

## Renesas R5F565NE(CK-RX65N)用サンプル

### (e2studio RX65N\_gnu\_dhcp\_gt202\_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

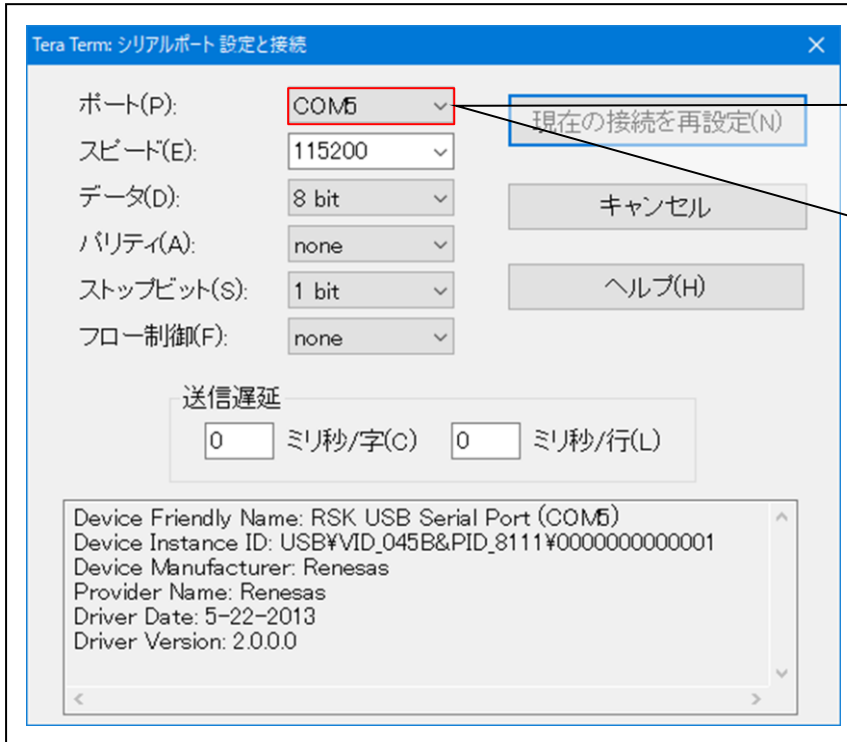
ワークスペース名	概要	プロジェクト名
Azure_sample_gnu_wifi_CK	無線 WiFi-module(GT202-QCA4002)を使用した DHCP と ping 確認のサンプル	RX65N_gnu_dhcp_gt202_uselib  Azure RTOS モードで動作  NetX DHCP Client (dhcp_client)

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 8.3.0.202305)

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

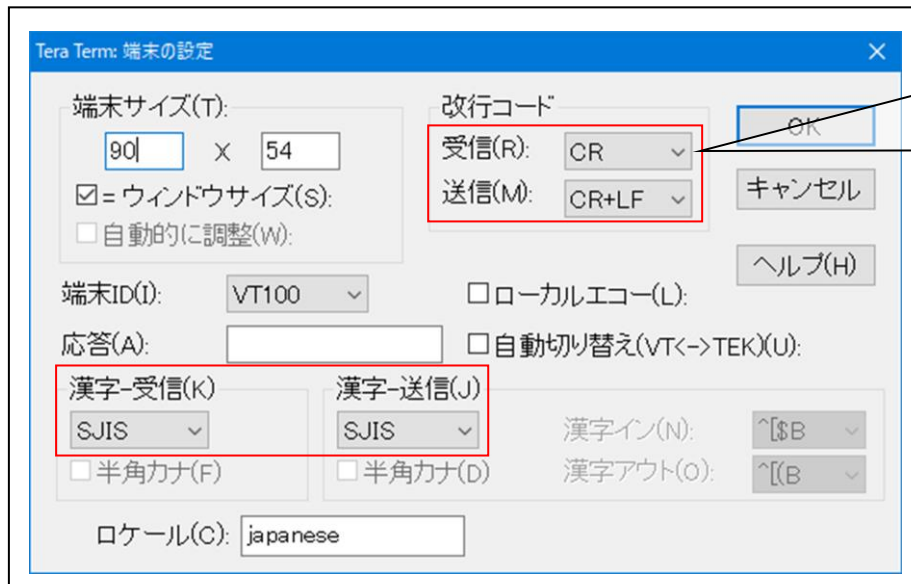


COM 番号は、  
PC 側でシリアル通信可能な  
番号を指定する。

115200BPS  
8bit  
none  
1bit  
none

の仕様にする。

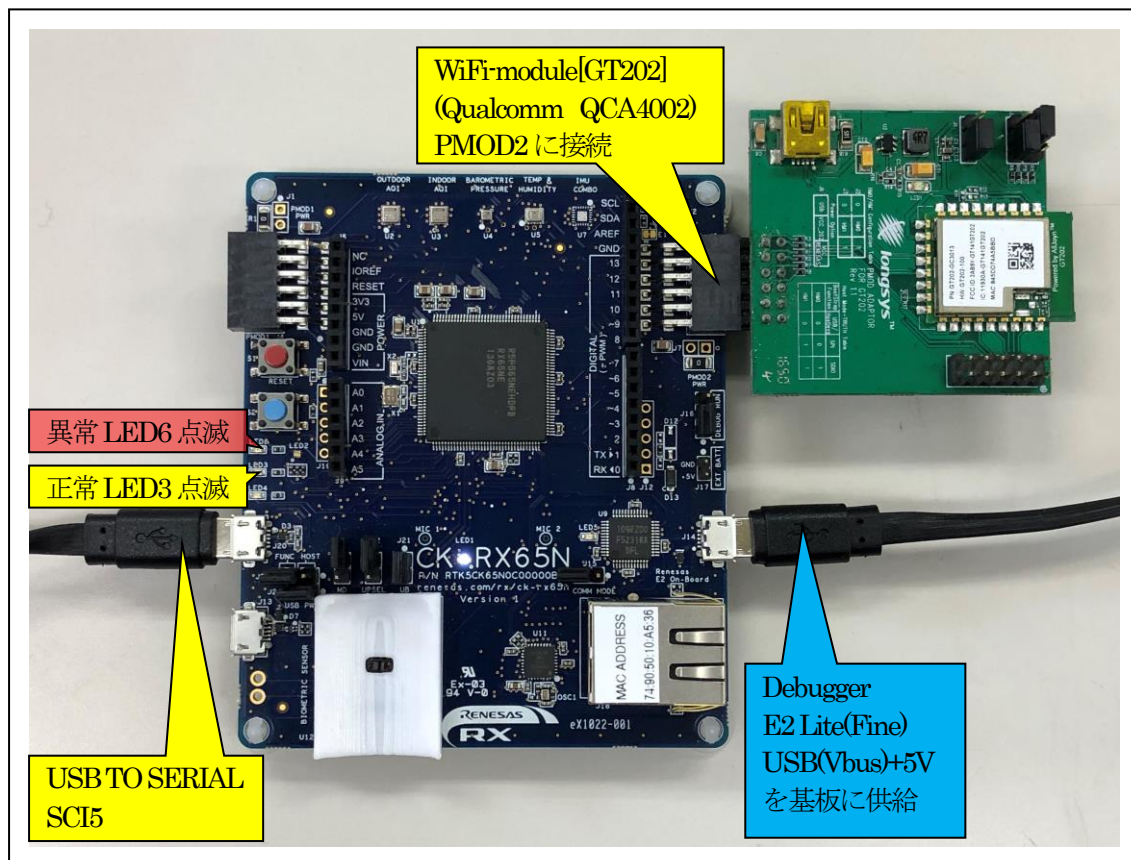
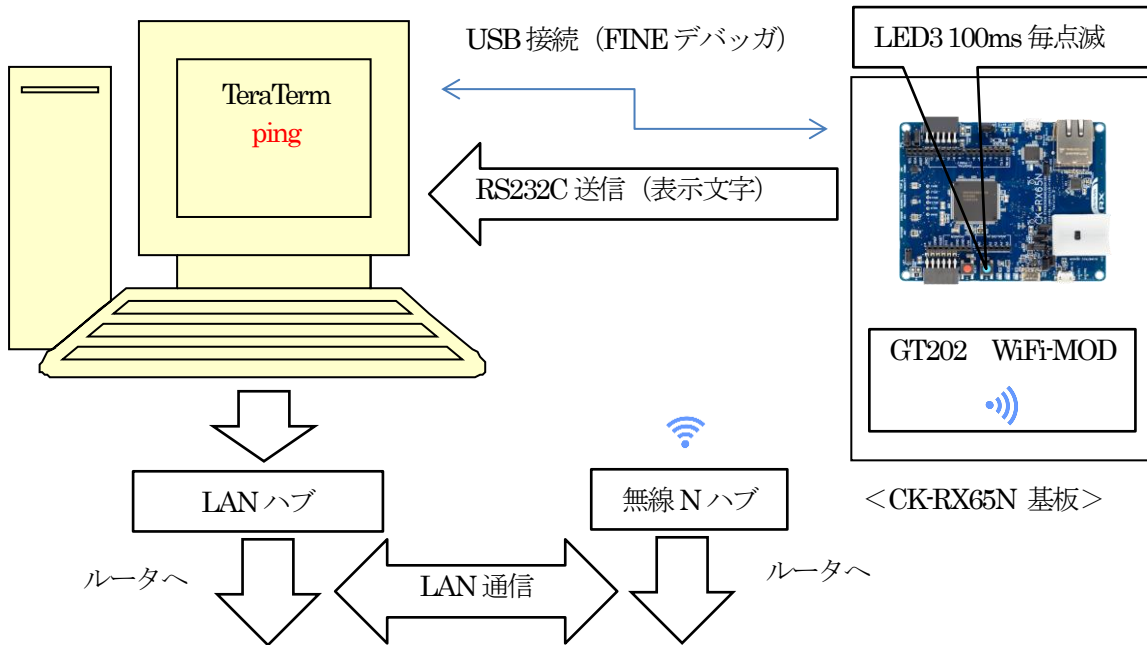
#### ④ 端末の設定



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



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\_gnu\_dhcp\_gt202\_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_gnu_wifi_CK>		
<rx_gnu_filex_lib>	AzureRTOS FileX ライブラリ作成用ホルダ	
<rx_gnu_netxduo_addons_lib>	AzureRTOS NetX Duo Addons 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_dhcp_gt202_uselib>		
<HardwareDebug>	RX65N_gnu_dhcp_gt202_uselib.elf	ELF ファイル、JTAG で使用
	RX65N_gnu_dhcp_gt202_uselib.map	MAP ファイル、アドレス情報
	RX65N_gnu_dhcp_gt202_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 Duo Addons ライブラリ
	librx_gnu_netxduo_lib.a	NetX ライブラリ
	ibrx_gnu_threadx_lib.a	ThredX ライブラリ
<src>	<driver>	
	<r_irq_rx>	FSP IRQ ドライバ (変更)
	<r_sci_rx>	FSP SPI ドライバ (変更)
	<rtos_config>	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
	<smc_gen>	スマートコンフィグレータにより作成
	<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 I/O デバイス初期化ソース
	hardware_setup.h	hardware_setup.c のヘッダー
	sample_netx_duo_ping.c	NetX 等初期化サンプルソース
	sf_wifi_nsal_api.c	NetX WiFi-API ソース

---

---

		sf_wifi_nsal_api.h	sf_wifi_nsal_api.c のヘッダ
		linker_script.ld	リンカスクリプトファイル
	RX65N_gnu_dhcp_gt202_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	

## 5-2. Macro Defines の説明

Macro Name	値	説明
A_PRINTF_ENABLED	0	処理進行内容の表示を無効にする
	1	処理進行内容の表示を有効にする
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;g_ip0, "NetX IP Instance 0", #if (NX_ENABLE_DHCP == 1) IP_ADDRESS(0,0,0,0), IP_ADDRESS(255,255,255,0), #else IP_ADDRESS(192,168,21,54), //固定 IP アドレス IP_ADDRESS(255,255,255,0), //サブネットマスク #endif &amp;g_pool0, nsal_netx_driver, (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;

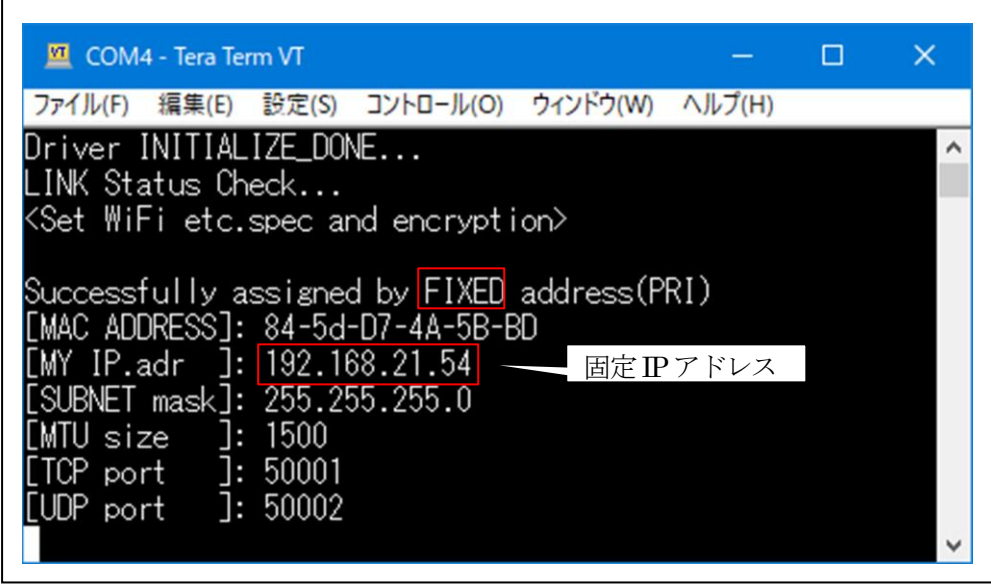
Term 画面

&lt; 1 &gt; 「"Driver INITIALIZE\_DONE..."」

&lt; 2 &gt; 「"LINK Status Check..."」

&lt; 3 &gt; 「"&lt;Set Wi Fi ect.spec and emcryption&gt;"」

&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



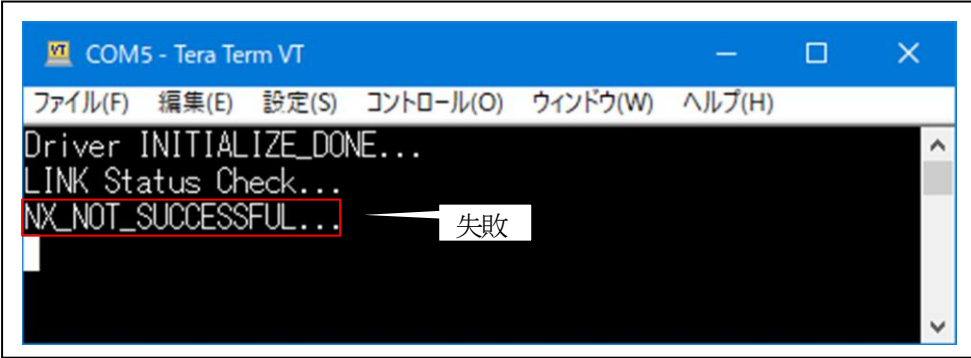
```

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...
LINK Status Check...
<Set Wi Fi ect.spec and emcryption>

Successfully assigned by FIXED address(PRI)
[MAC ADDRESS]: 84-5d-D7-4A-5B-BD
[MY IP.adr ]: 192.168.21.54
[SUBNET mask]: 255.255.255.0
[MTU size ]: 1500
[TCP port ]: 50001
[UDP port ]: 50002
  
```

固定 IP アドレス

&lt;失敗画面&gt;IP アドレス未確立により、基板上的 LED6（赤色）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...
NX_NOT_SUCCESSFUL...
  
```

失敗



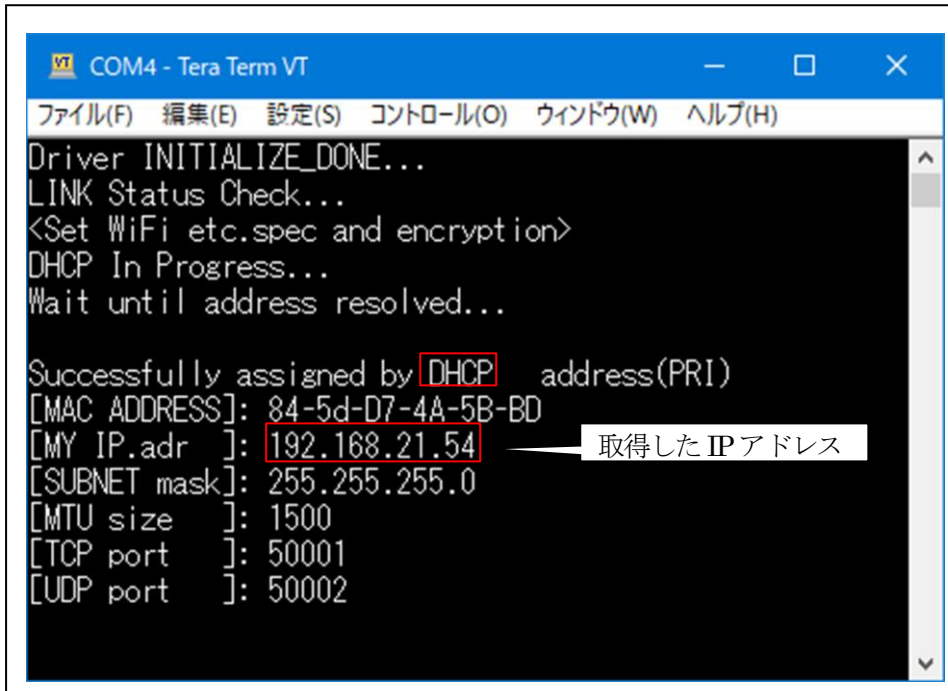
## 2) DHCP 有効時 (NX\_ENABLE\_DHCP=1)

&lt;DHCP FIXED Thread&gt;

Term 画面

- < 1 > ["Driver INITIALIZE\_DONE..."]
- < 2 > ["LINK Status Check..."]
- < 3 > ["<Set Wi Fi ect.spec and emryption>"]
- < 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

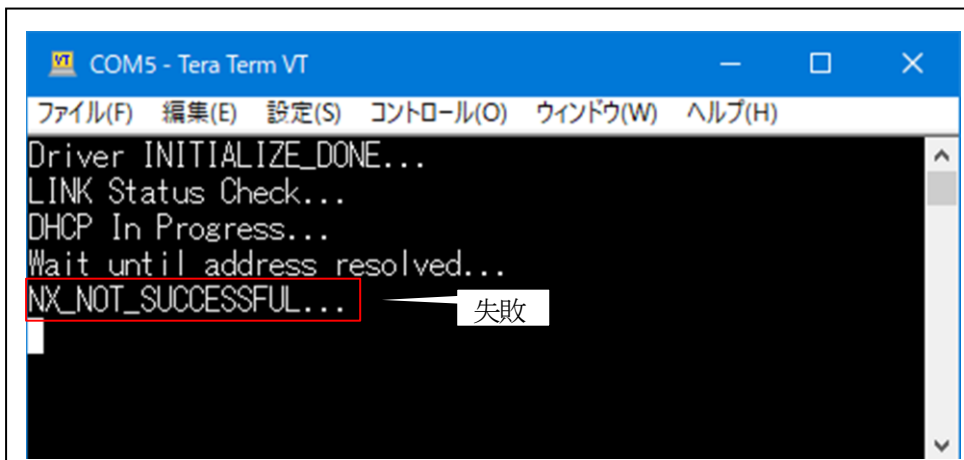


```

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...
LINK Status Check...
<Set Wi Fi ect.spec and emryption>
DHCP In Progress...
Wait until address resolved...

Successfully assigned by DHCP address(PRI)
[MAC ADDRESS]: 84-5d-D7-4A-5B-BD
[MY IP.adr ]: 192.168.21.54 ← 取得した IP アドレス
[SUBNET mask]: 255.255.255.0
[MTU size ]: 1500
[TCP port ]: 50001
[UDP port ]: 50002
  
```

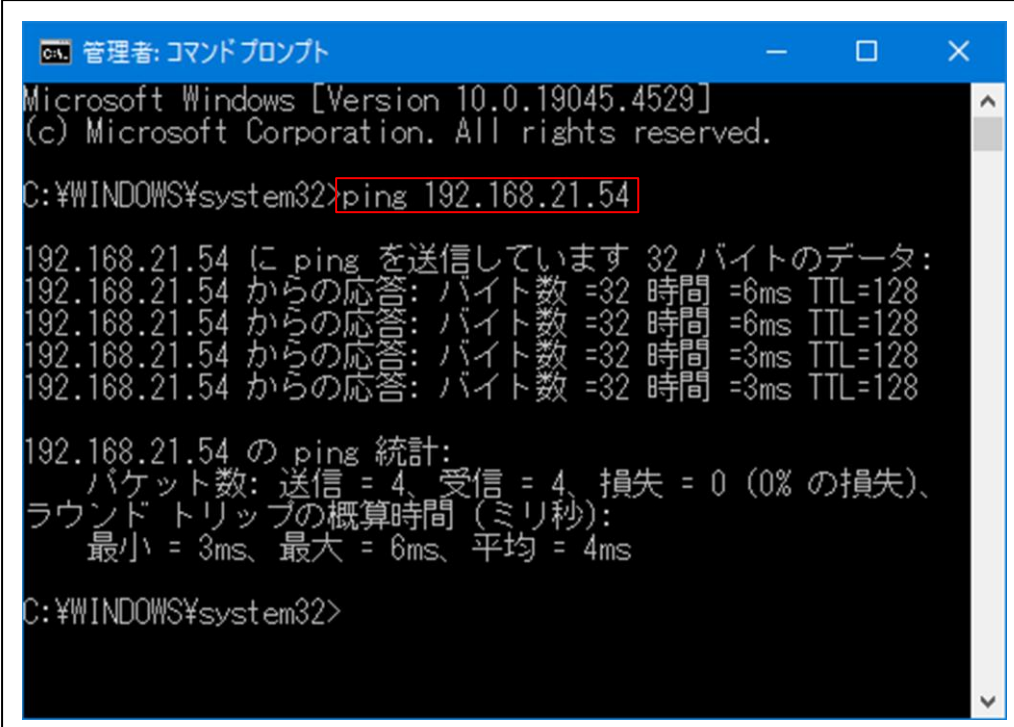
&lt;失敗画面&gt;IP アドレス未確立により、基板上の LED6 (赤色) を 100msec 毎に点滅



```

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...
NX_NOT_SUCCESSFUL... ← 失敗
  
```

3) Windows コマンドプロンプト ping で確認



```

  管理者: コマンドプロンプト
  Microsoft Windows [Version 10.0.19045.4529]
  (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 =6ms TTL=128
  192.168.21.54 からの応答: バイト数 =32 時間 =6ms TTL=128
  192.168.21.54 からの応答: バイト数 =32 時間 =3ms TTL=128
  192.168.21.54 からの応答: バイト数 =32 時間 =3ms 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 = 3ms、最大 = 6ms、平均 = 4ms
  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>

