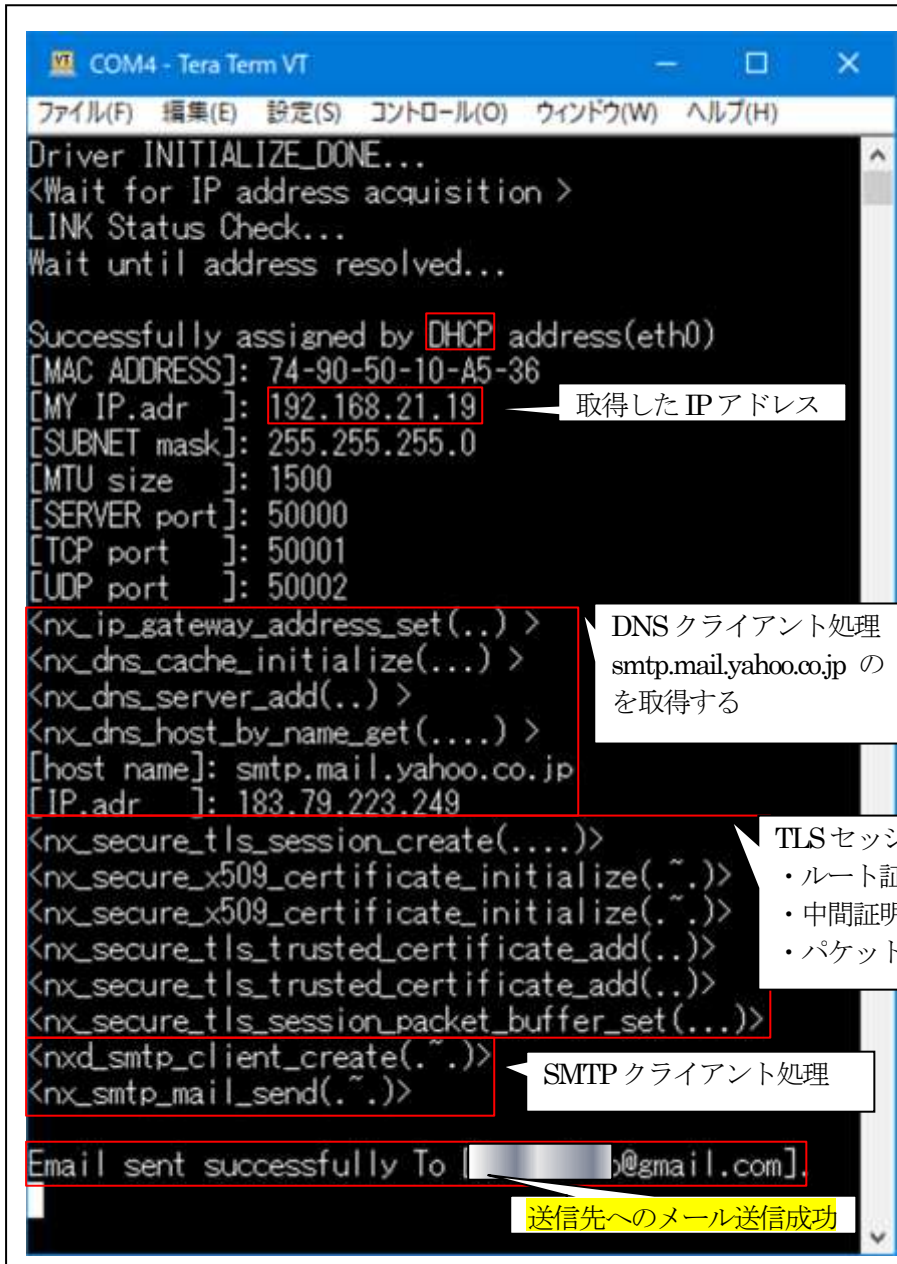
Term 画面

- < 1 > ["Driver INITIALIZE\_DONE..."]
- < 2 > ["<Wait for IP address acquisition>.."]
- < 3 > ["LINK Status Check..."]
- < 4 > ["DHCP In Progress..."]
- < 5 > [" Wait until address resolved..."]

&lt;成功画面&gt;

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) EINT0 割り込み毎に点滅



COM4 - 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) ヘルプ(H)

```

Driver INITIALIZE_DONE...
<Wait for IP address acquisition >
LINK Status Check...
Wait until address resolved...

Successfully assigned by DHCP address(eth0)
[MAC ADDRESS]: 74-90-50-10-A5-36
[MY IP.adr ]: 192.168.21.19
[SUBNET mask]: 255.255.255.0
[MTU size ]: 1500
[SERVER port]: 50000
[TCP port ]: 50001
[UDP port ]: 50002

<nix_ip_gateway_address_set(..)>
<nix_dns_cache_initialize(..)>
<nix_dns_server_add(..)>
<nix_dns_host_by_name_get(..)>
[host name]: smtp.mail.yahoo.co.jp
[IP.adr ]: 183.79.223.249

<nix_secure_tls_session_create(..)>
<nix_secure_x509_certificate_initialize(..)>
<nix_secure_x509_certificate_initialize(..)>
<nix_secure_tls_trusted_certificate_add(..)>
<nix_secure_tls_trusted_certificate_add(..)>
<nix_secure_tls_session_packet_buffer_set(..)>

<nixd_smtp_client_create(..)>
<nix_smtp_mail_send(..)>

Email sent successfully To [redacted]@gmail.com]
  
```

取得した IP アドレス

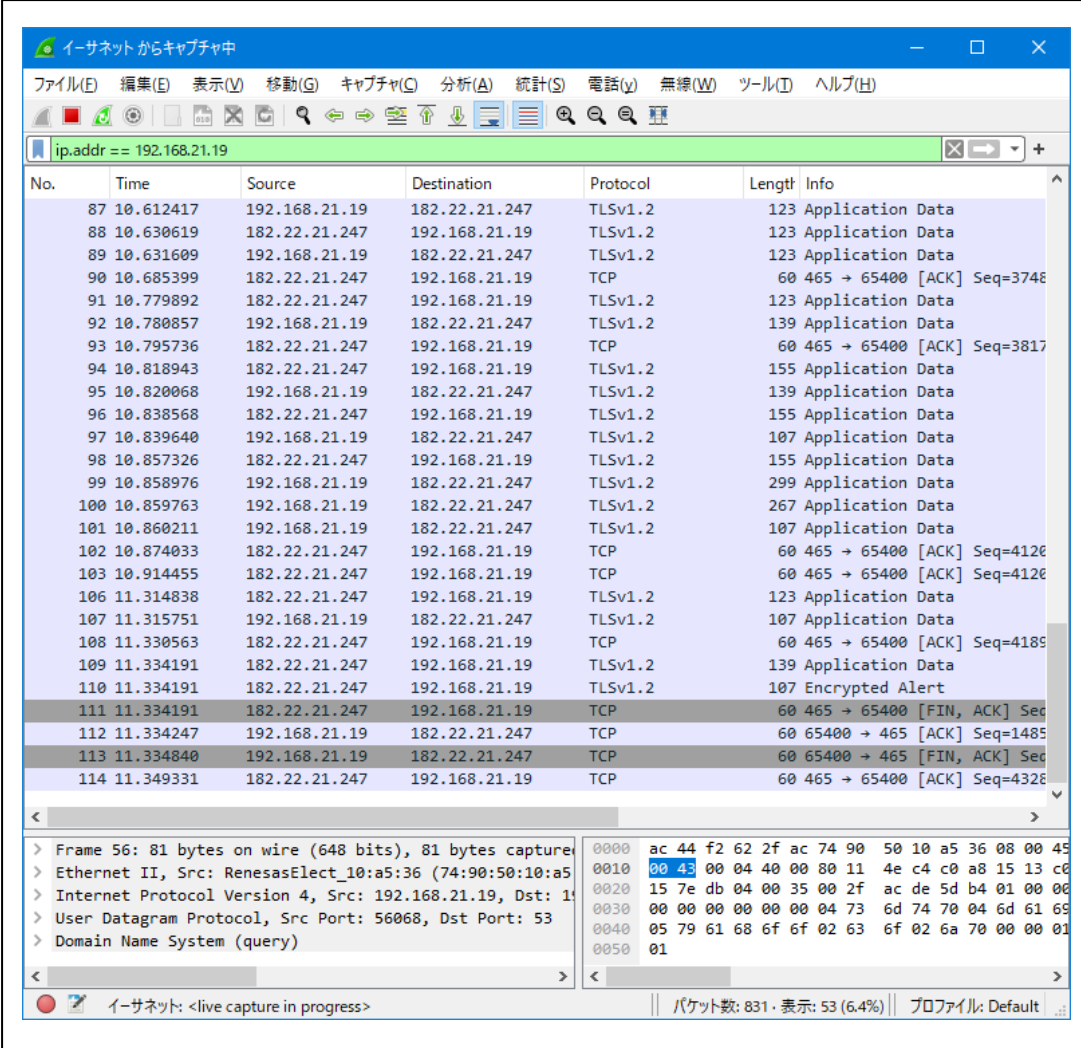
DNS クライアント処理  
smtp.mail.yahoo.co.jp の IP アドレス  
を取得する

TLS セッションの初期化  
・ルート証明書の追加  
・中間証明書の追加  
・パケットバッファの登録

SMTP クライアント処理

送信先へのメール送信成功

## 3) TLS 接続とメール送信が成功した時の WireShark のキャプチャ画面



The screenshot shows a Wireshark capture window titled "イーサネット からキャプチャ中". The filter is set to "ip.addr == 192.168.21.19". The packet list shows a series of TLS and TCP packets between 192.168.21.19 and 182.22.21.247. Packet 56 is highlighted, and its details pane is expanded to show the following structure:

- Frame 56: 81 bytes on wire (648 bits), 81 bytes captured (648 bits) on interface 0
- Ethernet II, Src: RenesasElect\_10:a5:36 (74:90:50:10:a5:36), Dst: 182.22.21.247
- Internet Protocol Version 4, Src: 192.168.21.19, Dst: 182.22.21.247
- User Datagram Protocol, Src Port: 56068, Dst Port: 53
- Domain Name System (query)

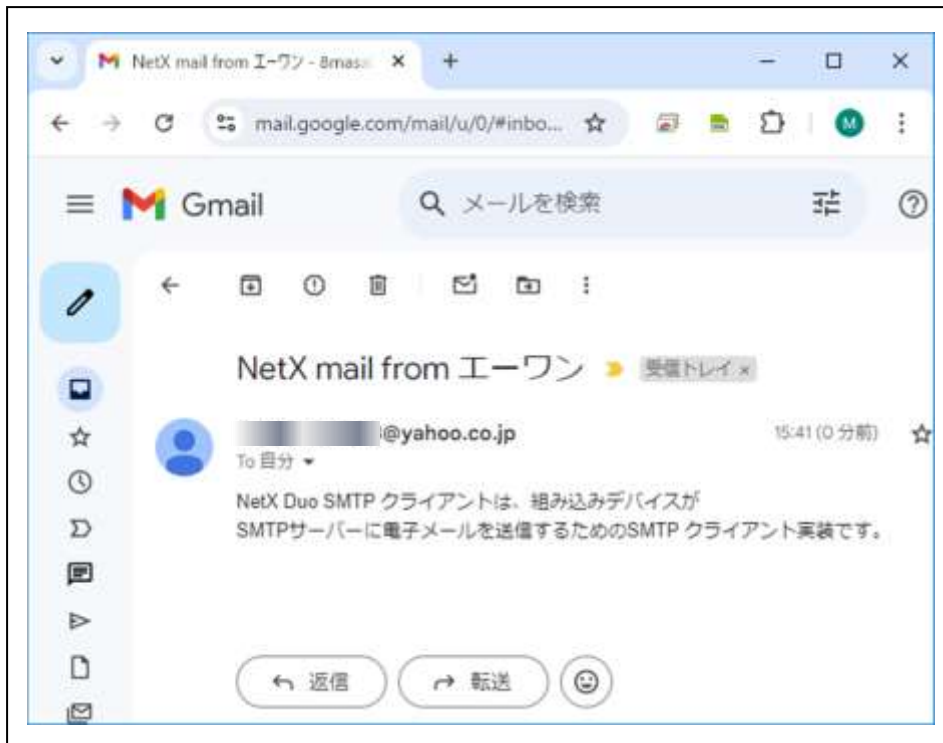
The hex dump for packet 56 is as follows:

```

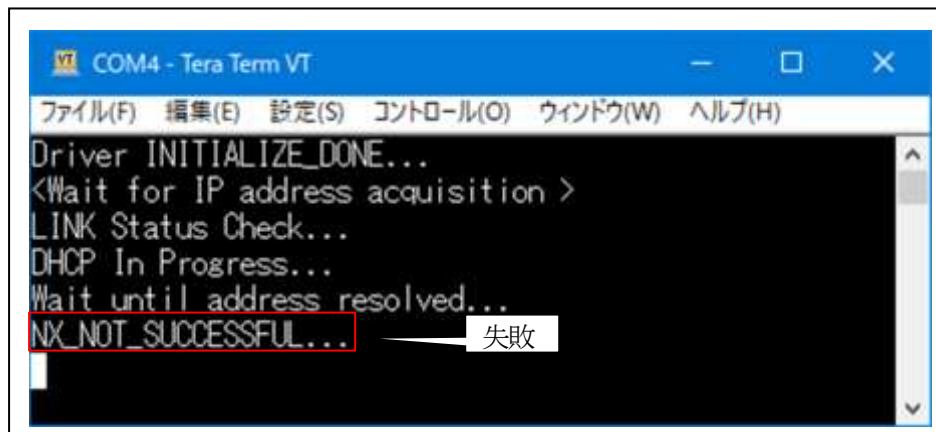
0000  ac 44 f2 62 2f ac 74 90 50 10 a5 36 08 00 45
0010  00 43 00 04 40 00 80 11 4e c4 c0 a8 15 13 c6
0020  15 7e db 04 00 35 00 2f ac de 5d b4 01 00 00
0030  00 00 00 00 00 00 04 73 6d 74 70 04 6d 61 69
0040  05 79 61 68 6f 6f 02 63 6f 02 6a 70 00 00 01
0050  01
  
```

The status bar at the bottom indicates: "イーサネット: <live capture in progress> | パケット数: 831 · 表示: 53 (6.4%) | プロファイル: Default"

4) メール送信成功時のメイラー画面



5) <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



## 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を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
- ・「AzureRTOS」 マイクロソフト株式会社
- ・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

〒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.jp>

