RZ/A2Mグループ
イーサネット サンプルプラグラム

要旨

本書は RZ/A2M グループ MCU を使用し Ethernet 通信を行うサンプルプログラムについて説明します。

動作確認デバイス

RZ/A2M

目次

1. 概要

イーサネットサンプルプログラム(以下サンプルプログラム)は、RZ/A2M Evaluation Board Kit の Ether2 コネクタから Echo サーバーにデータを出力し、応答を確認するプログラムになります。このプログラムは FreeRTOS-TCP を使用しています。また、通信アプリケーションは FreeRTOS 提供のサンプルコードです。詳細は以下の URL を参照ください。

https://www.freertos.org/FreeRTOS-Plus/FreeRTOS_Plus_TCP/TCP_Echo_Clients.html

表1-1　使用する周辺機能と用途

周辺機能	用途
イーサネットコントローラ(ETHERC)	イーサネットの制御
イーサネットコントローラ用 DMA コントローラ(EDMACa)	イーサネットの制御
SCIF	ターミナルソフトを使用した通信状況出力

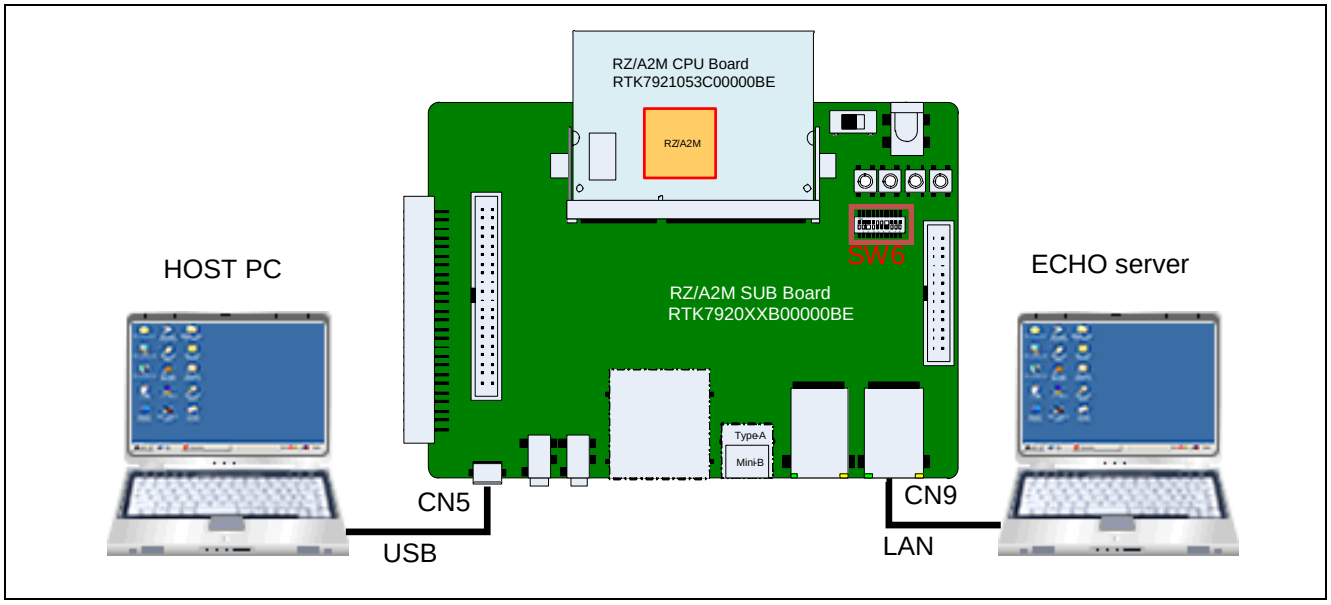


図1.1　動作環境

2. 動作確認条件

本アプリケーションノートのサンプルコードは、下記の条件で動作を確認しています。

表2-1　動作確認条件

項目	内容
使用 MCU	RZ/A2M
動作周波数（注）	CPU クロック（Iφ）：528MHz 画像処理クロック（Gφ）：264MHz 内部バスクロック（Bφ）：132MHz 周辺クロック 1（P1φ）：66MHz 周辺クロック 0（P0φ）：33MHz QSPI0_SPCLK：66MHz CKIO：132MHz
動作電圧	電源電圧（I/O）：3.3V 電源電圧（1.8/3.3V 切替 I/O（PVcc_SPI））：3.3V 電源電圧（内部）：1.2V
統合開発環境	e2 studio V7.4.0
C コンパイラ	GNU Arm Embedded Toolchain 6-2017-q2-update
	コンパイラオプション（ディレクトリパスの追加は除く） Release: -mcpu=cortex-a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mfpu=neon -mno-unaligned-access -Os -ffunction-sections -fdata-sections -Wunused -Wuninitialized -Wall -Wextra -Wmissing-declarations -Wconversion -Wpointer-arith -Wpadded -Wshadow -Wlogical-op -Waggregate-return -Wfloat-equal -Wnull-dereference -Wmaybe-uninitialized -Wstack-usage=100 -fabi-version=0 Hardware Debug: -mcpu=cortex-a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mfpu=neon -mno-unaligned-access -Og -ffunction-sections -fdata-sections -Wunused -Wuninitialized -Wall -Wextra -Wmissing-declarations -Wconversion -Wpointer-arith -Wpadded -Wshadow -Wlogical-op -Waggregate-return -Wfloat-equal -Wnull-dereference -Wmaybe-uninitialized -g3 -Wstack-usage=100 -fabi-version=0
動作モード	ブートモード 3 （シリアルフラッシュブート 3.3V 品）
ターミナルソフトの通信設定	• 通信速度：115200bps • データ長：8 ビット • パリティ：なし • ストップビット長：1 ビット • フロー制御：なし
使用ボード	RZ/A2M CPUボード　RTK7921053C00000BE RZ/A2M SUBボード　RTK79210XXB00000BE
使用デバイス （ボード上で使用する機能）	• シリアルフラッシュメモリ（SPI マルチ I/O バス空間に接続） メーカ名：Macronix 社、型名：MX25L51245GXD • RL78/G1C（USB 通信とシリアル通信を変換し、ホスト PC との通信に使用） • Ethernet PHY RTL8201FL-VB-CG（Realtek）

【注】 クロックモード 1（EXTAL 端子からの 24MHz のクロック入力）で使用時の動作周波数です。

3. ソフトウェア説明

本サンプルプログラムのシステムブロックを図 3-1に記載します。本サンプルプログラムは、FreeRTOS 標準のタイマタスクから Echo Client Task が生成され、Echo Client Task 内の通信アプリケーションで Echo Server との通信を行います。

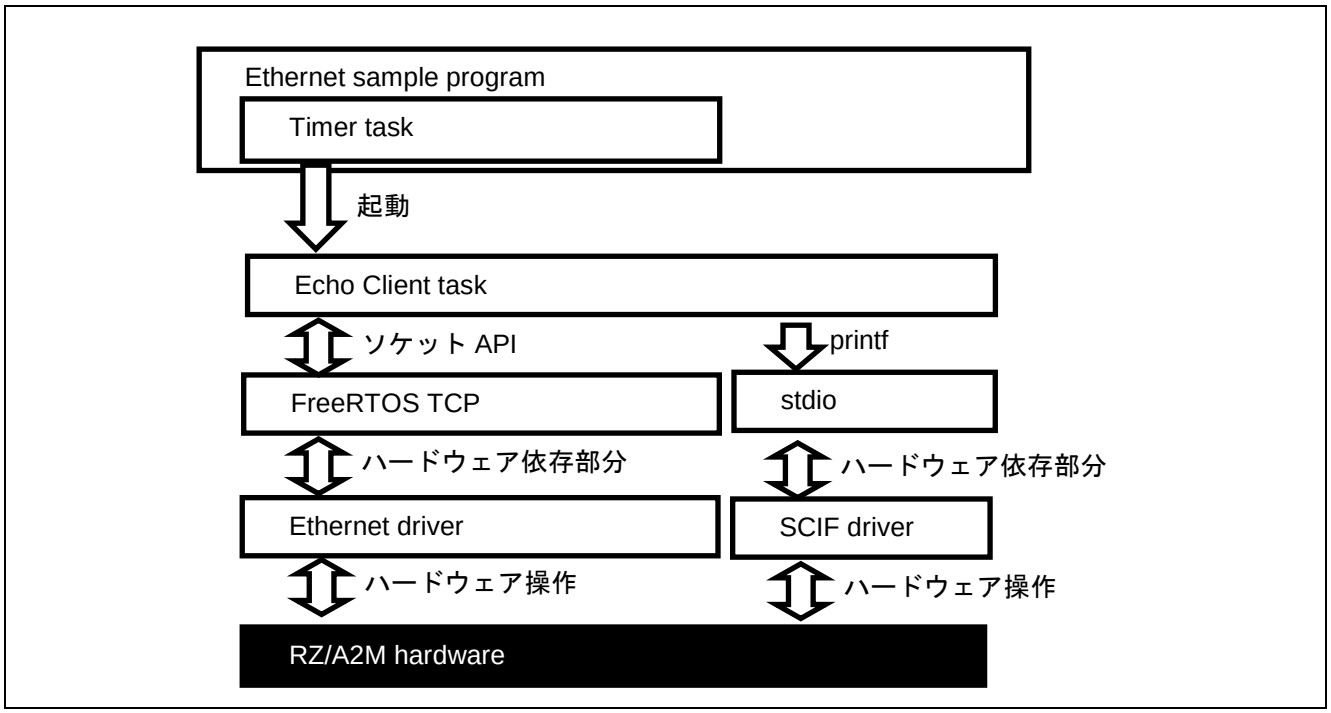


図 3-1 サンプルプログラムのシステムブロック

4.　サンプルプログラム動作確認方法

サンプルプログラムを動作させるためには、Echo Server を用意する必要があります。本章ではサンプルプログラムを動作させるために必要な設定について記載します。

4.1　Echo Server の設定

Windows 10 PC を Echo Server として使用する方法を記載します。

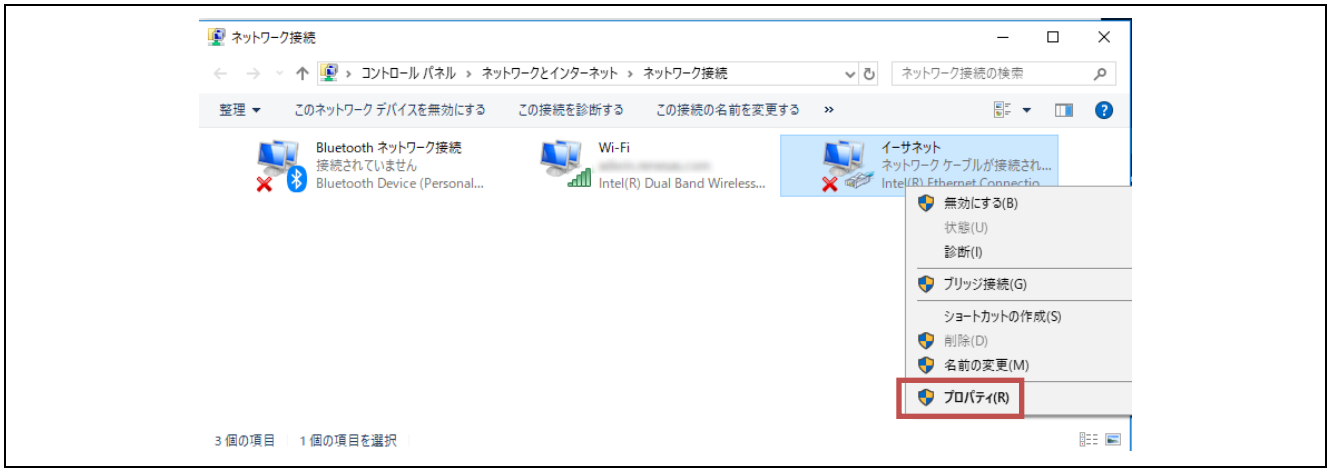
4.1.1　IP アドレスの変更

スタートメニューから「Windows システムツール」→「コントロールパネル」を選択します。

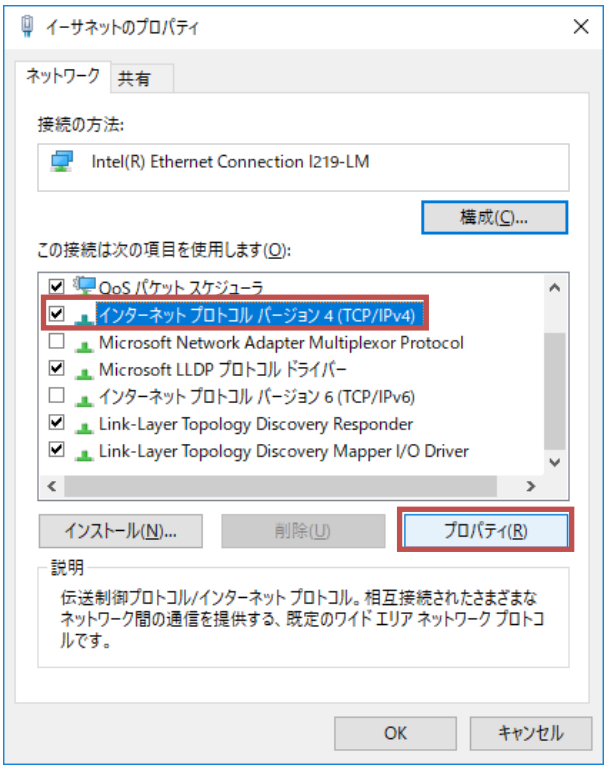
「ネットワークとインターネット」の「ネットワークの状態とタスクの表示」をクリックします。

「アダプターの設定の変更」をクリックします。

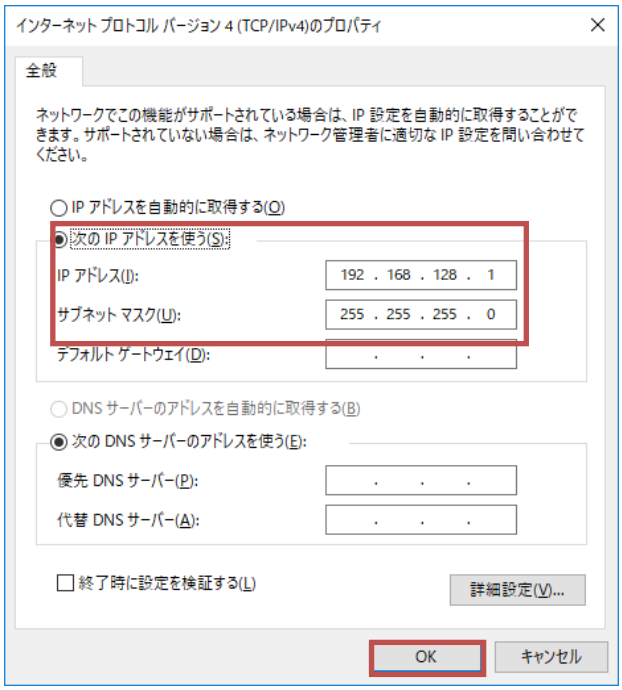
「イーサネット」を右クリックし、「プロパティ」を選択します。



「インターネット プロトコル バージョン 4(TCP/IPv4)」を選択し、「プロパティ」をクリックします。



「次の IP アドレスを使う」を選択し、IP アドレスに「192.168.128.1」を、サブネット マスクに「255.255.255.0」をそれぞれ入力し、「OK」をクリックします。



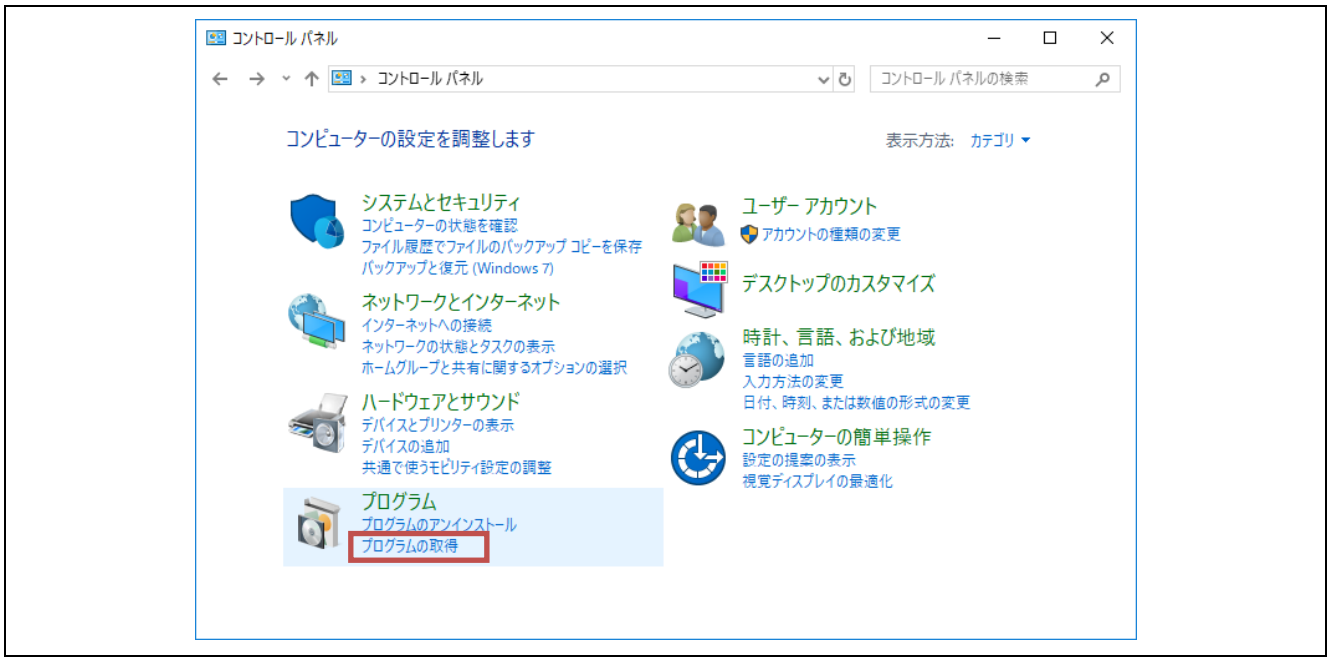
4.1.2　簡易 TCP/IP サービスのインストール

Windows を Echo サーバー化するために、Windows に付属している簡易 TCP/IP サービスをインストールします。このサービスをインストールすることで Windows に Echo 機能が追加されます。

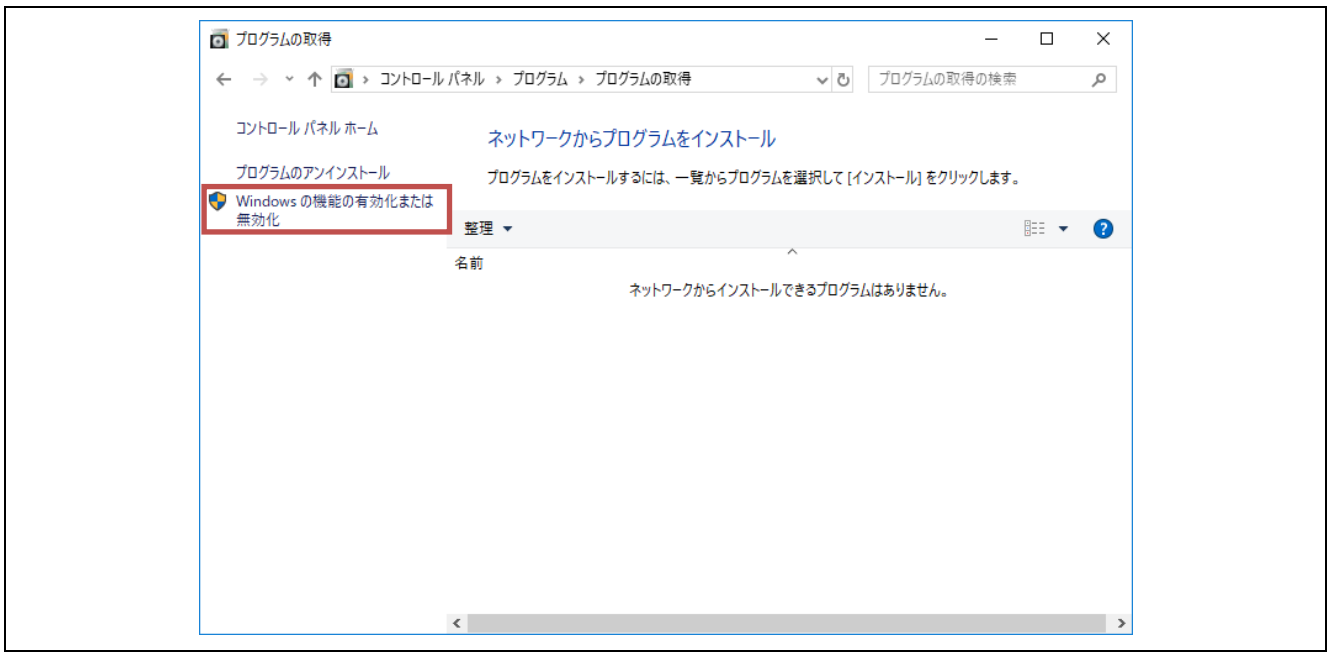
簡易 TCP/IP サービスはセキュリティホールを生み出す要因になりえるため、動作確認完了後は、このサービスをアンインストールすることをお勧めいたします。

スタートメニューから「Windows システムツール」→「コントロールパネル」を選択します。

「プログラム」の「プログラムの取得」をクリックします。

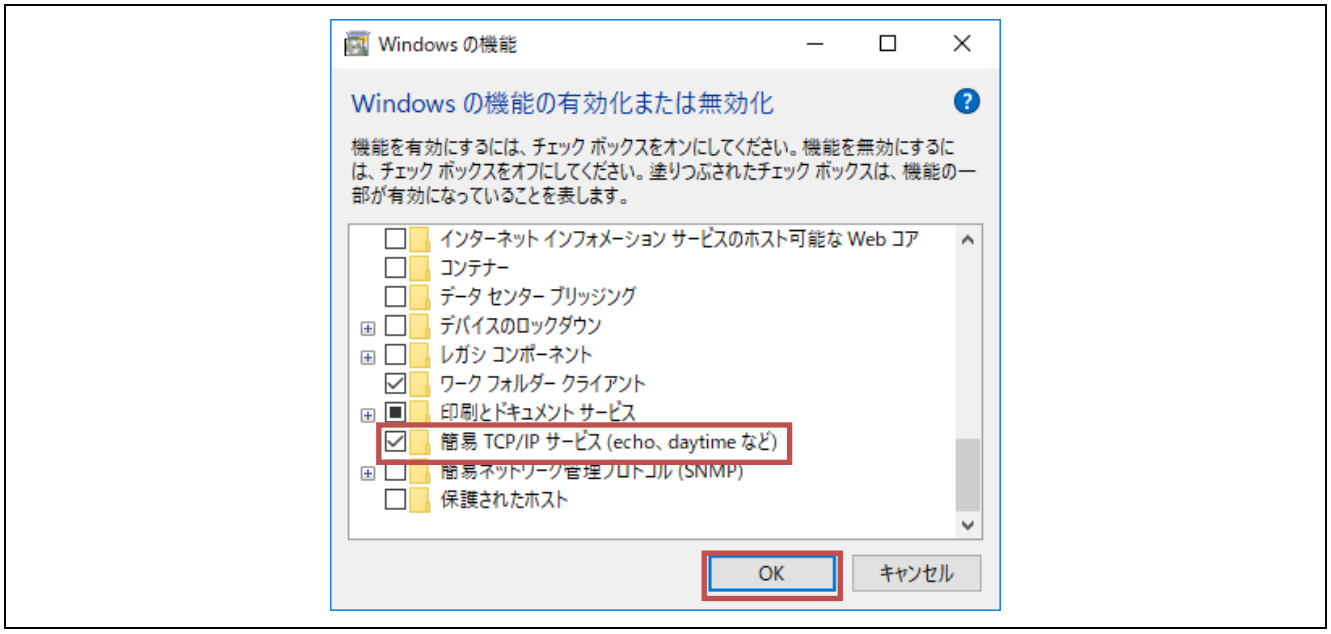


「Windows の機能の有効化または無効化」をクリックします。



「簡易 TCP/IP サービス(echo、daytime など)」にチェックを入れ、「OK」をクリックします。

動作確認後、チェックを外して「OK」をクリックすると、簡易 TCP/IP サービスがアンインストールされます。



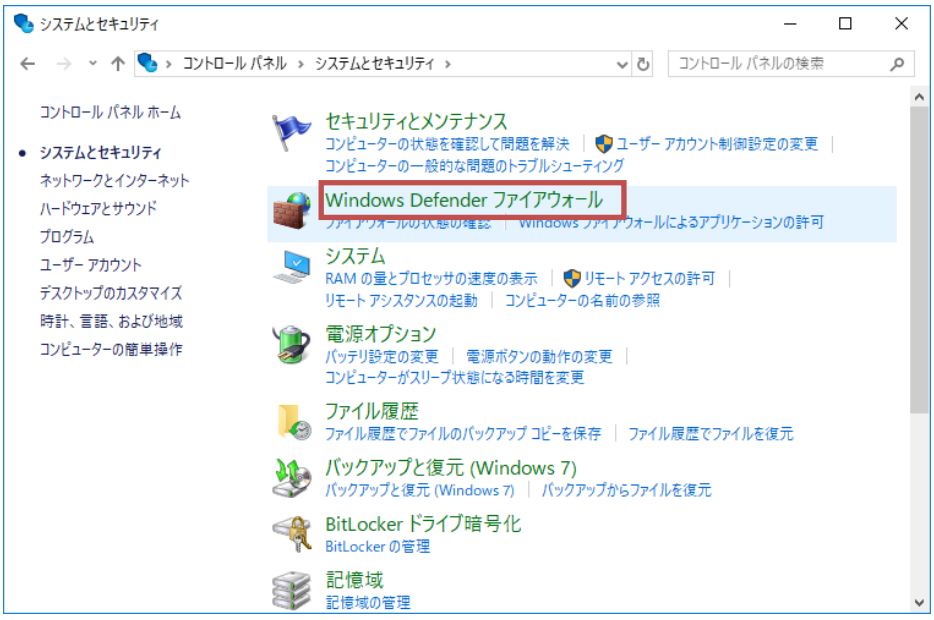
4.1.3　Firewall の設定

簡易 TCP/IP サービス内の Echo 機能が用いる TCP ポート 7 を外部との通信で使用できるようにします。

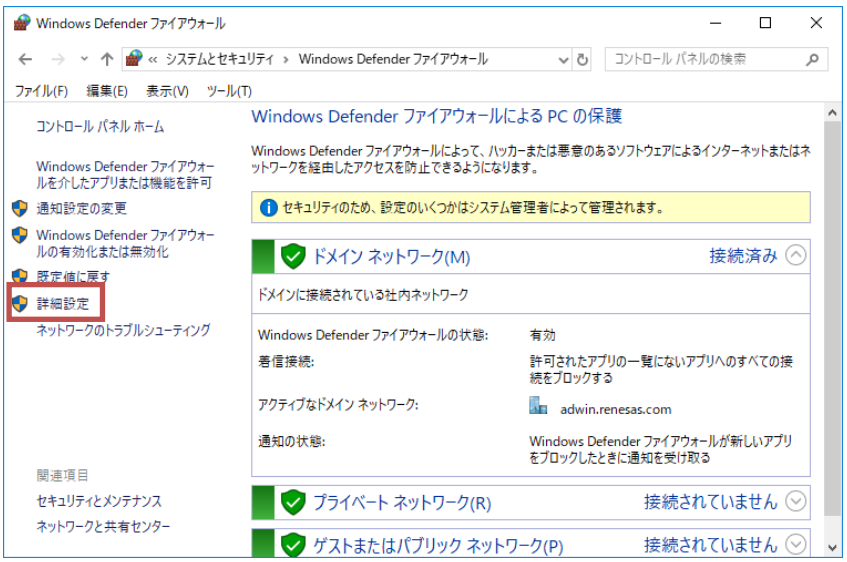
スタートメニューから「Windows システムツール」→「コントロールパネル」を選択します。

「システムとセキュリティ」をクリックします。

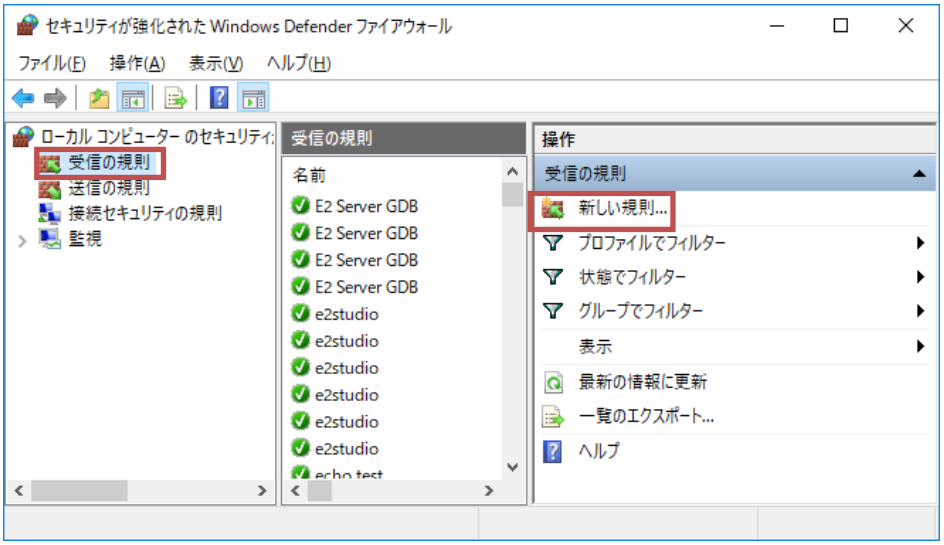
「Windows Defender ファイアウォール」をクリックします。



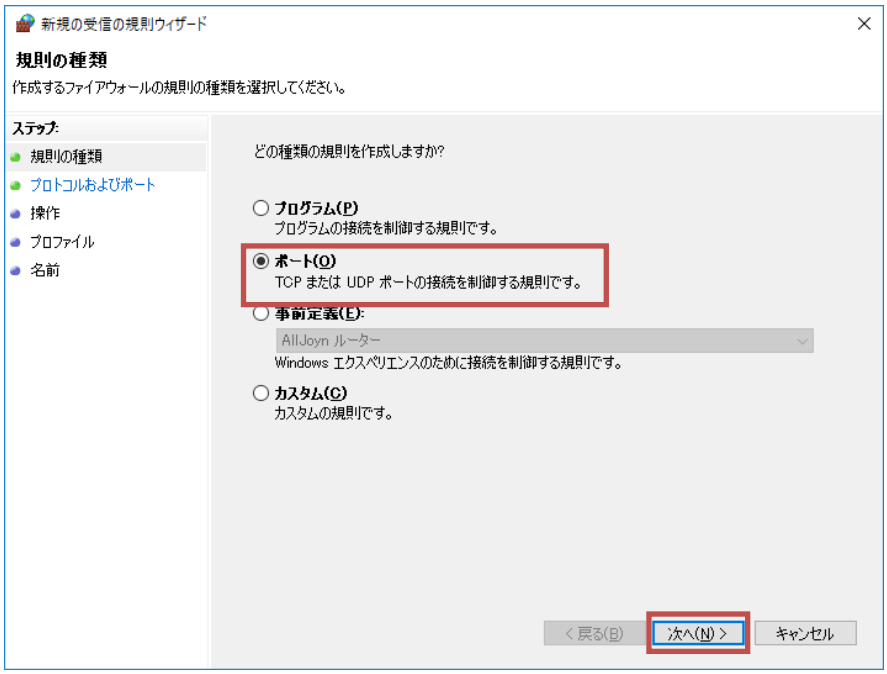
「詳細設定」をクリックします。



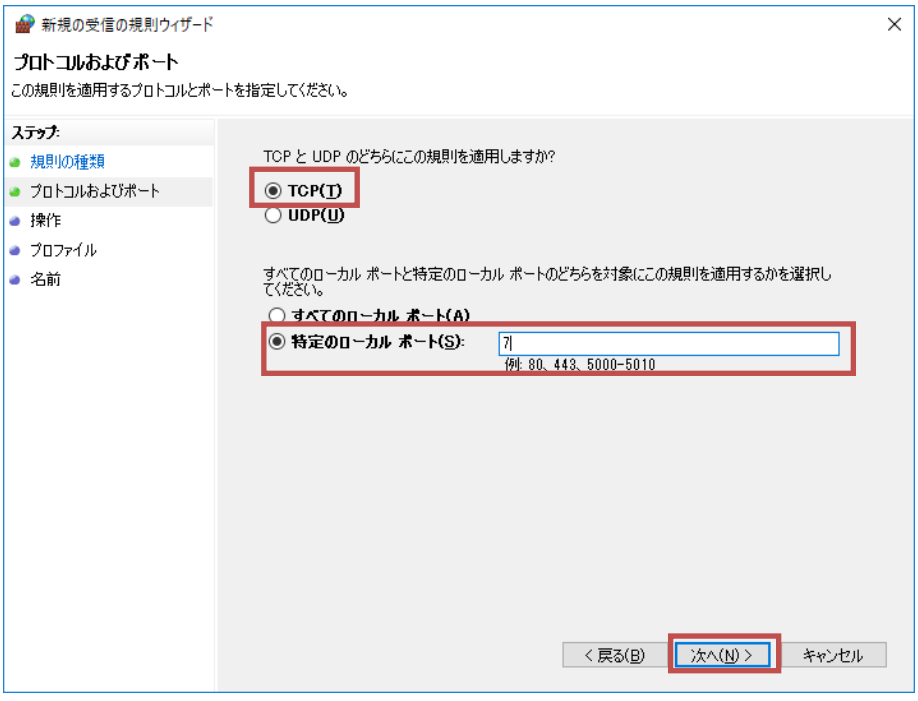
「受信の規則」を選択し、「新しい規則」をクリックします。



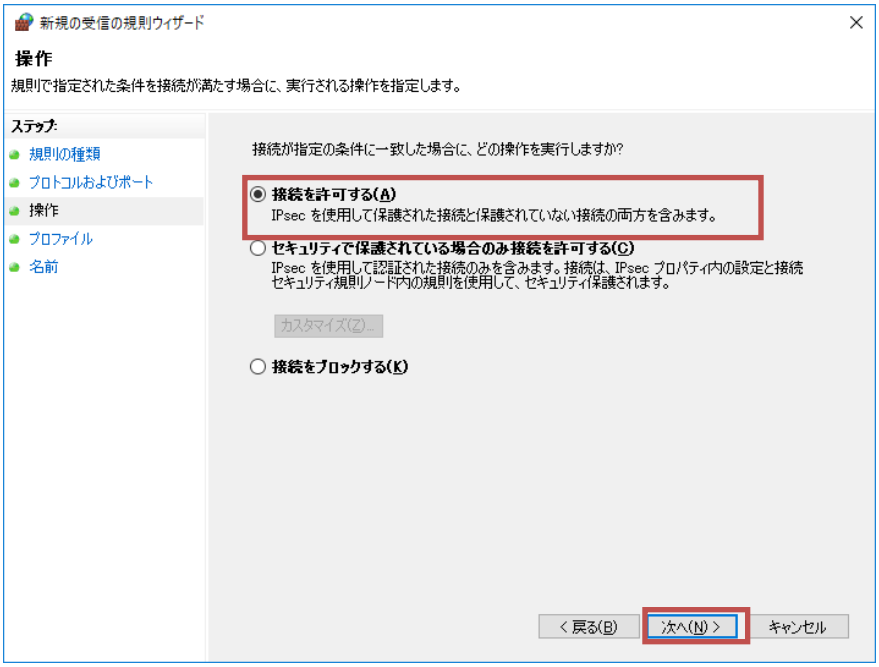
「ポート」を選択し、「次へ(N)」をクリックします。



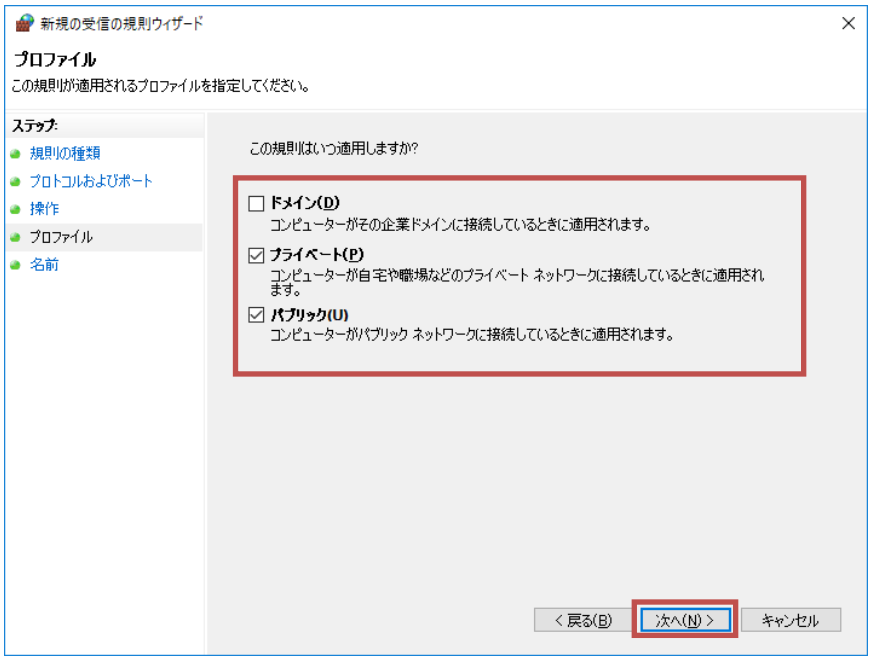
「TCP」と「特定のローカルポート」を選択し、「7」を入力し。「次へ(N)」をクリックします。



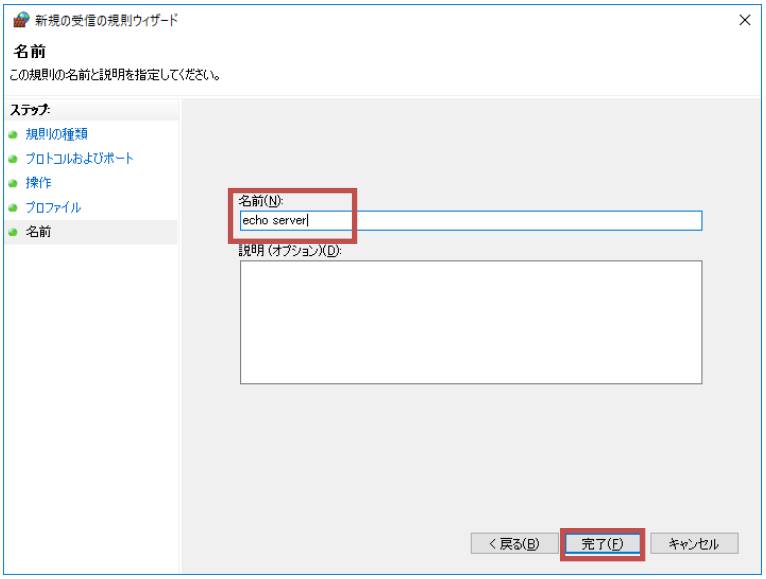
「接続を許可する」を選択し、「次へ(N)」をクリックします。



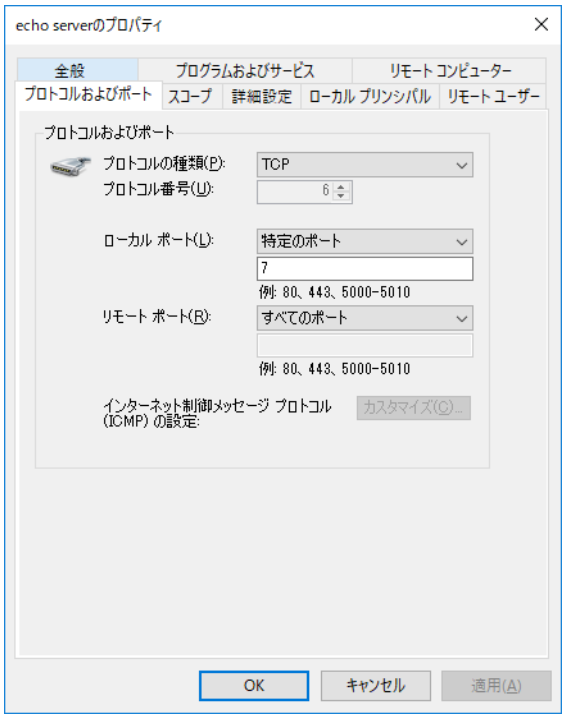
「プライベート」と「パブリック」にチェックを入れ、「次へ(N)」をクリックします。



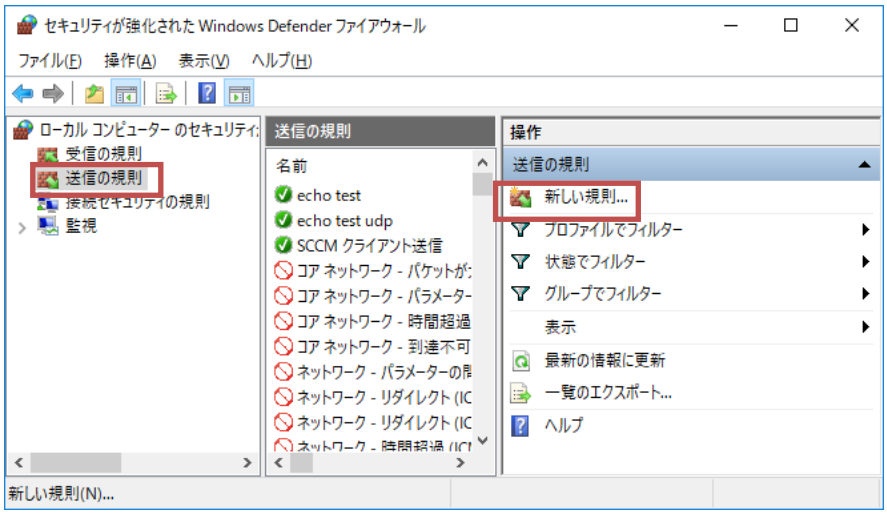
「名前」に任意の名前を入力し、「完了(F)」をクリックします。



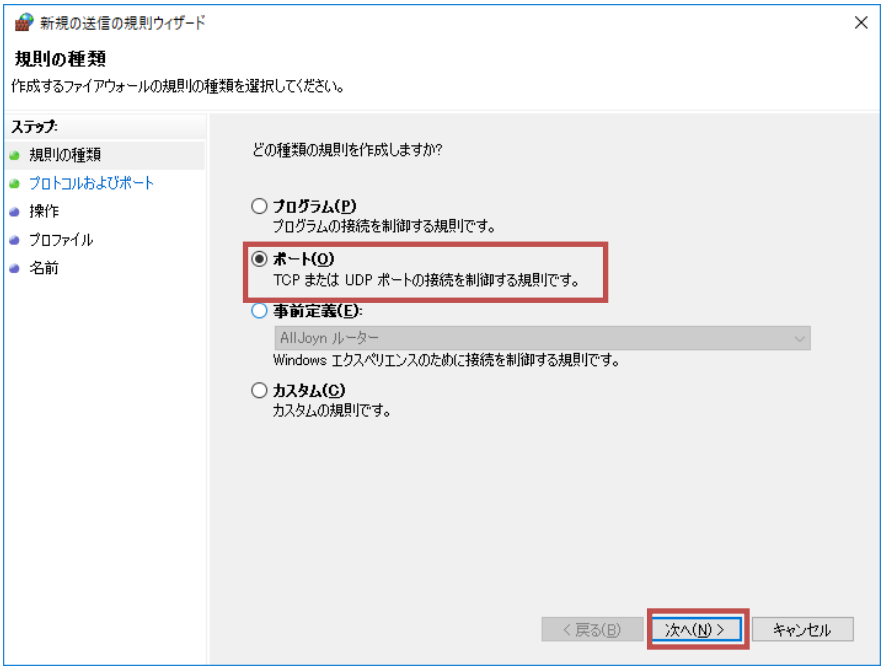
作成した規則をダブルクリックし、「プロトコルおよびポート」を選択すると、以下の図のような設定になっていることを確認できます。



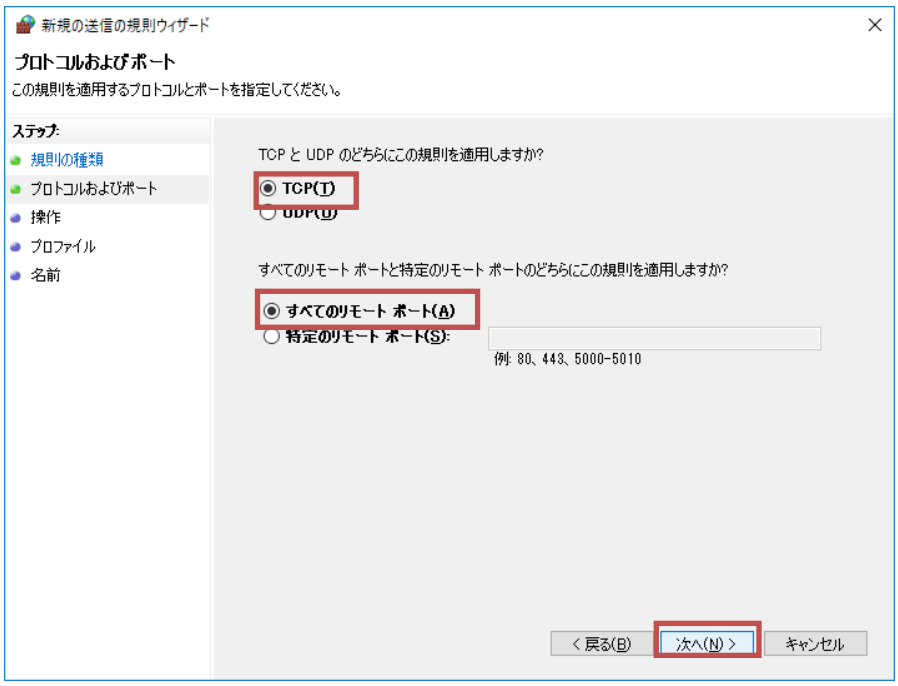
「送信の規則」を選択し、「新しい規則」をクリックします。



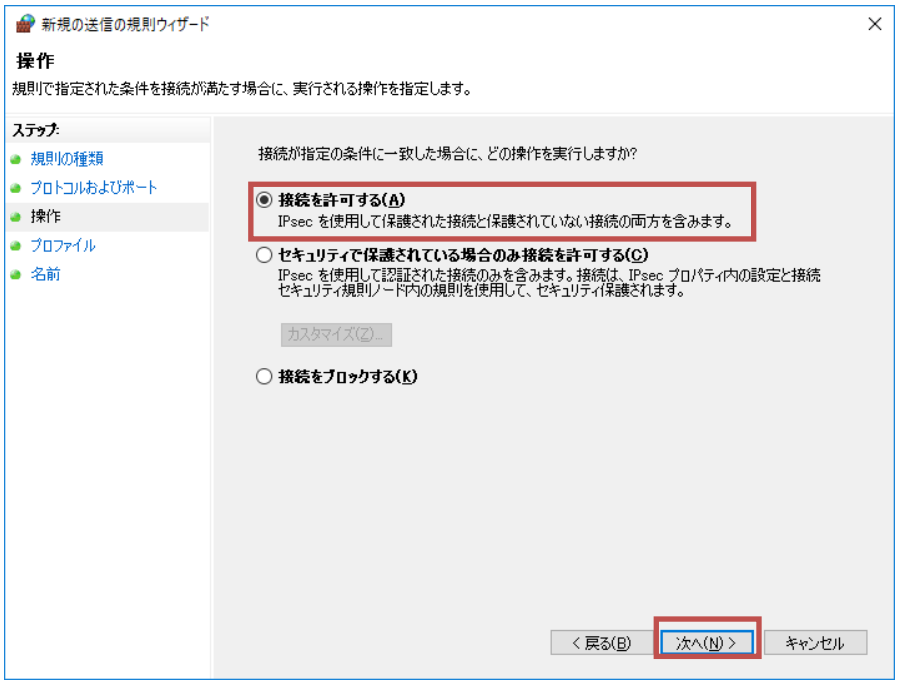
「ポート」を選択し、「次へ(N)」をクリックします。



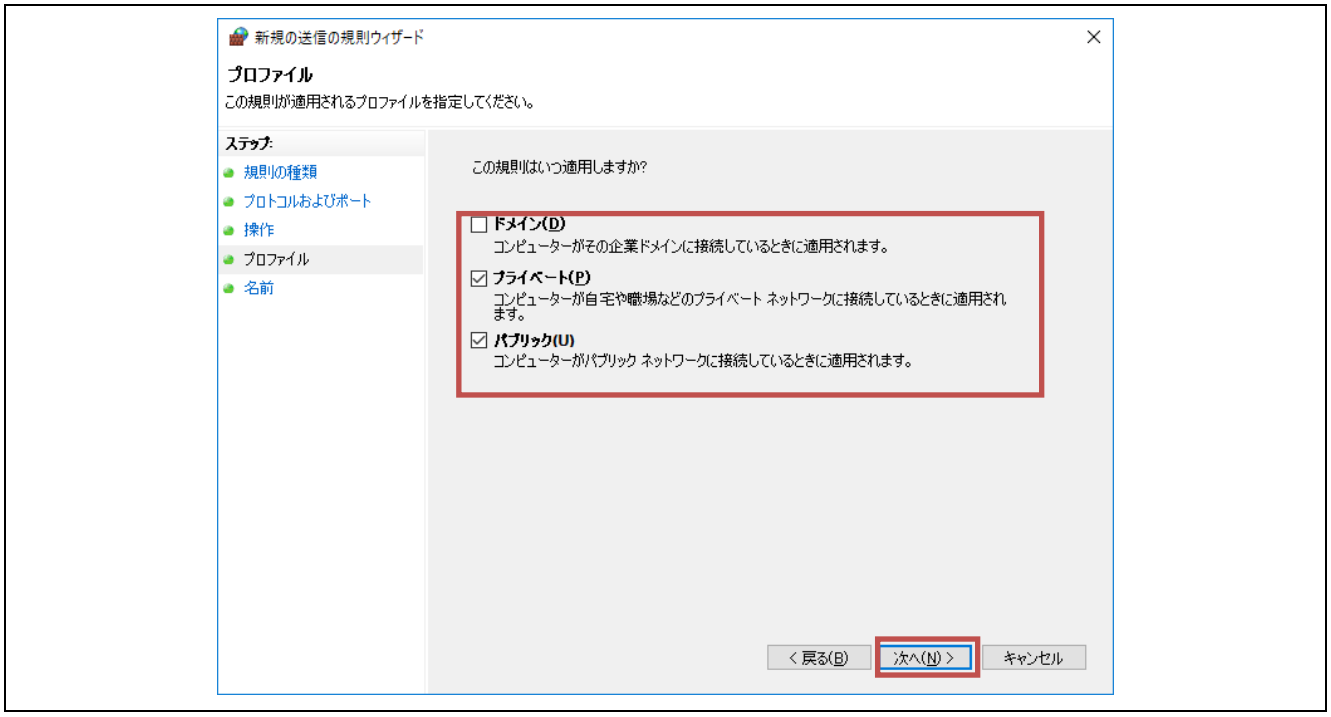
「TCP」と「すべてのリモート ポート」を選択し、「次へ(N)」をクリックします。



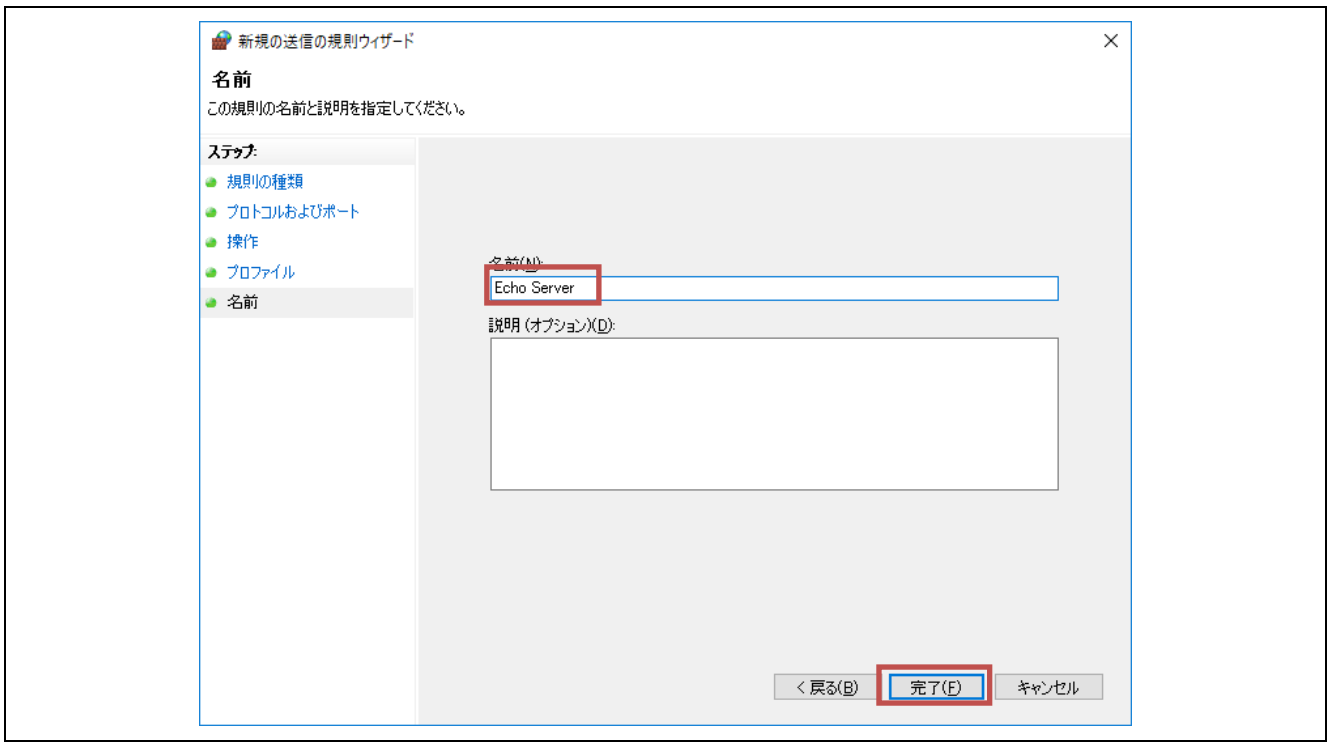
「接続を許可する」を選択し、「次へ(N)」をクリックします。



「プライベート」と「パブリック」にチェックを入れ、「次へ(N)」をクリックします。

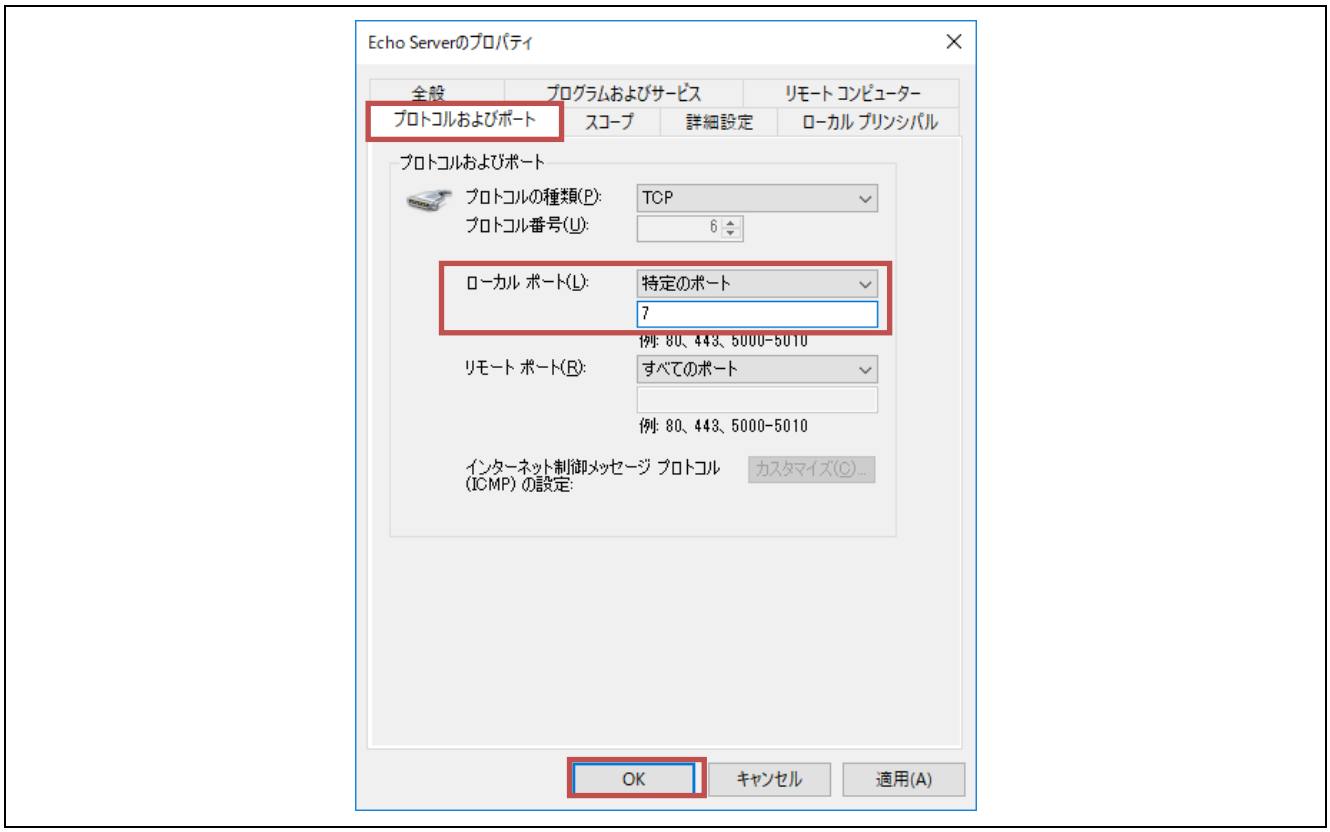


「名前」に任意の名前を入力し、「完了(F)」をクリックします。



作成した規則をダブルクリックし「プロトコルおよびポート」を選択します。

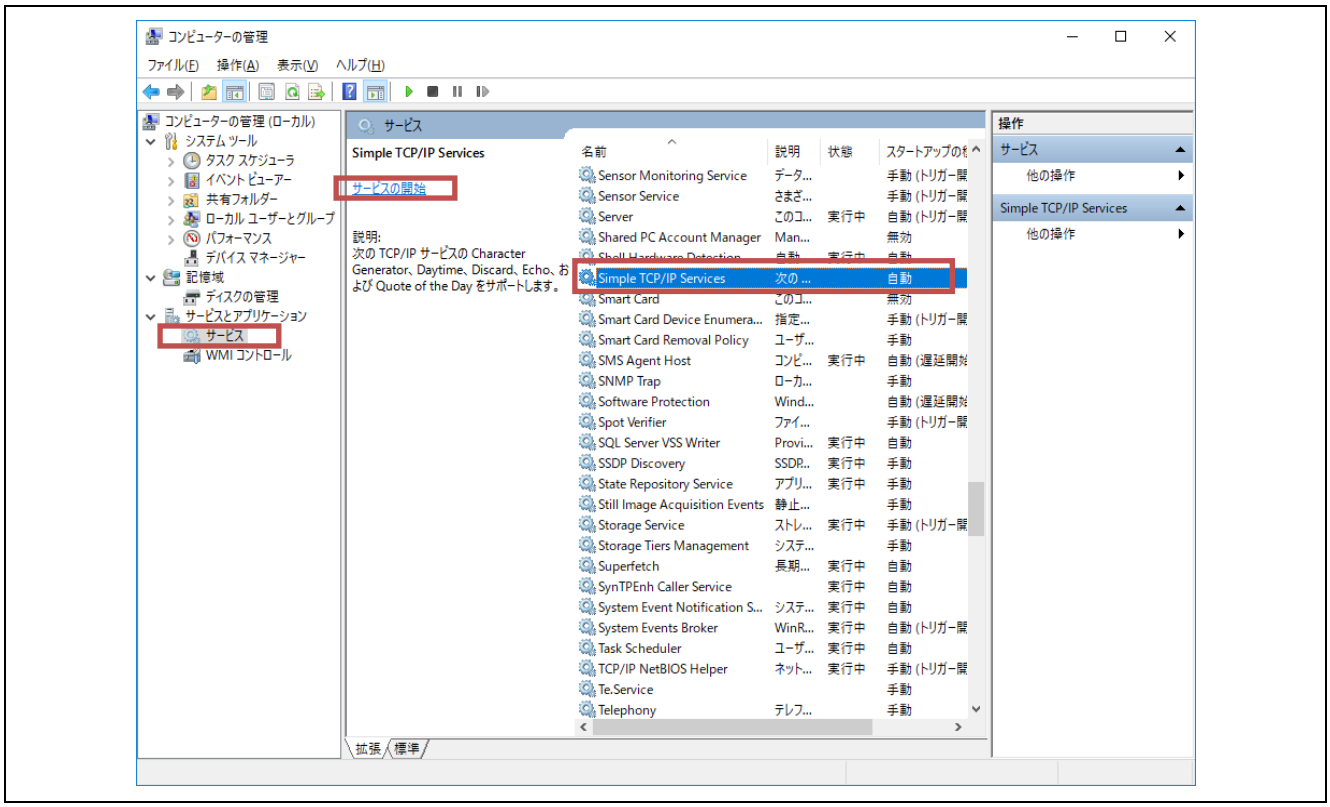
「ローカルポート」を「特定のポート」とし「7」を入力し、「OK」をクリックします。



4.1.4　簡易 TCP/IP サービスの起動

スタートメニュー上で右クリックし、「コンピューターの管理」をクリックします。

「サービスとアプリケーション」から「サービス」を選択し、「Simple TCP/IP Services」を選択します。
「サービスの開始」をクリックします。



4.2 HOST PC の設定

HOST PC のターミナルソフトは、以下の設定を行ってください。

表 4-1 HOST PC のターミナルソフトの設定

項目	設定値
シリアル ボー・レート	115,200bps
シリアル データ長	8bit
シリアル パリティビット有無	なし
シリアル ストップビット長	1bit
シリアル フロー制御有無	なし
受信時改行コード	LF

4.3　動作確認結果

RZ/A2M Evaluation Board Kit にビルドしたサンプルプログラムをダウンロードし、実行すると HOST PC のターミナルソフトに図 4-1の出力を確認できます。サンプルプログラムは、Echo サーバーに出力したメッセージと Echo サーバーから受信したメッセージが同一であるかを確認しています。

```
RZ/A2M network_sample for GCC Ver. 1.0
Copyright (C) 2018 Renesas Electronics Corporation. All rights reserved.
Build Info Date Dec 21 2018 at 15:01:32
FreeRTOS OS Abstraction Version 3.0
Connecting to echo server
Echo demo failed to connect to echo server 192.168.128.1.
Connecting to echo server
Connected to echo server
Sending TxRx message number 0 of length 1 to echo server
Received correct string from echo server.
Sending TxRx message number 1 of length 2 to echo server
Received correct string from echo server.
Sending TxRx message number 2 of length 3 to echo server
Received correct string from echo server.
Sending TxRx message number 3 of length 4 to echo server
Received correct string from echo server.
Sending TxRx message number 4 of length 5 to echo server
Received correct string from echo server.
Sending TxRx message number 5 of length 6 to echo server
Received correct string from echo server.
Sending TxRx message number 6 of length 7 to echo server
Received correct string from echo server.
Sending TxRx message number 7 of length 8 to echo server
Received correct string from echo server.
Sending TxRx message number 8 of length 9 to echo server
Received correct string from echo server.
Sending TxRx message number 9 of length 10 to echo server
Received correct string from echo server.
Shutting down connection to echo server.
Connecting to echo server
Connected to echo server
Sending TxRx message number 10 of length 11 to echo server
Received correct string from echo server.
                                    :
                                    :
                                    :
```

Echo サーバーと接続失敗
Echo サーバーと接続成功
メッセージ 0-9 送受信成功
10 メッセージで一度サーバーと切断
以後 10 メッセージごとに再接続しながらメッセージ 10 以降を繰り返す

図 4-1 通信ログ

5. Amazon Web Service への接続

本章では、MQTTEcho という Amazon 社から提供されているデモを用いて Amazon Web Service(AWS)への接続する方法について示します。

5.1 AWS の設定

AWS のアカウントを作成します。AWS アカウントの作成は以下リンクをブラウザで表示して行います。

https://portal.aws.amazon.com/billing/signup#/start

AWS アカウントを作成後、AWS の IoT Core サービスにてモノの作成を行います。

IoT Core サービスのモノを作成するには、このセクションの手順に従ってください。

5.1.1 AWS の IoT Core サービスを開く

1. AWS コンソールにサインインして AWS サービスの検索バーに IoT を入力します。
2. IoT Core をクリックします。

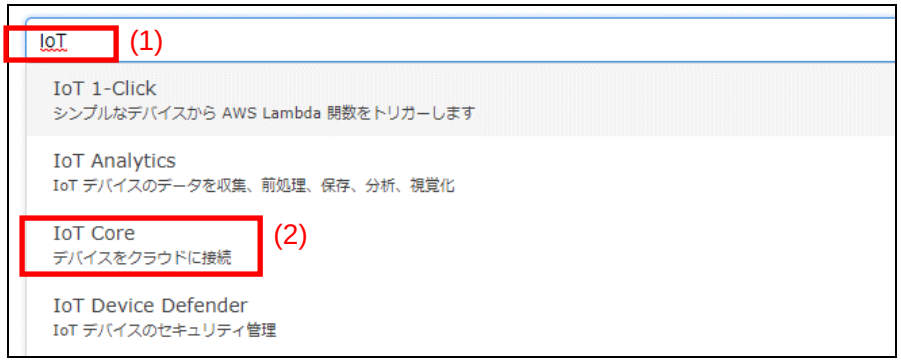


図 5-1 IoT Core サービス選択

5.1.2　AWS IoT モノ の作成

1. 左側にある **管理** を選択します。
2. **モノ** を選択します。
3. **モノの登録** を選択します。既に他にモノが存在する場合は図 5-3の**作成**を選択します。

図 5-2 モノの登録画面（初めてモノを登録する場合）

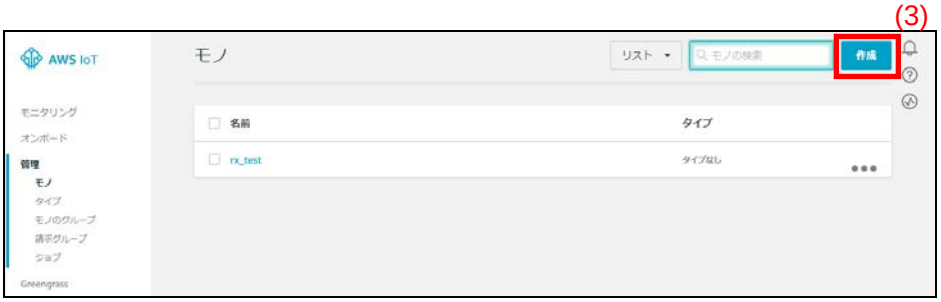


図 5-3 モノの登録画面（既にモノが存在する場合）

4. **単一のモノを作成する** を選択します。

図 5-4 モノを作成する 画面

5. **モノの名前** （例：rz_test） を入力します。
6. **次へ** を選択します。

図 5-5 デバイス追加 画面

7. 1-Click 証明書作成（推奨）の **証明書の作成** を選択します。

図 5-6 デバイス追加 画面

8. **このモノの証明書** をダウンロードします。
9. **パブリックキー** をダウンロードします。
10. **プライベートキー** をダウンロードします。
11. **有効化** を選択します。
12. **完了** を選択します。

図 5-7 証明書作成 画面

5.1.3　AWS IoT ポリシー の作成

1. 左側にある **安全性** を選択します。
2. **ポリシー** を選択します。
3. **ポリシーの作成** を選択します。既に他にポリシーが存在する場合は**作成**を選択します。

図 5-8 ポリシーの作成開始 画面

4. **ポリシーの名前** （例：rz_echo_test） を入力します。

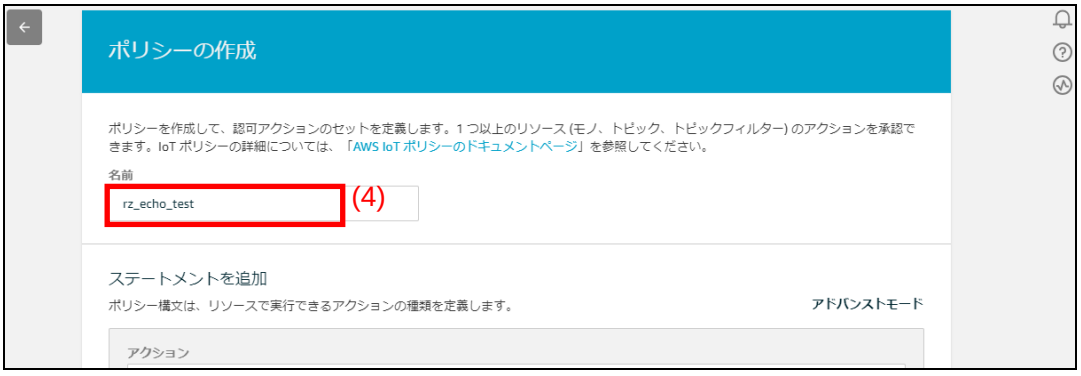


図 5-9 ポリシーの作成 名前入力 画面

5. **ステートメントを追加**のアクションに iot:Connect を入力します。
6. **リソース ARN** の replaceWithAClientid 部分を Amazon FreeRTOS プロジェクト内 <root>/demos/common/mqtt/aws_hello_world.c の #define echoCLIENT_ID に記述された文字列に置き換えます。
7. **効果 許可** を選択します。

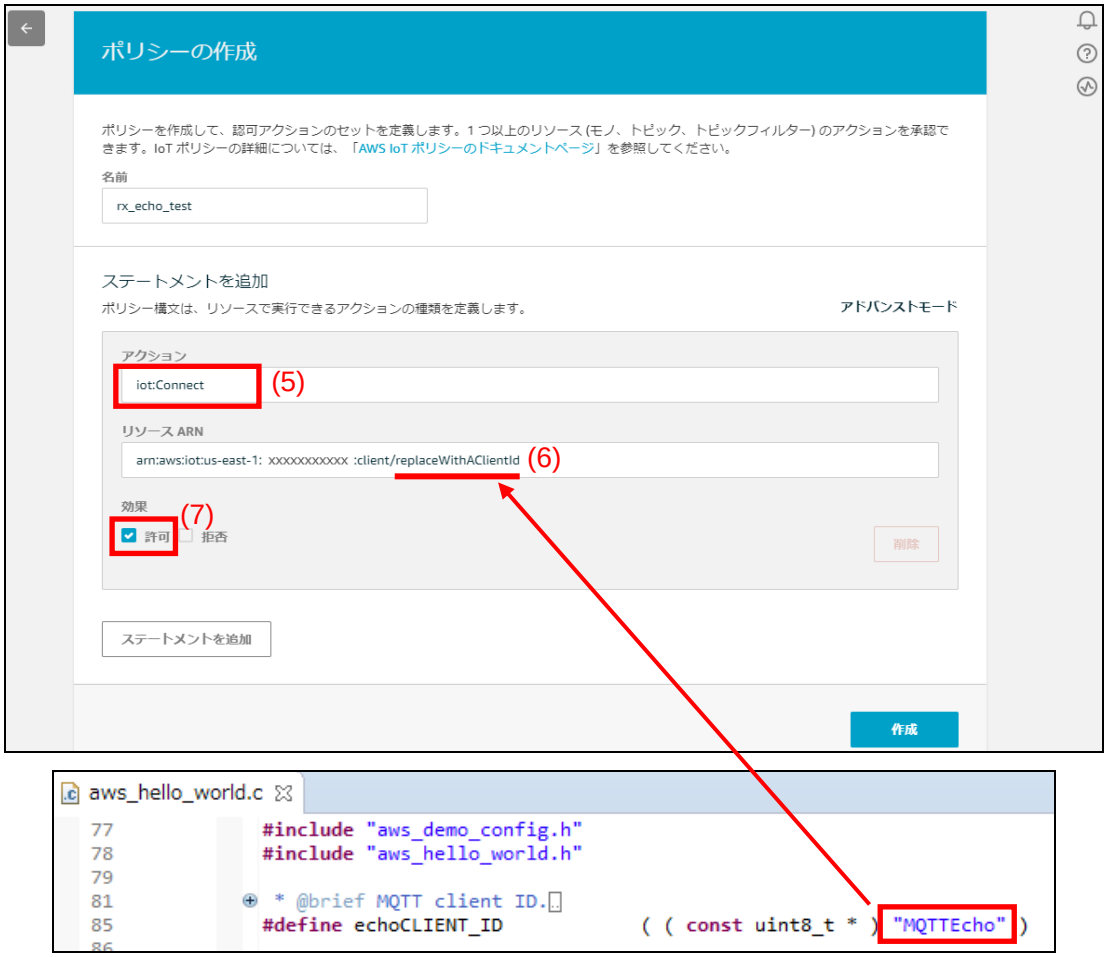


図 5-10 ステートメント入力 画面（置換前）

図 5-11 ステートメント入力 画面（置換後）

8. **ステートメントを追加** を選択します。

図 5-12 ステートメント入力 画面（ステートメント追加）

9. **アクション** に iot:Publish を入力します。
10. **リソース ARN** の replaceWithATopic 部分を Amazon FreeRTOS プロジェクト内 <root>/demos/common/mqtt/aws_hello_world.c の #define echoTOPIC_NAME に記述された文字列に置き換えます。
11. **効果** **許可** を選択します。

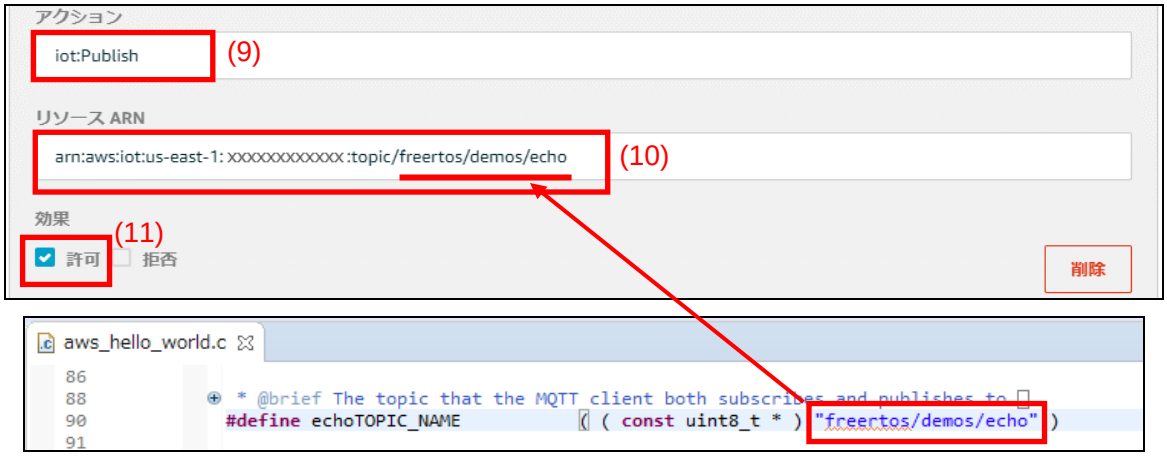


図 5-13 ステートメント入力 画面（iot:Publish 追加）

12. 8.～11. を繰り返し、iot:Subscribe、iot:Receive のステートメントを追加します。
13. **作成** を選択します。

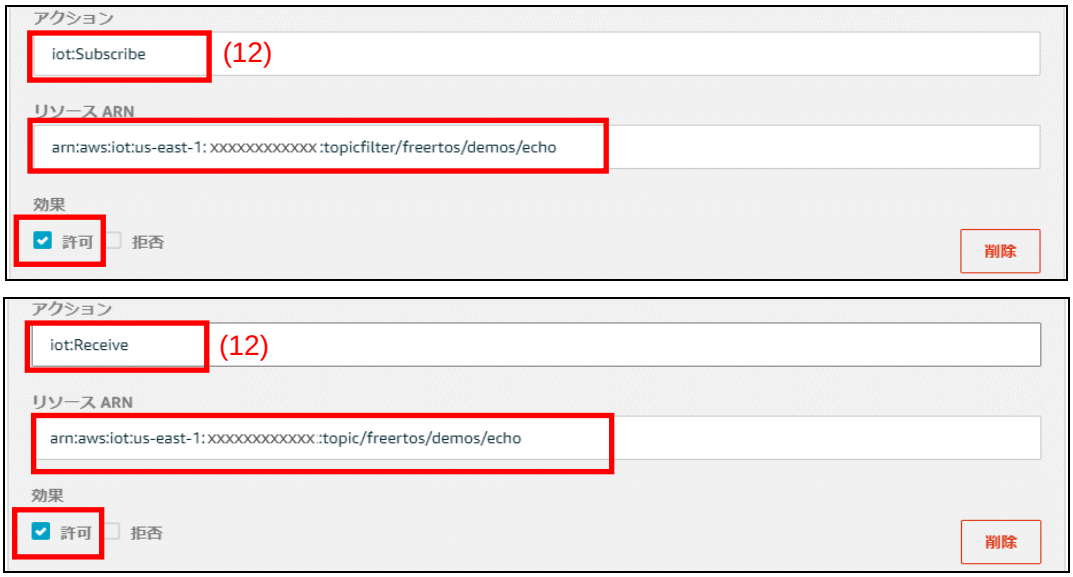


図 5-14 ステートメント入力 画面（iot:Subscribe, iot:Receive 追加）

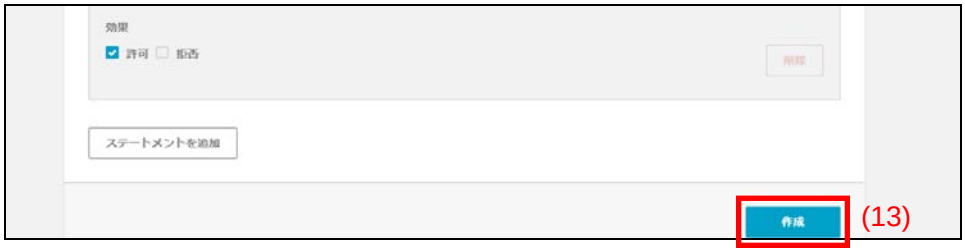


図 5-15 ポリシー作成完了 画面

5.1.4 AWS IoT ポリシー を証明書へアタッチ

1. 左側にある **安全性** を選択します。
2. **証明書** を選択します。
3. 5.1.2で作成した証明書の右上の[…]を選択して、ドロップダウンメニューを表示します。
4. **ポリシーのアタッチ** を選択します。

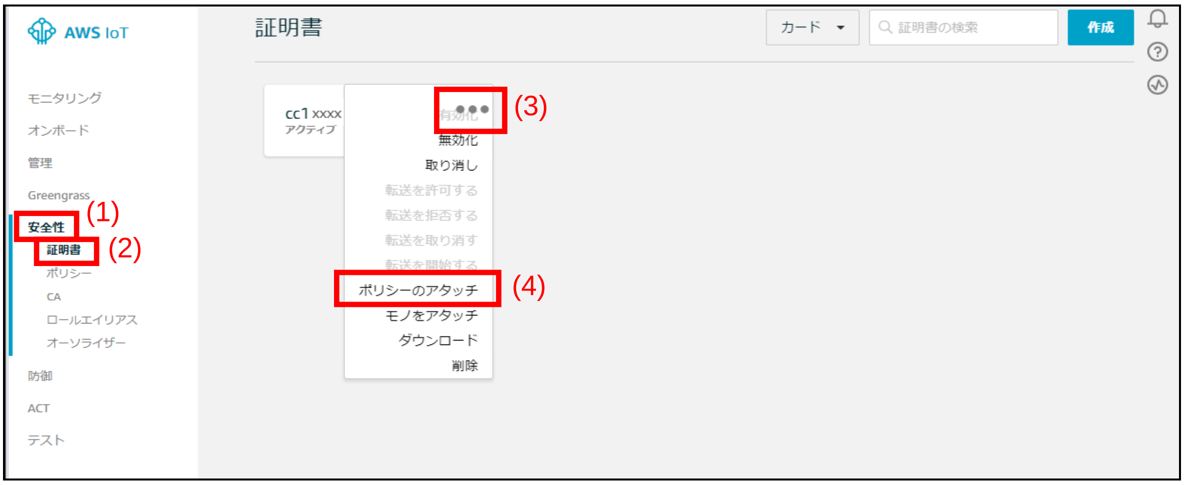


図 5-16 ポリシーのアタッチ開始 画面

5. 5.1.3で作成した**ポリシー** を選択します。
6. **アタッチ** を選択します。

図 5-17 ポリシーのアタッチ選択 画面

5.2　認証情報設定

RZ/A2M 用 Amazon FreeRTOS プロジェクトに AWS IoT の情報を設定するにはこのセクションの手順に従ってください。

5.2.1　AWS IoT 情報の設定

1. AWS IoT の左側にある **設定** を選択します。
2. **エンドポイント** を確認します。
3. <root>/demos/common/include/aws_clientcredential.h の clientcredentialMQTT_BROKER_ENDPOINT[]にエンドポイントを入力します。

図 5-18 エンドポイントの設定

4. AWS IoT の左側にある **管理** を選択します。
5. **モノ** を選択します。
6. **モノの名前** を確認します。
7. <root>/demos/common/include/aws_clientcredential.h の clientcredentialIOT_THING_NAME にモノの名前 を入力します。

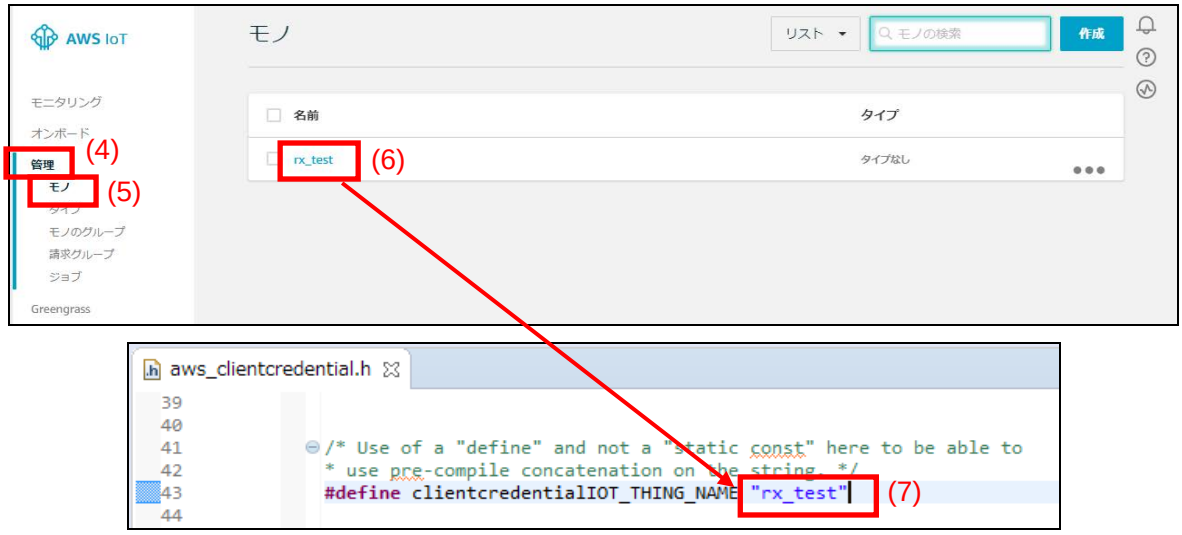
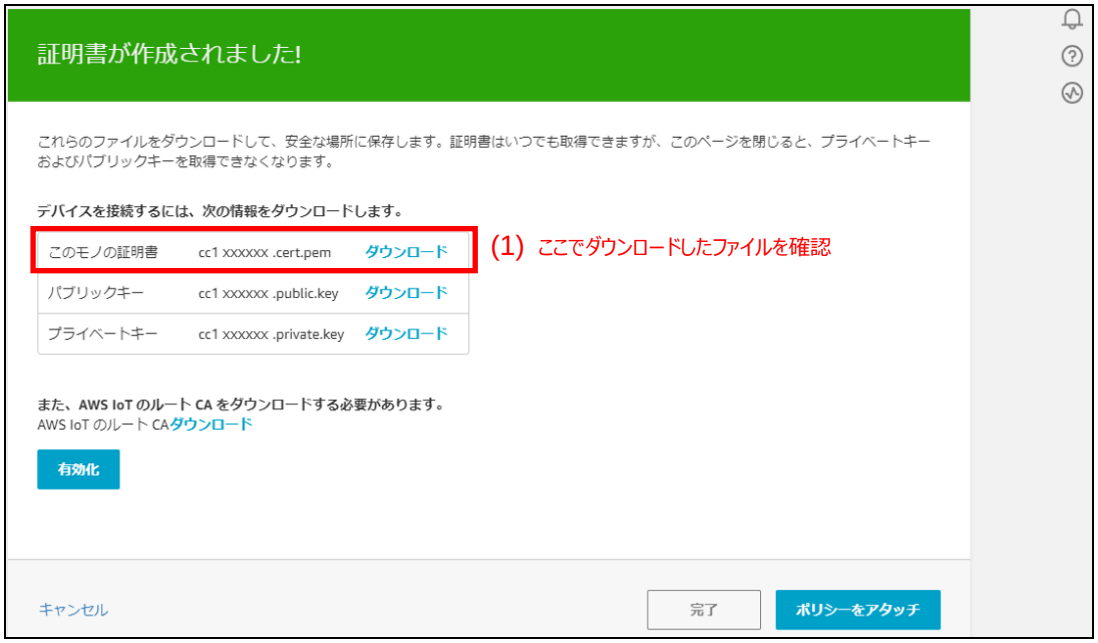


図 5-19 モノの名前の設定

5.2.2　AWS IoT 認証情報の設定

1. 5.1.2の証明書作成でダウンロードした **このモノの証明書** の内容を
   <root>/demos/common/include/aws_clientcredential_keys.h の
   clientcredentialCLIENT_CERTIFICATE_PEM[] に入力します。
   下図のように各行に “ と ¥n”¥ を追加します。



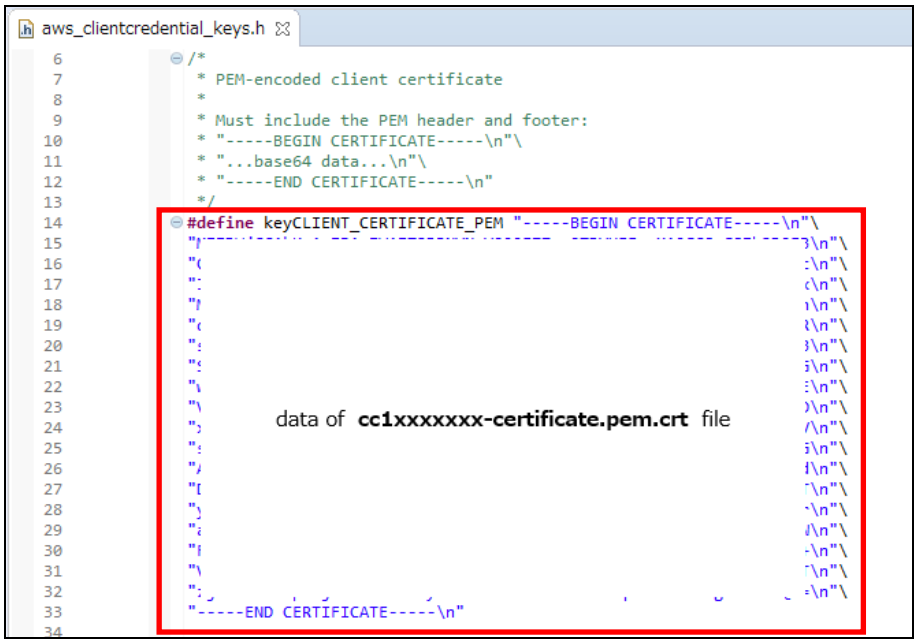
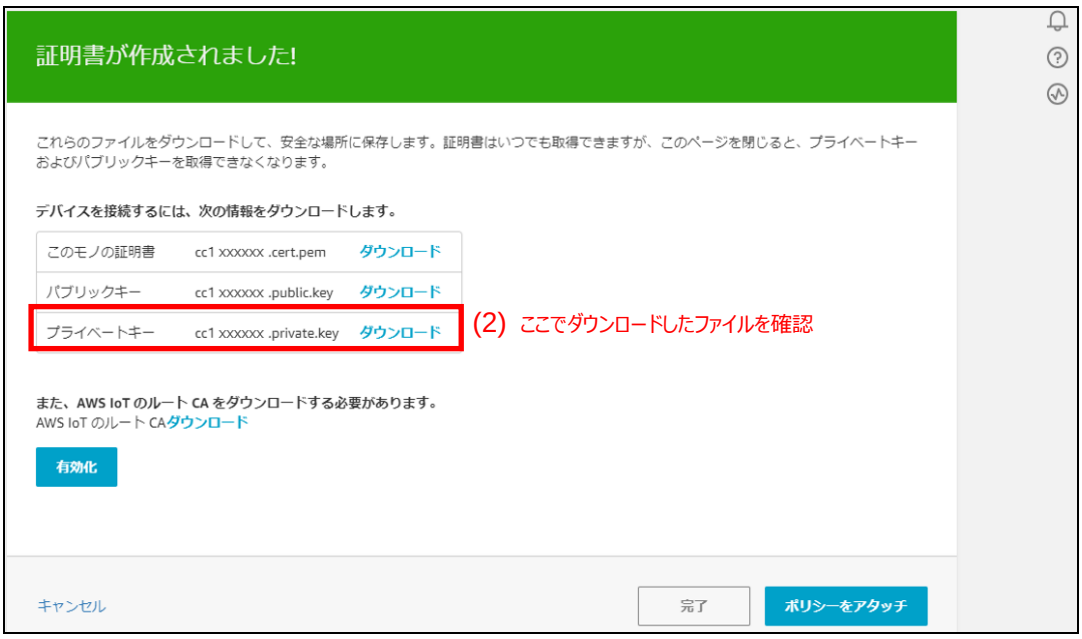


図 5-20 証明書情報入力

2. 5.1.2の証明書作成でダウンロードした **プライベートキー** の内容を <root>/demos/common/include/aws_clientcredential_keys.h の keyCLIENT_PRIVATE_KEY_PEM[] に入力します。



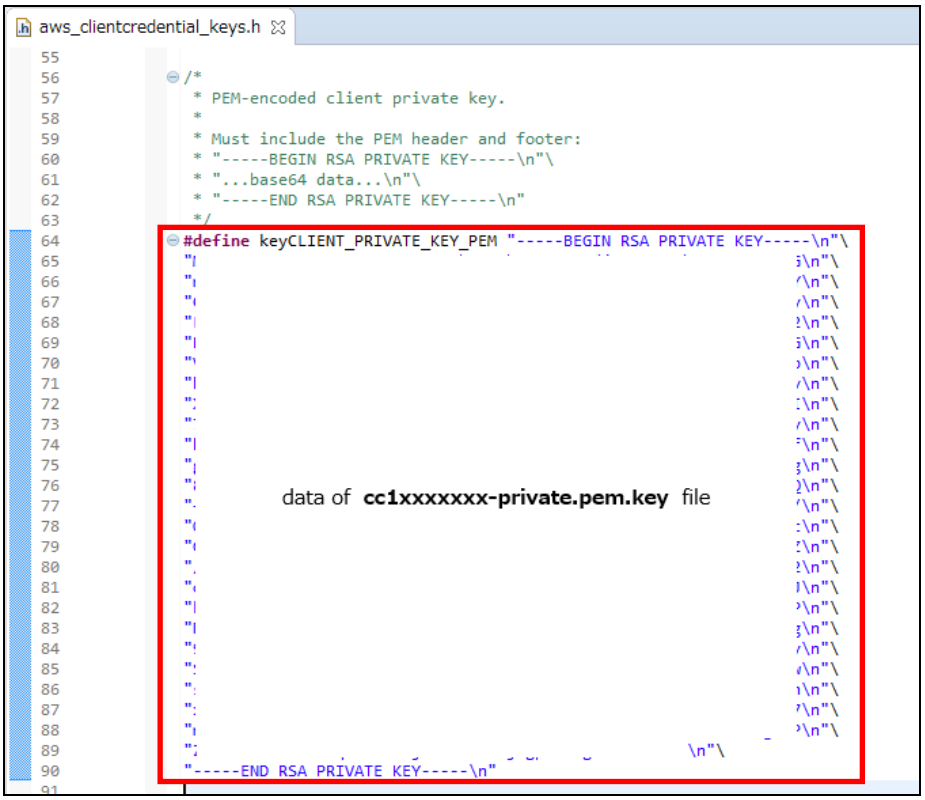


図 5-21 プライベートキー情報入力

5.3　RZ/A2M 側の設定

5.3.1　デバッグメッセージの抑止

FreeRTOSIPConfig.h の#define ipconfigHAS_DEBUG_PRINTF を 1 から 0 に変更します。

```
#define ipconfigHAS_DEBUG_PRINTF    0
```

5.3.2　MAC アドレスの設定

FreeRTOSConfig.h の#define configMAC_ADDR0 から #define configMAC_ADDR5 をボードの MAC アドレスに変更します。

```
#define configMAC_ADDR0                 0x74
#define configMAC_ADDR1                 0x90
#define configMAC_ADDR2                 0x50
#define configMAC_ADDR3                 0x00
#define configMAC_ADDR4                 0x79
#define configMAC_ADDR5                 0x03
```

5.3.3　IP アドレスの設定

- DHCP サーバーを用いる場合

FreeRTOSIPConfig.h の#define ipconfigUSE_DHCP を 0 から 1 に変更します。

```
#define ipconfigUSE_DHCP                    1
```

- DHCP サーバーを用いない場合

FreeRTOSIPConfig.h の#define ipconfigUSE_DHCP が 0 であることを確認します。

```
#define ipconfigUSE_DHCP                    0
```

FreeRTOSConfig.h の#define configIP_ADDR0 から #define configIP_ADDR3 をボードの IP アドレスに変更します。

```
#define configIP_ADDR0                      192
#define configIP_ADDR1                      168
#define configIP_ADDR2                      128
#define configIP_ADDR3                      2
```

FreeRTOSConfig.h の#define configGATEWAY_ADDR0 から #define configGATEWAY_ADDR3 をゲートウェイサーバーの IP アドレスに変更します。

```
#define configGATEWAY_ADDR0                 192
#define configGATEWAY_ADDR1                 168
#define configGATEWAY_ADDR2                 128
#define configGATEWAY_ADDR3                 100
```

FreeRTOSConfig.h の#define configDNS_ADDR0 から #define configDNS_ADDR3 を DNS サーバーの IP アドレスに変更します。

```
#define configDNS_SERVER_ADDR0              192
#define configDNS_SERVER_ADDR1              168
#define configDNS_SERVER_ADDR2              128
#define configDNS_SERVER_ADDR3              200
```

FreeRTOSConfig.h の#define configNET_MASK0 から #define configNET_MASK3 をサブネットマスクの値に変更します。

```
#define configNET_MASK0                     255
#define configNET_MASK1                     255
#define configNET_MASK2                     255
#define configNET_MASK3                     0
```

5.3.4　接続用テストプログラムの切り替え

demos¥common¥demo_runner¥aws_demo_runner.c を vStartMQTTEchoDemo() が有効になるように変更します。vStartTCPEchoClientTasks_SingleTasks()は無効にしてください。

```
/* Demo declarations. */
/* extern void vStartDeviceDefenderDemo( void ); */
/* extern void vStartGreenGrassDiscoveryTask( void ); */
extern void vStartMQTTEchoDemo( void );
/* extern void vStartOTAUpdateDemoTask( void ); */
/* extern void vStartShadowDemoTasks( void ); */
/* extern void vStartSimpleTCPServerTasks( void ); */
/* extern void vStartSubpubDemoTasks( void ); */
/* extern void vStartTCPEchoClientTasks_SeparateTasks( void ); */
/* extern void vStartTCPEchoClientTasks_SingleTasks( void ); */

/*-----------------------------------------------------------*/

/**
 * @brief Runs demos in the system.
 */
void DEMO_RUNNER_RunDemos( void )
{
    /* vStartDeviceDefenderDemo(); */
    /* vStartGreenGrassDiscoveryTask(); */
    vStartMQTTEchoDemo();
    /* vStartOTAUpdateDemoTask(); */
    /* vStartShadowDemoTasks(); */
    /* vStartSimpleTCPServerTasks(); */
    /* vStartSubpubDemoTasks(); */
    /* vStartTCPEchoClientTasks_SeparateTasks(); */
    /* vStartTCPEchoClientTasks_SingleTasks(); */
}
```

5.3.5　インポートとビルドとダウンロード

プロジェクトを e2 studio にインポートし、ビルド後、評価ボードにダウンロードします。

5.4 MQTTEcho デモの実行

5.4.1 AWS IoT テストの設定

1. AWS コンソールにサインインして AWS IoT を開きます。
2. 左側にある **テスト** を選択します。
3. **トピックのサブスクリプション** に Amazon FreeRTOS プロジェクト内 <root>/demos/common/mqtt/aws_hello_world.c の #define echoTOPIC_NAME に記述された文字列を入力します。
4. **MQTT ペイロード表示** の **ペイロードを文字列として表示（より正確）**を選択します。
5. **トピックへのサブスクライブ** を選択します。

図 5-22 MQTTEcho テスト設定 画面

6. RZ/A2M の MQTTEcho デモプログラムを実行します。
7. MQTT メッセージを確認します。

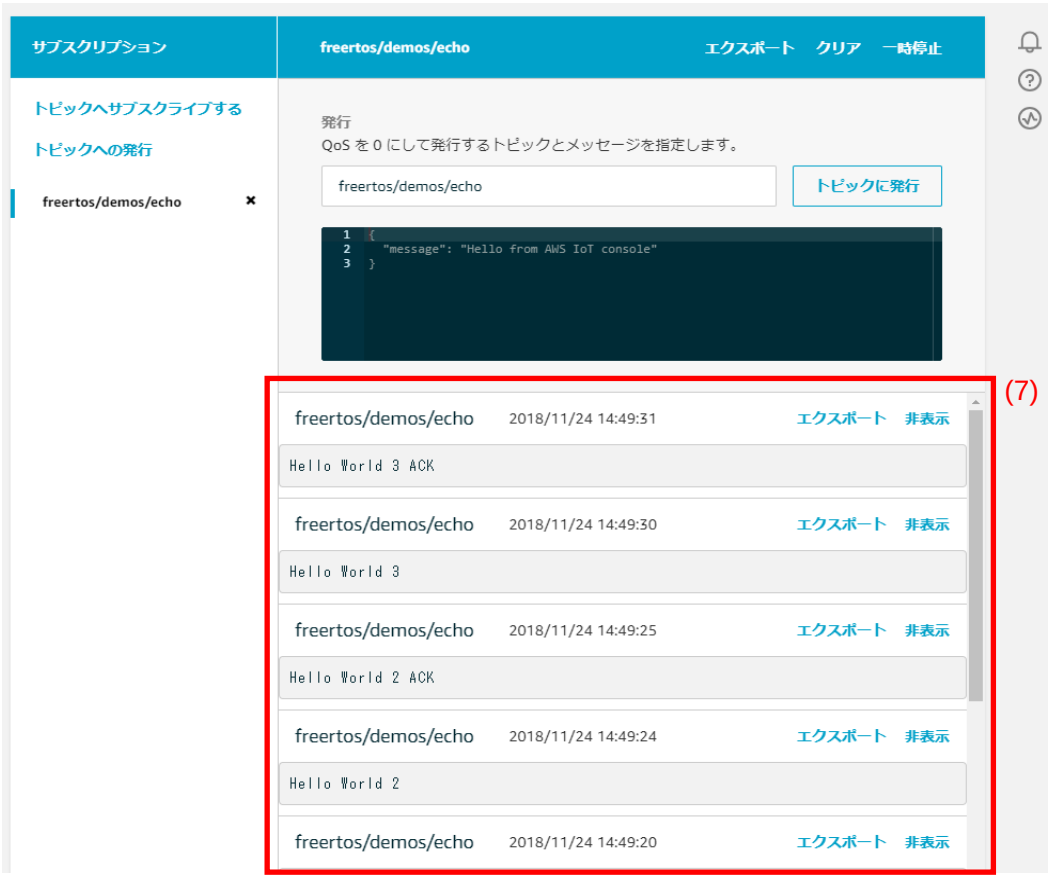


図 5-23 MQTTEcho テスト結果 画面

6. 参考ドキュメント

ユーザーズマニュアル：ハードウェア

RZ/A2M グループ ユーザーズマニュアル ハードウェア編
（最新版をルネサス エレクトロニクスホームページから入手してください。）

RTK7921053C00000BE（RZ/A2M CPU ボード）ユーザーズマニュアル
（最新版をルネサス エレクトロニクスホームページから入手してください。）

RTK79210XXB00000BE（RZ/A2M SUB ボード）ユーザーズマニュアル
（最新版をルネサス エレクトロニクスホームページから入手してください。）

Arm Architecture Reference Manual ARMv7-A and ARMv7-R edition Issue C
（最新版を Arm ホームページから入手してください。）

Arm CortexTM-A9 Technical Reference Manual Revision: r4p1
（最新版を Arm ホームページから入手してください。）

Arm Generic Interrupt Controller Architecture Specification - Architecture version2.0
（最新版を Arm ホームページから入手してください。）

Arm CoreLinkTM Level 2 Cache Controller L2C-310 Technical Reference Manual Revision: r3p3
（最新版を Arm ホームページから入手してください。）

テクニカルアップデート／テクニカルニュース
（最新の情報をルネサス エレクトロニクスホームページから入手してください。）

ユーザーズマニュアル：統合開発

統合開発環境 e2 studio のユーザーズマニュアルは、ルネサス エレクトロニクスホームページから入手してください。
（最新版をルネサス エレクトロニクスホームページから入手してください。）

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2018.12.28	—	新規発行
1.01	2019. 4.15	4	動作確認した e^2 studio のバージョンを 7.4.0 に更新
1.10	2019. 5.17	3	表 2-1 動作確認条件 コンパイラオプション”-mthumb-interwork”を削除
		24	5. Amazon Web Serviceへの接続 を追加

製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本ドキュメントおよびテクニカルアップデートを参照してください。

1. 静電気対策
   CMOS 製品の取り扱いの際は静電気防止を心がけてください。CMOS 製品は強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、当社が出荷梱包に使用している導電性のトレーやマガジンケース、導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。また、CMOS 製品を実装したボードについても同様の扱いをしてください。
2. 電源投入時の処置
   電源投入時は、製品の状態は不定です。電源投入時には、LSI の内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。
3. 電源オフ時における入力信号
   当該製品の電源がオフ状態のときに、入力信号や入出力プルアップ電源を入れないでください。入力信号や入出力プルアップ電源からの電流注入により、誤動作を引き起こしたり、異常電流が流れ内部素子を劣化させたりする場合があります。資料中に「電源オフ時における入力信号」についての記載のある製品は、その内容を守ってください。
4. 未使用端子の処理
   未使用端子は、「未使用端子の処理」に従って処理してください。CMOS 製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI 周辺のノイズが印加され、LSI 内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。
5. クロックについて
   リセット時は、クロックが安定した後、リセットを解除してください。プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。
6. 入力端子の印加波形
   入力ノイズや反射波による波形歪みは誤動作の原因になりますので注意してください。CMOS 製品の入力がノイズなどに起因して、V_{IL}（Max.）から V_{IH}（Min.）までの領域にとどまるような場合は、誤動作を引き起こす恐れがあります。入力レベルが固定の場合はもちろん、V_{IL}（Max.）から V_{IH}（Min.）までの領域を通過する遷移期間中にチャタリングノイズなどが入らないように使用してください。
7. リザーブアドレス（予約領域）のアクセス禁止
   リザーブアドレス（予約領域）のアクセスを禁止します。アドレス領域には、将来の拡張機能用に割り付けられている リザーブアドレス（予約領域）があります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。
8. 製品間の相違について
   型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。同じグループのマイコンでも型名が違うと、フラッシュメモリ、レイアウトパターンの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ幅射量などが異なる場合があります。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器・システムの設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因して生じた損害（お客様または第三者いずれに生じた損害も含みます。以下同じです。）に関し、当社は、一切その責任を負いません。
2. 当社製品、本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
3. 当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
4. 当社製品を、全部または一部を問わず、改造、改変、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、改変、複製、リバースエンジニアリング等により生じた損害に関し、当社は、一切その責任を負いません。
5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。
   標準水準：　コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット等
   高品質水準：輸送機器（自動車、電車、船舶等）、交通制御（信号）、大規模通信機器、金融端末基幹システム、各種安全制御装置等
   当社製品は、データシート等により高信頼性、Harsh environment 向け製品と定義しているものを除き、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じても、当社は一切その責任を負いません。
6. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合および事故につきましては、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は、データシート等において高信頼性、Harsh environment 向け製品と定義しているものを除き、耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
8. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。かかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
9. 当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
10. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものといたします。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。
12. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。

注 1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社が直接的、間接的に支配する会社をいいます。

注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

(Rev.4.0-1　2017.11)

本社所在地

〒135-0061 東京都江東区豊洲 3-2-24（豊洲フォレシア）

www.renesas.com

お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。

www.renesas.com/contact/

商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。すべての商標および登録商標は、それぞれの所有者に帰属します。