

## Renesas R5F565N9(RX65N-1MB)用サンプル

## (e2studio RX65N\_gnu\_ping\_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

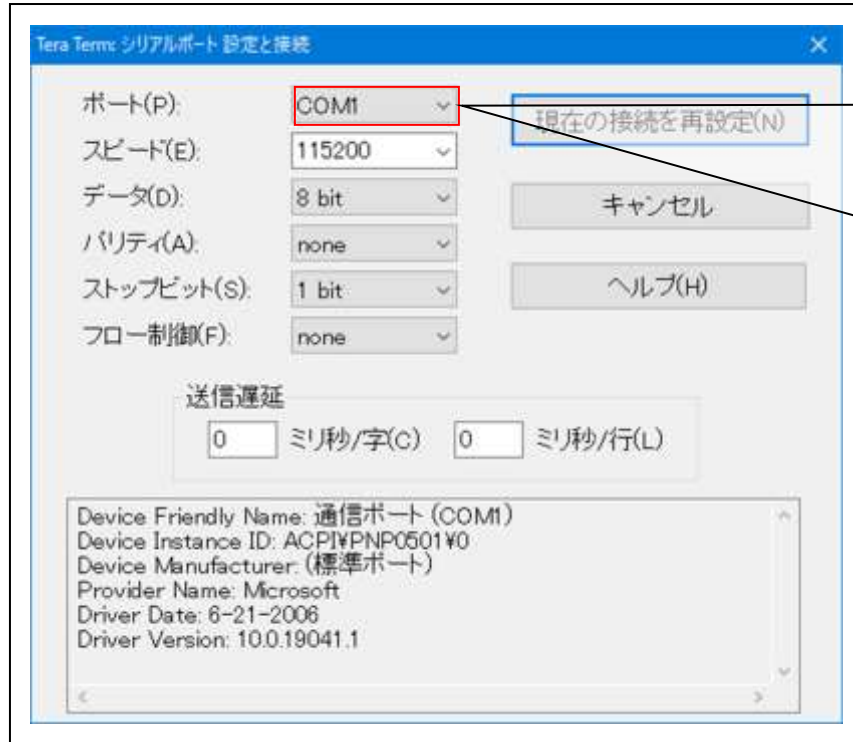
ワークスペース名	概要	プロジェクト名
Azure_sample_gnu	有線 LAN 接続した DHCP と ping 確認のサンプル	RX65N_gnu_ping_uselib  Azure RTOS モードで動作  NetX DHCP Client (dhcp_client)

統合開発環境
Renesas e2studio (Version 2024-04)
Azure RTOS (Version 6.2.1 rel-rx 2.0.0)
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
AP-RX65N-0A (アルファプロジェクト製)

### 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



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。

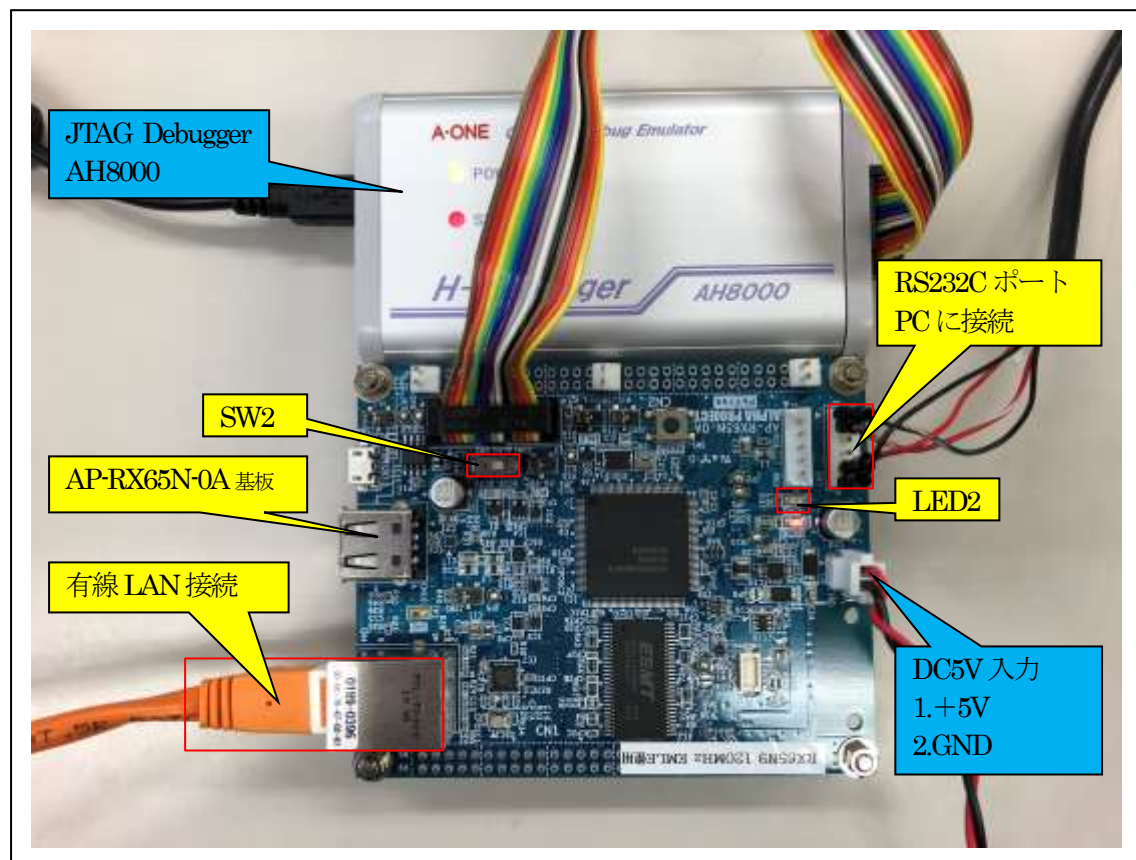
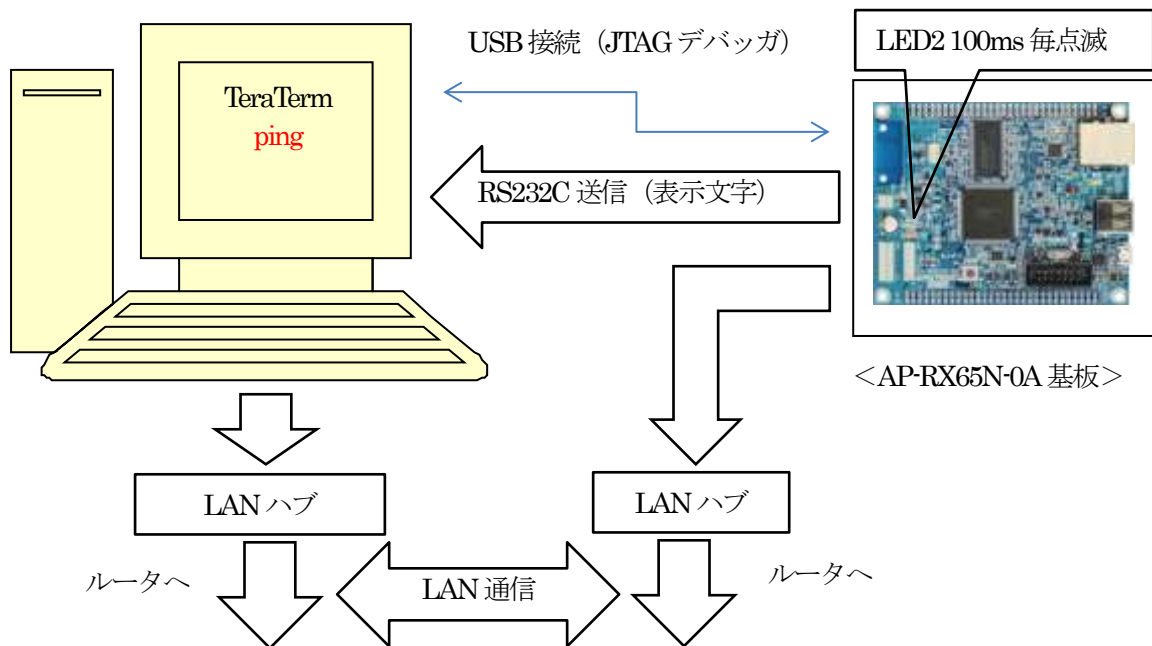
#### ④端末の設定



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

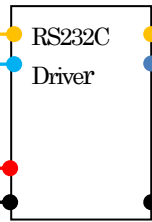


- ①Windows-PC と接続する RS232C ケーブルは、製作が必要です。
- ②「RS232C-Driver」は、下記 URL の「RS232CAB4」を推奨します。  
[http://tool-kobo.ddo.jp/Files/Product/RS232\\_422/RS232CAB.htm](http://tool-kobo.ddo.jp/Files/Product/RS232_422/RS232CAB.htm)

AP-RX65N-0A(CN5)

6pin メス	
1	RXD6
2	TXD6
3	
4	
5	VCC
6	GND

SW2-SCI6



PC-Window's

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

## 5. 「RX65N\_gnu\_ping\_uselib」 サンプルの説明

### 5-1. フォルダ構成とファイル名【<ホルダ名>を示す】

<Azure_sample_gnu>			
	<rx_gnu_filex_lib>	AzureRTOS FileX ライブラリ作成用ホルダ	
	<rx_gnu_netxduo_addons_lib>	AzureRTOS NetX DuoAddons ライブラリ作成用ホルダ	
	<rx_gnu_netxduo_lib>	AzureRTOS NetX Duo ライブラリ作成用ホルダ	
	<rx_gnu_threadx_lib>	AzureRTOS ThredX ライブラリ作成用ホルダ	
	<RX65N_gnu_ping_uselib>	DHCP/ping サンプルプロジェクト	
	<HardwareDebug>	RX65N_gnu_ping_uselib.elf	ELF ファイル、デバッガで使用
		RX65N_gnu_ping_uselib.map	MAP ファイル、アドレス情報
		RX65N_gnu_ping_uselib.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)
		<threadx>	ThreadX (全C ソースはビルド除外)
		librx_gnu_filex_lib.a	FileX ライブラリ
		librx_gnu_netxduo_addons_lib.a	NetX DuoAddons ライブラリ
		librx_gnu_netxduo_lib.a	NetX ライブラリ
		librx_gnu_threadx_lib.a	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
		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
		sample_netx_duo_ping.c	NetX 等初期化サンプルソース
		RX65N_gnu_ping_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

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

## 5 - 2. Macro Defines の説明

Macro Name	値	説明
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 : status = nx_ip_create( &ip_0, "NetX IP Instance 0", #if(NX_ENABLE_DHCP == 1) IP_ADDRESS(0,0,0,0), IP_ADDRESS(255,255,255,0), #else IP_ADDRESS(192,168,21,95), // 固定 IP アドレス IP_ADDRESS(255,255,255,0), // サブネットマスク #endif &pool_0, nx_driver_rx_fit, (UCHAR*)ip_thread_stack, sizeof(ip_thread_stack), 1);
	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 AP-RX65N-0A）

#### 1) DHCP 無効時（NX\_ENABLE\_DHCP=0）

<DHCP FIXED Thread>

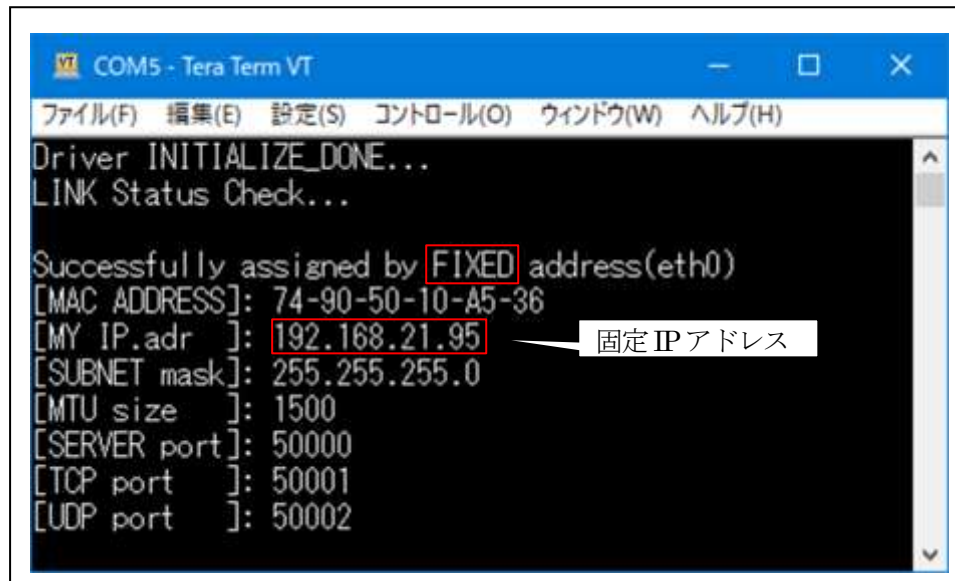
Term 画面

< 1 > 「"Driver INITIALIZE\_DONE..."」

< 2 > 「"LINK Status Check..."」

<成功画面>

IP アドレス確立により、基板上の LED2 を 100msec 毎に点滅



The screenshot shows a terminal window titled "COM5 - Tera Term VT". The menu bar includes "ファイル(F)", "編集(E)", "設定(S)", "コントロール(O)", "ウインドウ(W)", and "ヘルプ(H)". The output text is as follows:

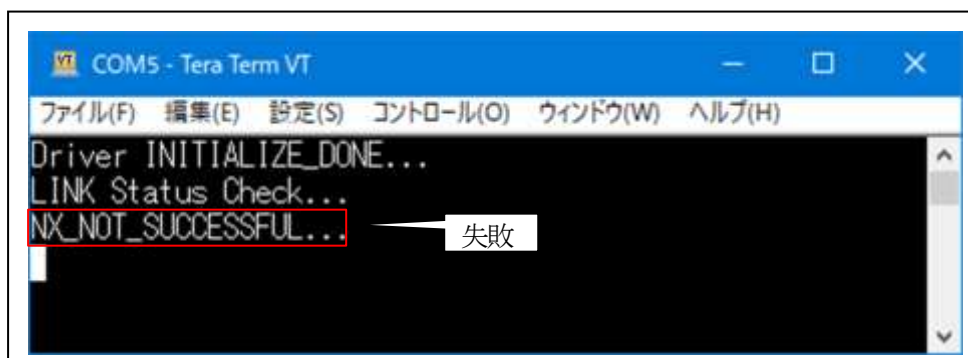
```
Driver INITIALIZE_DONE...
LINK Status Check...

Successfully assigned by FIXED address(eth0)
[MAC ADDRESS]: 74-90-50-10-A5-36
[MY IP.adr ]: 192.168.21.95
[SUBNET mask]: 255.255.255.0
[MTU size ]: 1500
[SERVER port]: 50000
[TCP port ]: 50001
[UDP port ]: 50002
```

In the original image, the word "FIXED" and the IP address "192.168.21.95" are highlighted with red boxes. A callout box with the text "固定 IP アドレス" (Fixed IP Address) points to the IP address.

<失敗画面>

IP アドレス未確立により、基板上の LED2 を 500msec 毎に点滅



The screenshot shows a terminal window titled "COM5 - Tera Term VT" with the same menu bar as the previous image. The output text is as follows:

```
Driver INITIALIZE_DONE...
LINK Status Check...
NX_NOT_SUCCESSFUL...
```

In the original image, the text "NX\_NOT\_SUCCESSFUL..." is highlighted with a red box. A callout box with the text "失敗" (Failure) points to this text.



## 2) DHCP 有効時 (NX\_ENABLE\_DHCP = 1)

<DHCP FIXED Thread>

Term 画面

< 1 > 「Driver INITIALIZE\_DONE...」

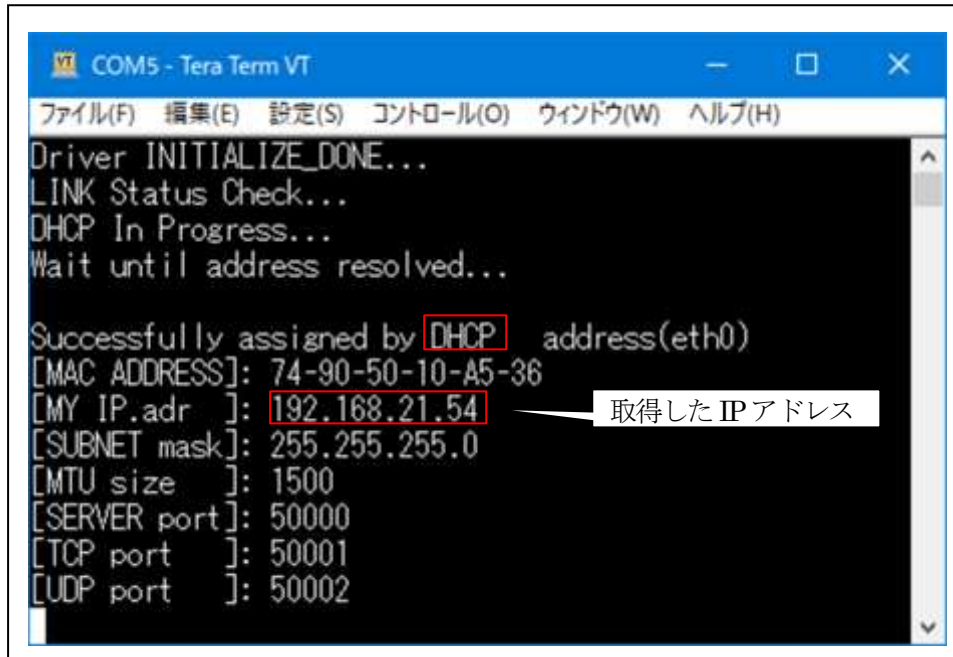
< 2 > 「LINK Status Check...」

< 3 > 「DHCP In Progress...」

< 4 > 「Wait until address resolved...」

<成功画面>

IP アドレス取得により、基板上の LED2 を 100msec 毎に点滅



```

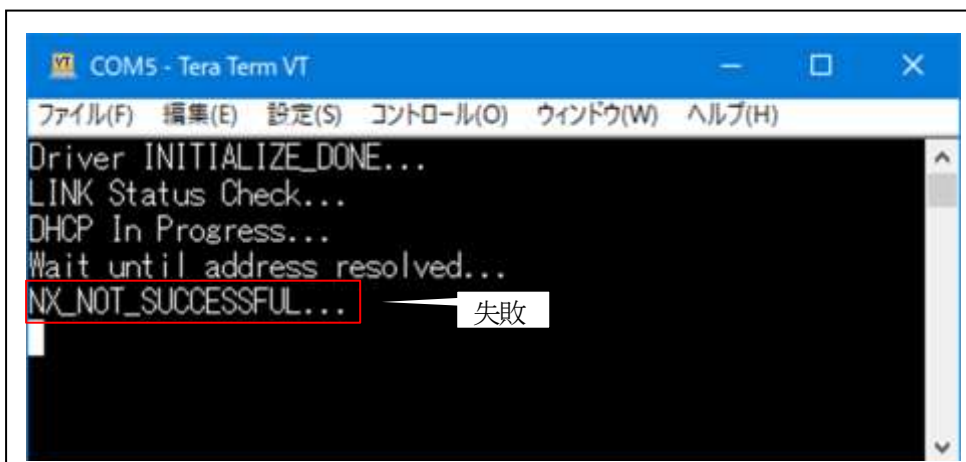
COM5 - Tera Term VT
ファイル(F) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) ヘルプ(H)
Driver INITIALIZE_DONE...
LINK Status Check...
DHCP In Progress...
Wait until address resolved...

Successfully assigned by DHCP address(eth0)
[MAC ADDRESS]: 74-90-50-10-A5-36
[MY IP.adr ]: 192.168.21.54
[SUBNET mask]: 255.255.255.0
[MTU size ]: 1500
[SERVER port]: 50000
[TCP port ]: 50001
[UDP port ]: 50002
  
```

取得した IP アドレス

<失敗画面>

IP アドレス未取得により、基板上の LED2 を 500msec 毎に点滅

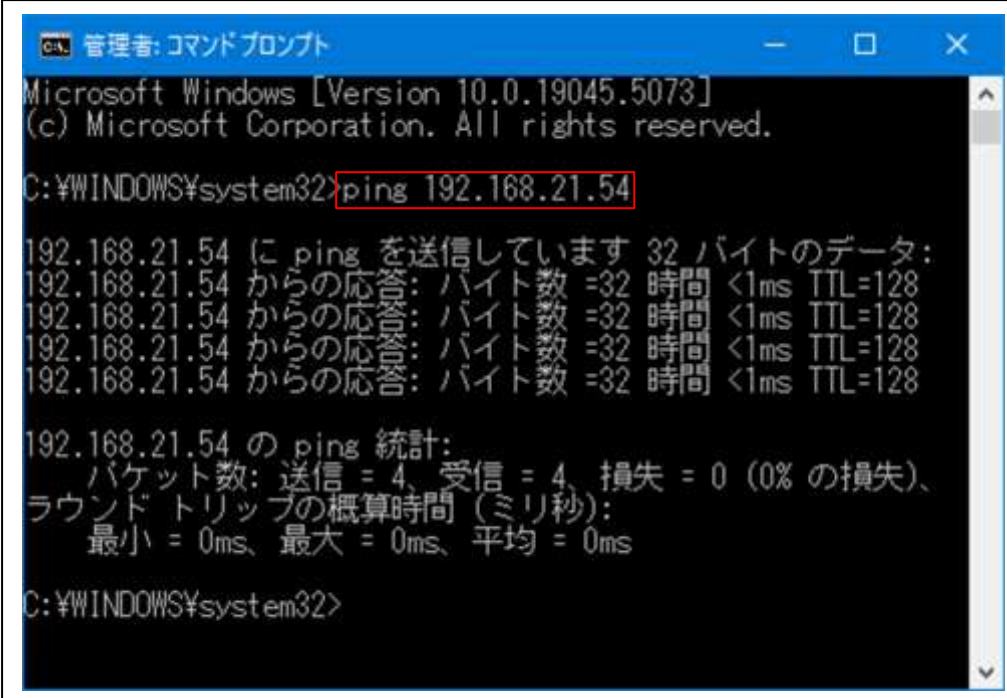


```

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) ヘルプ(H)
Driver INITIALIZE_DONE...
LINK Status Check...
DHCP In Progress...
Wait until address resolved...
NX_NOT_SUCCESSFUL...
  
```

失敗

3) Windows コマンドプロンプト ping で確認



```
管理: コマンドプロンプト
Microsoft Windows [Version 10.0.19045.5073]
(c) Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>ping 192.168.21.54

192.168.21.54 に ping を送信しています 32 バイトのデータ:
192.168.21.54 からの応答: バイト数 =32 時間 <1ms TTL=128
192.168.21.54 からの応答: バイト数 =32 時間 <1ms TTL=128
192.168.21.54 からの応答: バイト数 =32 時間 <1ms TTL=128
192.168.21.54 からの応答: バイト数 =32 時間 <1ms TTL=128

192.168.21.54 の ping 統計:
    パケット数: 送信 = 4、受信 = 4、損失 = 0 (0% の損失)、
    ラウンドトリップの概算時間 (ミリ秒):
        最小 = 0ms、最大 = 0ms、平均 = 0ms

C:\WINDOWS\system32>
```

## 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。
- ・本文章に関して、アルファプロジェクト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。
- ・AP-RX65N-0A は、株式会社アルファプロジェクトの商品名です。
- ・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.in>

