RX ファミリ

無線モジュール（Wi-Fi/Bluetooth）を使用したデータ送信（温度センサ）

要旨

本アプリケーションノートでは、RX マイコンと無線モジュールを組み合わせて無線通信を行うサンプルプログラムについて説明します。RX マイコンに無線モジュールを接続することで、容易に無線（Wi-Fi 注 1/Bluetooth® 注 2）通信を行うことが出来ます。本アプリケーションノートでは、以下のボードを使用したサンプルプログラムを用意しています。

ボード名称	ボード呼称	型番
Renesas Solution Starter Kit for RX23E-A	RSSKRX23E-A	RTK0ESXB10C00001BJ
低消費電力 Bluetooth® Pmod™ボード	DA14531 Pmod™ Board	US159-DA14531EVZ
超低消費電力 Wi-Fi Pmod™ボード	DA16600 Pmod™ Board	US159-DA16600EVZ

本アプリケーションノートでは、「RX23E-A グループ　熱電対を使用した温度計測例」アプリケーションノート（R01AN4747）をベースに温度計測を行い、DA14531 Pmod™ Board の Bluetooth®機能、または、DA16600 Pmod™ Board の Wi-Fi 機能を用いて、計測した温度を無線通信する方法を説明します。

さらに、Wi-Fi 経由でインターネットを介し、クラウドサービスの一つである Amazon Web Service（AWS）注 3 に接続する方法についても説明します。

注 1　Wi-Fi は Wi-Fi Alliance の商標登録です。
注 2　Bluetooth®のワードマークおよびロゴは Bluetooth SIG, Inc が所有する登録商標であり、ルネサス エレクトロニクス株式会社はこれらのマークをライセンスに基づいて使用しています。その他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。
注 3　Amazon Web Service、AWS は Amazon.com,Inc.または関連会社の商標登録です。

各サンプルプログラムにおけるボードの組み合わせと通信対象を以下に示します。

プロジェクト名	ボード組み合わせ	データ送信先	通信方法
r01an6677_rx23ea_ble	RSSKRX23E-A＋DA14531 Pmod™ Board	スマートフォン	Bluetooth®
r01an6677_rx23ea_wifi	RSSKRX23E-A＋DA16600 Pmod™ Board	PC	Wi-Fi
r01an6677_rx23ea_aws	RSSKRX23E-A＋DA16600 Pmod™ Board	AWS	Wi-Fi

動作確認デバイス

RX23E-A グループ

本アプリケーションノートはその他の RX マイコンへ適用できます。そのマイコンの仕様に合わせて変更し、十分評価の上、ご使用ください。

本アプリケーションノートでは、ボード（RSSKRX23E-A）の加工が必要となります。各サンプルプログラムの動作準備（3.1.5、3.2.5、3.3.5　ハードウェアの準備）を参照してください。

DA14531 Pmod™ Board、および DA16600 Pmod™ Board は、RSSKRX23E-A に同梱していません。別途購入いただく必要があります。

目次

1. 概要

本アプリケーションノートは、「RX23E-A グループ　熱電対を使用した温度計測例」アプリケーションノート（R01AN4747）をベースに温度計測を行い、DA14531 Pmod™ Board の Bluetooth®機能、または、DA16600 Pmod™ Board の Wi-Fi 機能を用いて、計測した温度データを無線通信によりインタフェース端末へ送信する方法を説明します。

サンプルプログラムは、Bluetooth®の場合、Wi-Fi の場合、AWS の場合の 3 つを準備しています。

1.1 Bluetooth®の場合

Bluetooth®の場合は、RSSKRX23E-A と DA14531 Pmod™ Board を接続し、スマートフォンと通信します。Bluetooth®通信のデータは、スマートフォンのアプリで確認します。スマートフォンのアプリの名前は Renesas の SmartConsole です。

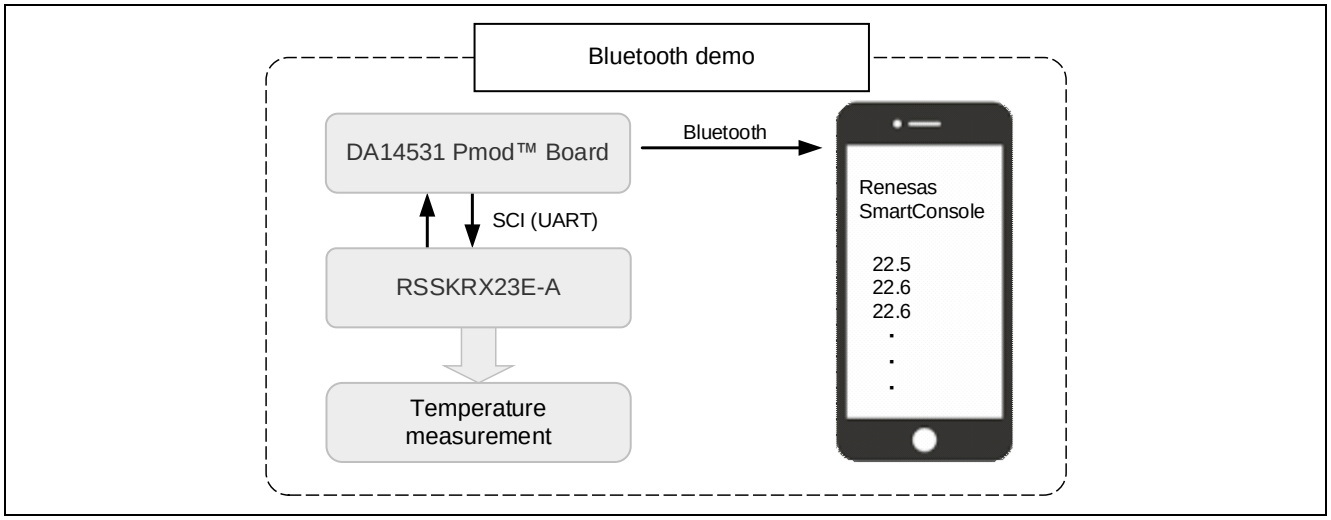

図 1-1　Bluetooth®デモ構成図

1.2 Wi-Fi の場合

Wi-Fi の場合は、RSSKRX23E-A と DA16600 Pmod™ Board を接続し、PC と通信します。Wi-Fi 通信のデータは、Tera Term で確認します。

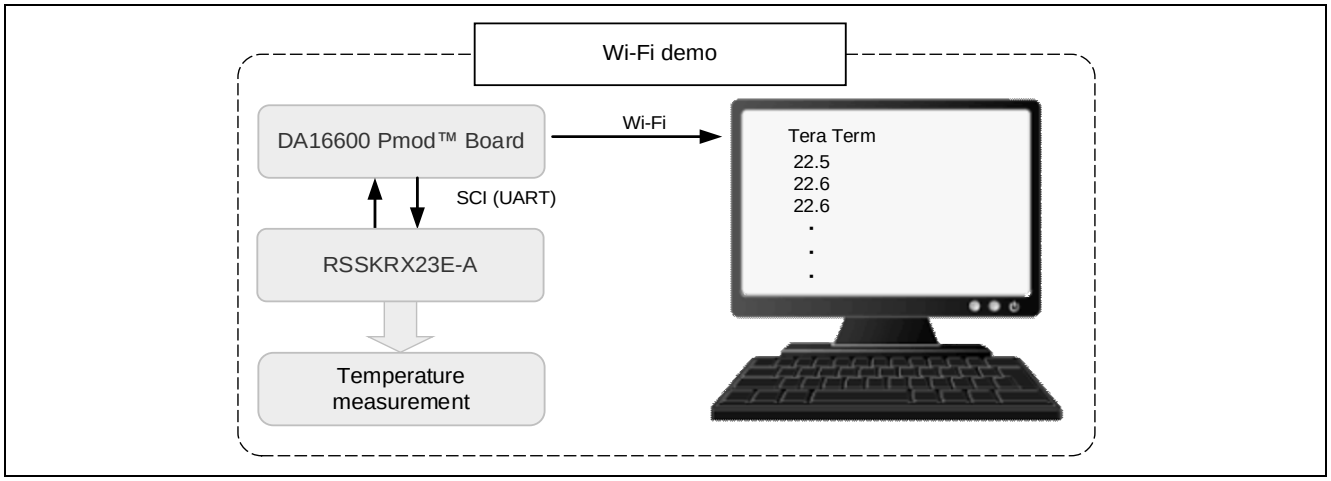

図 1-2　Wi-Fi デモ構成図

1.3　AWS の場合

AWS の場合は、お客様ご自身のアカウントを作成する必要があります。アカウントの作成方法については、「3.3.6 AWS 準備」参照してください。

AWS デモでは、RSSKRX23E-A と DA16600 Pmod™ Board を接続し、AWS の AWS IoT Core サービスを使用して MQTT プロトコルで通信します。AWS の通信データは、AWS コンソールで確認します。

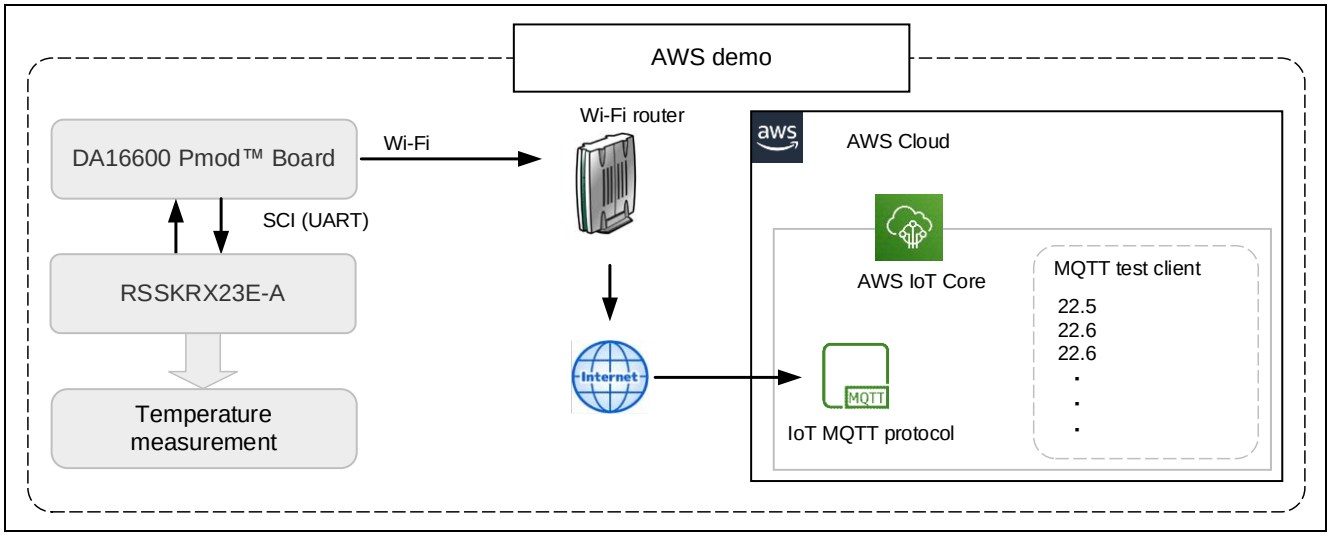

図 1-3　AWS デモ構成図

2. 動作確認条件

本サンプルプログラムは、下記の条件で動作を確認しています。

表 2-1　マイコン動作確認条件（RX23E-A）

項目	内容
使用マイコン	R5F523E6ADFL（RX23E-A　グループ）
CPU 最大動作周波数	32MHz
bit 数	32bit
パッケージ / ピン数	LFQFP / 48 ピン
ROM	256K バイト
RAM	32K バイト
動作電圧	3.3V

表 2-2　使用ツール

項目	内容
総合開発環境	ルネサスエレクトロニクス e^2 studio Version 2023.07
C コンパイラ	ルネサスエレクトロニクス C/C++ Compiler Package for RX Family V3.05.00
	コンパイラオプション 統合開発環境のデフォルト設定が適用されます。
スマート・コンフィグレータ	V2.18.0
ボードサポートパッケージ（r_bsp）	V7.41
エンディアン	リトルエンディアン
動作モード	シングルチップモード
プロセッサモード	スーパバイザモード
エミュレータ	E2 エミュレータ Lite
使用ボード	RSSKRX23E-A ボード（RTK0ESXB10C00001BJ）
通信ソフト	Tera Term (Version 4.106)
OS	なし

表 2-3　動作確認条件（DA16600 Pmod™ Board）

項目	内容
使用ボード	US159-DA16600EVZ
ファームウェア	DA 16600 v3.2.8.0 • Wi-Fi への接続や、AWS への TLS 通信などは、本ファームウェアで実行 • バージョンが異なる場合は「5.1.2 DA16600 Pmod™ Board のファームウェアバージョンを更新する」を参照してください

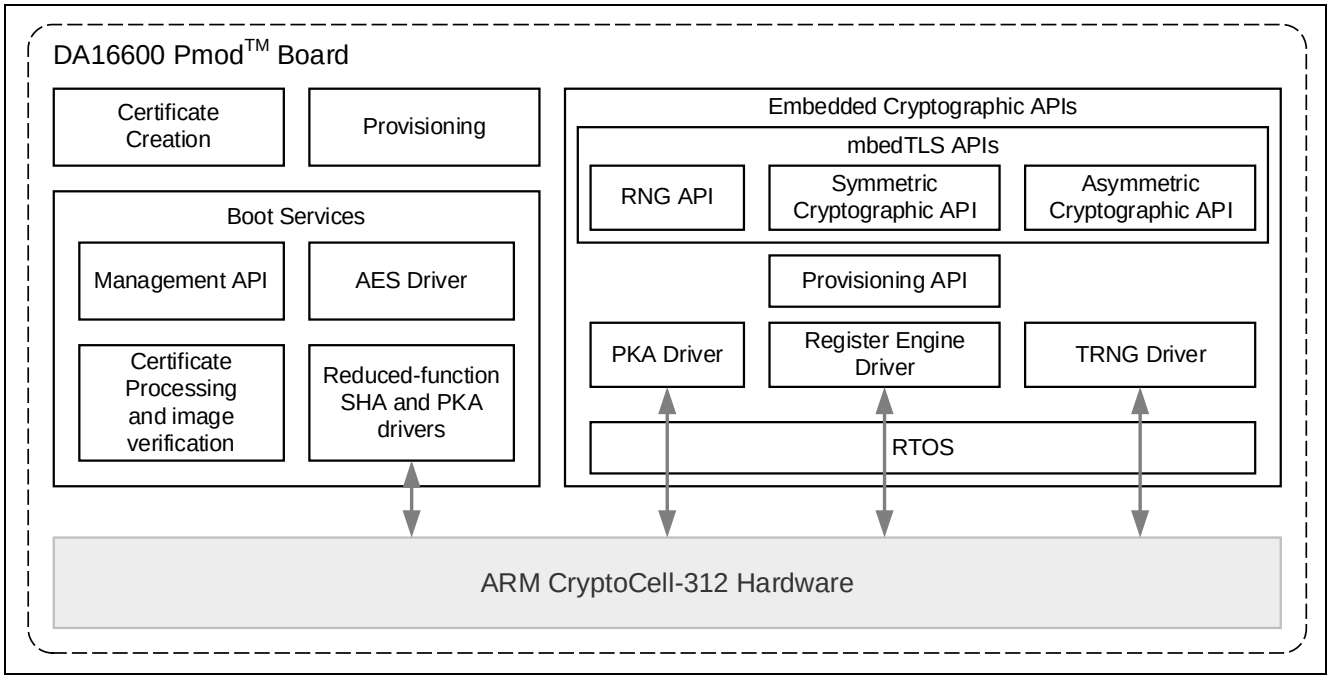

図 2-1　DA16600 ソフトウェアスタック

表 2-4　動作確認条件（DA14531 Pmod™ Board ）

項目	内容
使用ボード	US159-DA14531EVZ
ファームウェア	非公開 • Bluetooth®への Advertise 通信などは、本ファームウェアで実行

3. サンプルプログラム

本アプリケーションノートは、以下のサンプルプログラムを用意しています。これらのサンプルログラムは e² studio で動作を確認しています。

表 3-1　サンプルプログラム

プロジェクト名	内容	参照
r01an6677_rx23ea_ble	DA14531 Pmod™ Board を接続し、Bluetooth®デモを行う	3.1
r01an6677_rx23ea_wifi	DA16600 Pmod™ Board を接続し、Wi-Fi デモを行う	3.2
r01an6677_rx23ea_aws	DA16600 Pmod™ Board を接続し、AWS デモを行う	3.3

3.1　Bluetooth®デモプログラム（r01an6677_rx23ea_ble）

Bluetooth®デモを行うために、RSSKRX23E-A と DA14531 Pmod™ Board を接続します。

Bluetooth®デモを行うプロジェクトは、r01an6677_rx23ea_ble です。本プロジェクトを実行するためには、ハードウェアを改造する必要があります。プロジェクトを実行する場合は、「3.1.5 ハードウェアの準備」に進んでください。

3.1.1　システム構成

本サンプルプログラムのシステム構成を以下に示します。

RSSKRX23E-A Board の接続については、「RX23E-A グループ 熱電対を使用した温度計測例」アプリケーションノート（R01AN4747）の Page4 図 4-1 を参照してください。

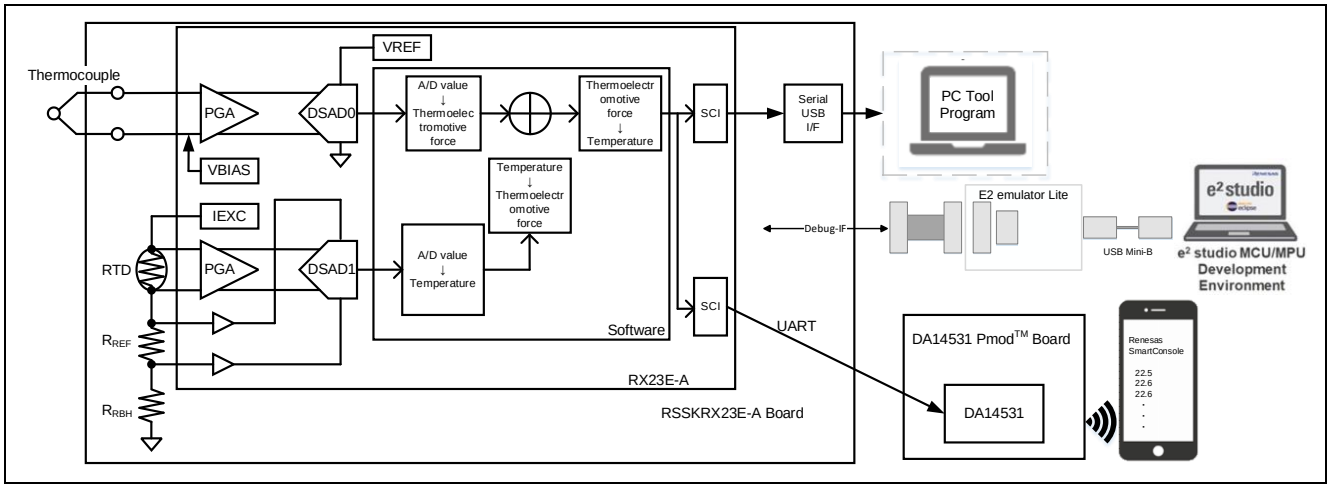

図 3-1　Bluetooth®デモのシステム構成

注：PC ツールは、本デモでは必要ありませんが、必要な場合は使用できます。

3.1.2　ソフトウェア構成

本サンプルプログラムのソフトウェア構成を以下に示します。RSSKRX23E-A Board の青色の部分は流用元のサンプルプログラムから変更がない部分です。Bluetooth®の制御は、全て DA14531 Pmod™ Board が行います。DA14531 Pmod™ Board のソフトウェア構成は公開していません。

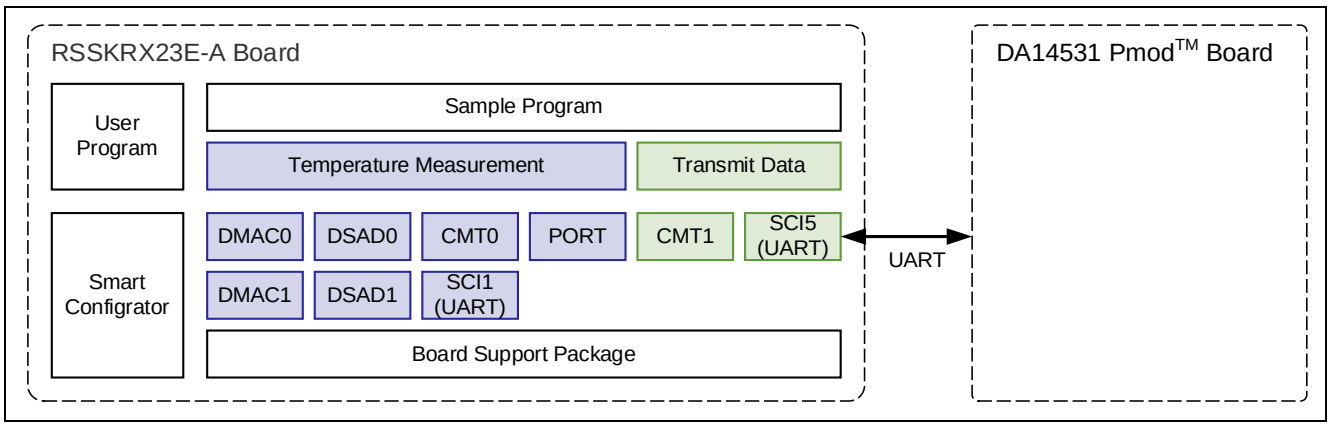

図 3-2　Bluetooth®デモのソフトウェア構成

3.1.3　概略フローチャート

本サンプルプログラムの概略フローチャートを以下に示します。

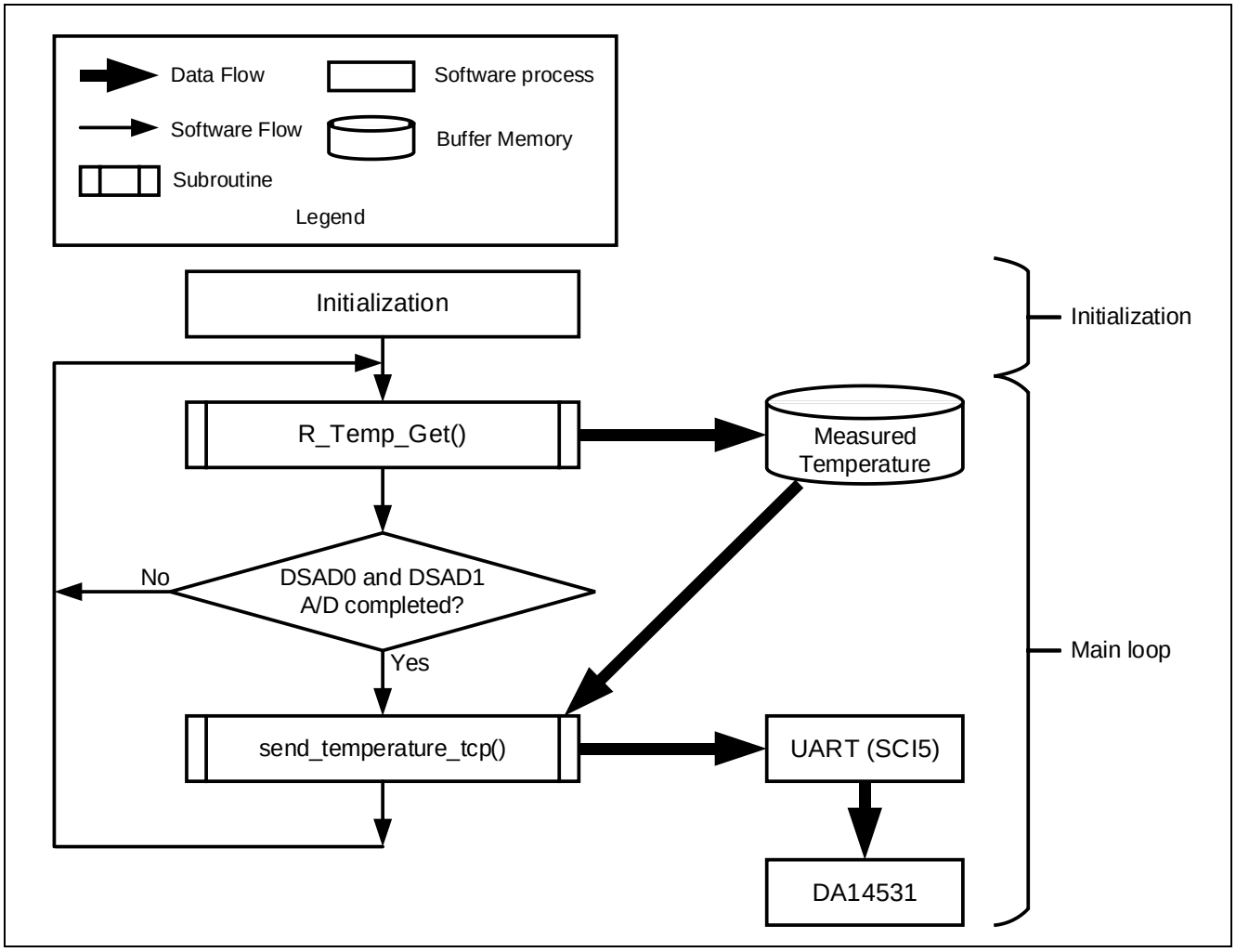

図 3-3　概略フローチャート

3.1.4　サンプルプログラムの構成

3.1.4.1　使用端子一覧

本サンプルプログラムの RX23E-A の使用端子一覧を以下に示します。

表 3-2　使用端子一覧

端子名	入出力	用途
PH2	出力	LED1 点灯制御
P26/TXD1	出力	UART1 送信端子
P30/RXD1	入力	UART1 受信端子
P31/CTS1#	入力	CTS 信号入力端子
AIN11	入力	熱電対+側入力端子
AIN10	入力	熱電対-側入力端子
AIN9	出力	RTD 励起電流出力端子
AIN7	入力	RTD +側入力端子
AIN6	入力	RTD -側入力端子
AIN5/REF1P	入力	RTD 測定 DSAD+側基準電圧
AIN4/REF1N	入力	RTD 測定 DSAD-側基準電圧
PH1/TXD5 注 1	出力	DA14531 Pmod™ Board の TXD に接続
PH0/RXD5 注 1	入力	DA14531 Pmod™ Board の RXD に接続
VCC 注 1	―	DA14531 Pmod™ Board に 3.3V を供給
VSS 注 1	―	DA14531 Pmod™ Board の VSS に接続

注 1.　ベースとする「RX23E-A グループ 熱電対を使用した温度計測例」アプリケーションノート（R01AN4747）から追加した端子

3.1.4.2　使用する周辺機能

本サンプルプログラムで使用する周辺機能を以下に示します。

表 3-3　使用する周辺機能一覧

周辺機能	用途	追加
AFE、DSAD0、DSAD1	熱電対、RTD の駆動（AFE）、熱電対の A/D 変換（DSAD0）、RTD の A/D 変換（DSAD1）	―
SCI1	PC ツールプログラムとの UART 通信	―
DMAC0	SCI1 の受信完了割り込みをトリガにデータ転送	―
DMAC3	SCI1 のバッファ空き割り込みをトリガにデータ転送	―
CMT0	SCI1 の通信タイムアウト検出	―
PH2	LED1 点灯制御	―
SCI5 注 1	DA14531 Pmod™ Board との UART 通信	yes
CMT1 注 1	温度データ送信のインターバル制御	yes

注 1.　ベースとする「RX23E-A グループ 熱電対を使用した温度計測例」アプリケーションノート（R01AN4747）から追加した周辺機能

3.1.4.3　周辺機能の設定

本サンプルプログラムで使用している周辺機能の設定はスマート・コンフィグレータのコード生成機能を用いています。スマート・コンフィグレータの設定条件を以下に示します。ベースとする「RX23E-A グループ 熱電対を使用した温度計測例」アプリケーションノート（R01AN4747）から追加した周辺機能を記載します。

表 3-4　SCI5 の設定

項目	設定
シリアル通信方式	調歩同期式
スタートビット検出設定	RXD5 端子の Low レベル
データ・ビット長	8 ビット
パリティ設定	禁止
ストップビット設定	1 ビット
データ転送方向設定	LSB ファースト
転送速度設定	• 転送クロック：内部クロック • ビットレート：115200bps • ビットレートモジュレーション機能有効 • SCK5 端子機能：SCK5 を使用しない
ノイズフィルタ設定	ノイズフィルタ無効
ハードウェアフロー制御設定	ハードウェアフロー制御設定：禁止
データ処理設定	送信データ処理：割り込みサービスルーチンで処理 受信データ処理：割り込みサービスルーチンで処理
割り込み設定	受信エラー割り込み許可 優先順位：レベル 15
コールバック機能設定	使用しない
入出力端子	• 出力：TXD5 (PH1) • 入力：RXD5 (PH0)

表 3-5　CMT1 の設定

項目	設定
クロック設定	PCLKB/512
コンペアマッチ設定	• インターバル時間：10ms • コンペアマッチ割り込みを許可 (CMI1) • 優先順位：レベル 15 (割り込み禁止)

3.1.4.4　ファイル構成

本サンプルプログラムのファイル構成を以下に示します。

表 3-6　ファイル構成

フォルダ名、ファイル名	説明
src	プログラム格納用フォルダ
├ main.c 注 1	メイン処理
├ r_ring_buffer_control_api.c	リングバッファ制御プログラム
├ r_ring_buffer_control_api.h	リングバッファ制御 API 定義
├ r_sensor_common_api.c	テーブル検索、直線補間処理プログラム
├ r_sensor_common_api.h	テーブル検索、直線補間処理 API 定義
├ r_thermocouple_api.c	熱電対計測演算プログラム、温度対熱起電力テーブル
├ r_thermocouple_api.h	熱電対計測演算 API 定義
├ r_rtd_api.c	測温抵抗体計測演算プログラム、温度対抵抗値テーブル
├ r_rtd_api.h	測温抵抗体計測演算 API 定義
├ r_communication_control_api.c	通信制御プログラム
├ r_communication_control_api.h	通信制御 API 定義
├ string_func.c 注 1	AT コマンド制御プログラム
├ string_func.h 注 1	AT コマンド制御 API 定義
└ smc_gen	スマート・コンフィグレータ生成
├ Config_AFE	
├ Config_CMT0	
├ Config_CMT1 注 1	
├ Config_DMAC0	
├ Config_DMAC3	
├ Config_DSAD0	
├ Config_DSAD1	
├ Config_PORT	
├ Config_SCI1	
├ Config_SCI5 注 1	
├ general	
├ r_bsp	
├ r_config	
└ r_pincfg	

注 1.　ベースとする「RX23E-A グループ 熱電対を使用した温度計測例」アプリケーションノート（R01AN4747）から追加、変更を加えたファイル

3.1.4.5　変数一覧

本サンプルプログラムで使用する変数一覧を以下に示します。

ベースとする「RX23E-A グループ 熱電対を使用した温度計測例」アプリケーションノート（R01AN4747）から追加した変数を記載します。追加した変数以外は、R01AN4747 を参照下さい。

表 3-7　サンプルプログラムで使用する変数一覧

変数名	型	内容
g_temp	volatile float	温度データ
g_send_flg	volatile uint8_t	温度データ送信完了フラグ
g_rcv_buf	uint8_t	受信データを格納するバッファ

3.1.4.6　定数一覧

ベースとする「RX23E-A グループ 熱電対を使用した温度計測例」アプリケーションノート（R01AN4747）から追加した定数はありません。定数一覧は、R01AN4747 を参照下さい。

3.1.4.7　関数一覧

本サンプルプログラムで使用する関数一覧を以下に示します。

ベースとする「RX23E-A グループ 熱電対を使用した温度計測例」アプリケーションノート（R01AN4747）から追加、変更を加えた関数を記載します。

表 3-8　サンプルプログラムで使用する関数一覧

関数名	概要	
main	メイン処理	変更
send_temperature_ble	温度データを送信	追加

3.1.4.8　関数仕様

本サンプルプログラムの関数仕様を以下に示します。

[関数名]main

概要	メイン処理
ヘッダ	なし
宣言	void main (void)
説明	周辺機能を初期化します。熱電対を使用した温度計測、DA14531 Pmod™ Board の制御、温度データの送信を行います。
引数	なし
戻り値	なし
備考	なし

[関数名] send_temperature_ble

概要	温度データを送信
ヘッダ	string_func.h
宣言	void send_temperature_ble (void)
説明	送信バッファに送信用コマンドと温度データをセットし、DA14531 Pmod™ Board に送信します。
引数	なし
戻り値	なし
備考	なし

3.1.4.9　Bluetooth®デモのフローチャート

Bluetooth®デモの main の関数フローチャートを示します。

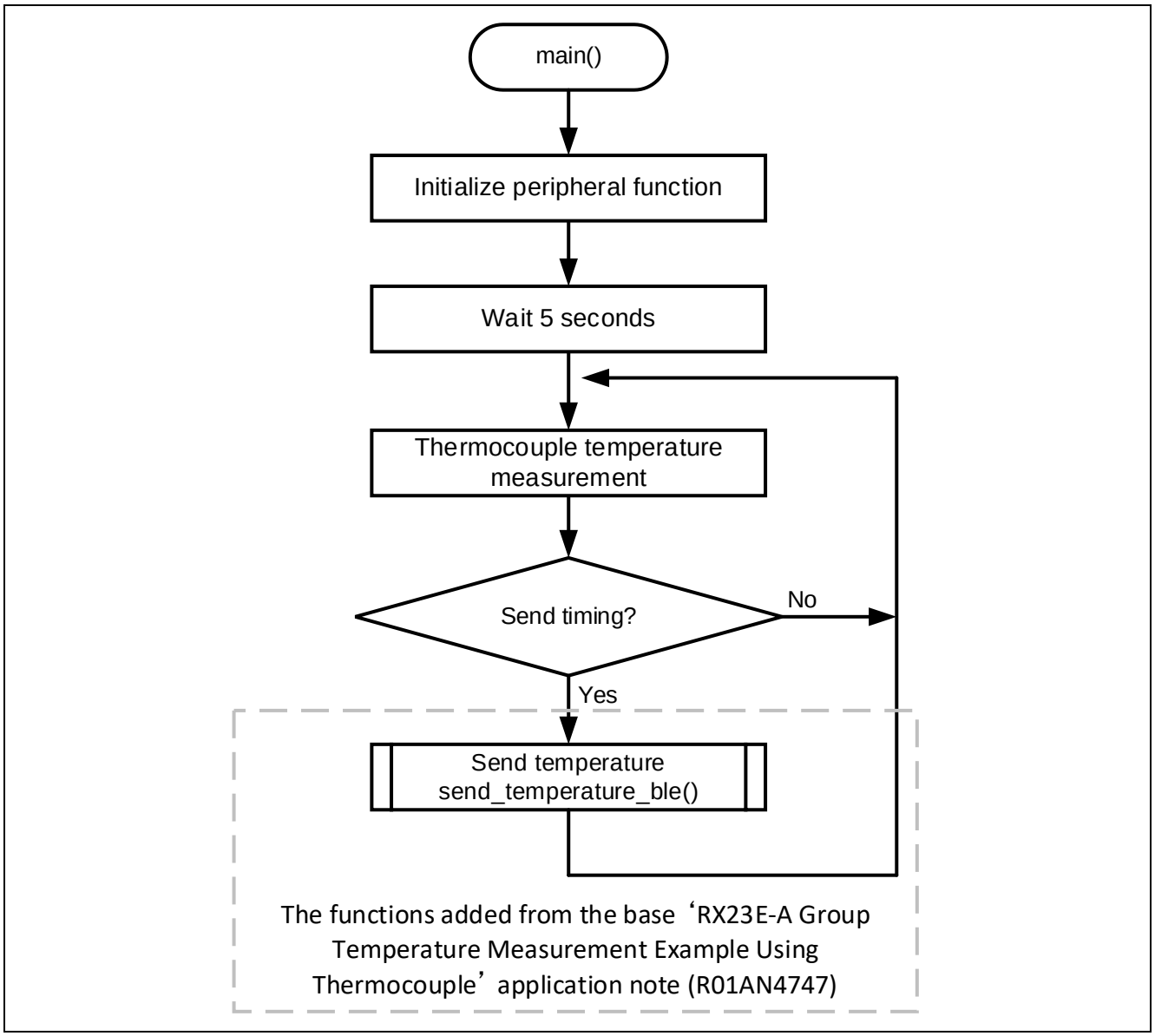

図 3-4　main 関数フローチャート

3.1.5　ハードウェアの準備

本アプリケーションノートでは、RSSKRX23E-A ボードの熱電対計測回路を使用します。使用方法の詳細は、「RSSKRX23E-A ユーザーズマニュアル」の「2.4 アナログ入力回路の使用方法」を参照してください。

RSSKRX23E-A と DA14531 Pmod™ Board を接続するために、RSSKRX23E-A ボードを改造する必要があります。

改造する端子一覧を以下に示します。ピン番号の詳細は、「RSSKRX23E-A ユーザーズマニュアル」を参照してください。

表 3-9　改造する端子一覧

ピン番号	MCU 端子番号	機能	入出力	説明
1	ー	VSS	出力	VSS 端子
2	ー	VCC	出力	VCC 端子 外部への電源供給に使用
3	23	PH1/TXD5	入出力	PH1/TXD5 端子
4	24	PH0/RXD5	入出力	PH0/RXD5 端子

3.1.5.1　チップ抵抗の除去

TXD5 と RXD5 を使用するために、チップ抵抗を除去する必要があります。除去するチップ抵抗は、R91 と R90 です。

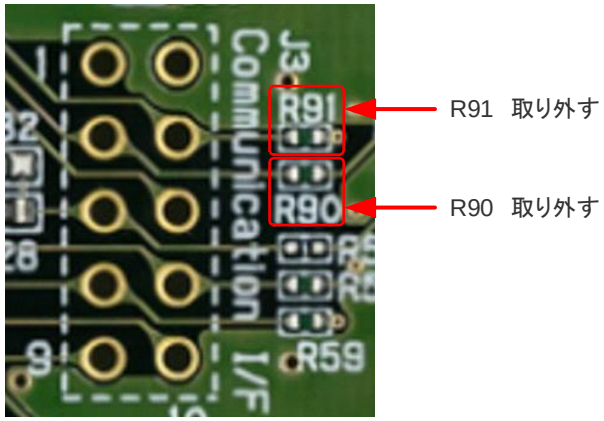

図 3-5　チップ抵抗の除去

3.1.5.2　ピンヘッダーの実装

VSS、VCC、TXD5、RXD5 を使用するために、ピンヘッダーを実装します。

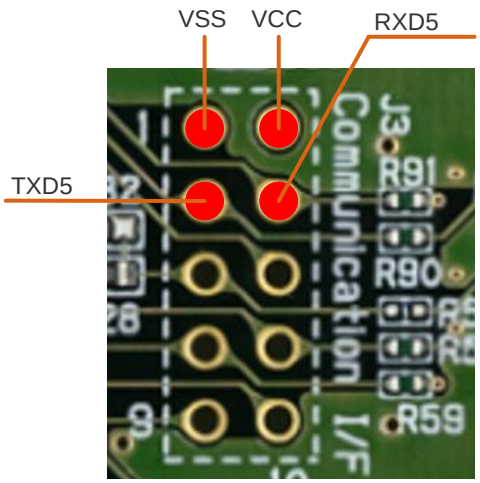

図 3-6　DA14531 Pmod™ Board との接続端子

3.1.5.3　RSSKRX23E-A と DA14531 Pmod™ Board を接続する

VSS、VCC、TXD5、RXD5 と DA14531 Pmod™ Board を以下のように接続します。

RSSKRX23E-A の TXD5 と DA14531 Pmod™ Board の TXD を接続することに注意してください。

表 3-10　接続表

RSSKRX23E-A		DA14531 Pmod™ Board		補足
ピン番号	端子名	ピン番号	信号	
―	―	1	CTS	OPEN
3	PH1/TXD5	2	TXD	
4	PH0/RXD5	3	RXD	
―	―	4	RTS	OPEN
1	VSS	5	GND	
2	VCC	6	VCC	
―	―	7	GPIO	OPEN
―	―	8	GPIO	OPEN
―	―	9	GPIO	OPEN
―	―	10	GPIO	OPEN
―	―	11	GND	OPEN
―	―	12	VCC	OPEN

3.1.6　ソフトウェアの準備

3.1.6.1　スマートフォンアプリの事前準備

Bluetooth®デモでは、RSSKRX23E-A が計測した温度データを、スマートフォンに送信します。このため、スマートフォンにアプリケーション（Renesas SmartConsole）をインストールする必要があります。スマートフォンのアプリケーションは、Apple App Store と Google Play Store の両方でグローバルに提供されています。スマートフォン用アプリケーションは、以下のリンクからダウンロードできます。

Renesas SmartConsole の使用方法については、http://lpccs-docs.renesas.com/UM-140-DA145x-CodeLess/smartconsole.html#を参照してください。

図 3-7　Link to Smartphone application

3.1.7　サンプルプログラム動作概要

サンプルプログラムの動作概要を示します。サンプルプログラムを動作させるための詳細手順は、「3.1.8 サンプルプログラム動作詳細」を参照してください。

(1) Start
サンプルプログラムをインポートして、実行します。

(2) Initialization
RSSKRX23E-A は自動的に初期化を行います。

(3) Advertise & Scan
電源が投入されると DA14531 Pmod™ Board は、自動的にアドバタイズを開始します。ここでスマートフォンのアプリケーションを使用して、DA14531 Pmod™ Board をスキャンします。

(4) Connection
スマートフォンから、DA14531 Pmod™ Board をスキャンし、DA14531 Pmod™ Board に接続します。

(5) Temperature data transmission
接続が完了すると、自動的に RSSKRX23E-A からスマートフォンに温度データが送信されます。

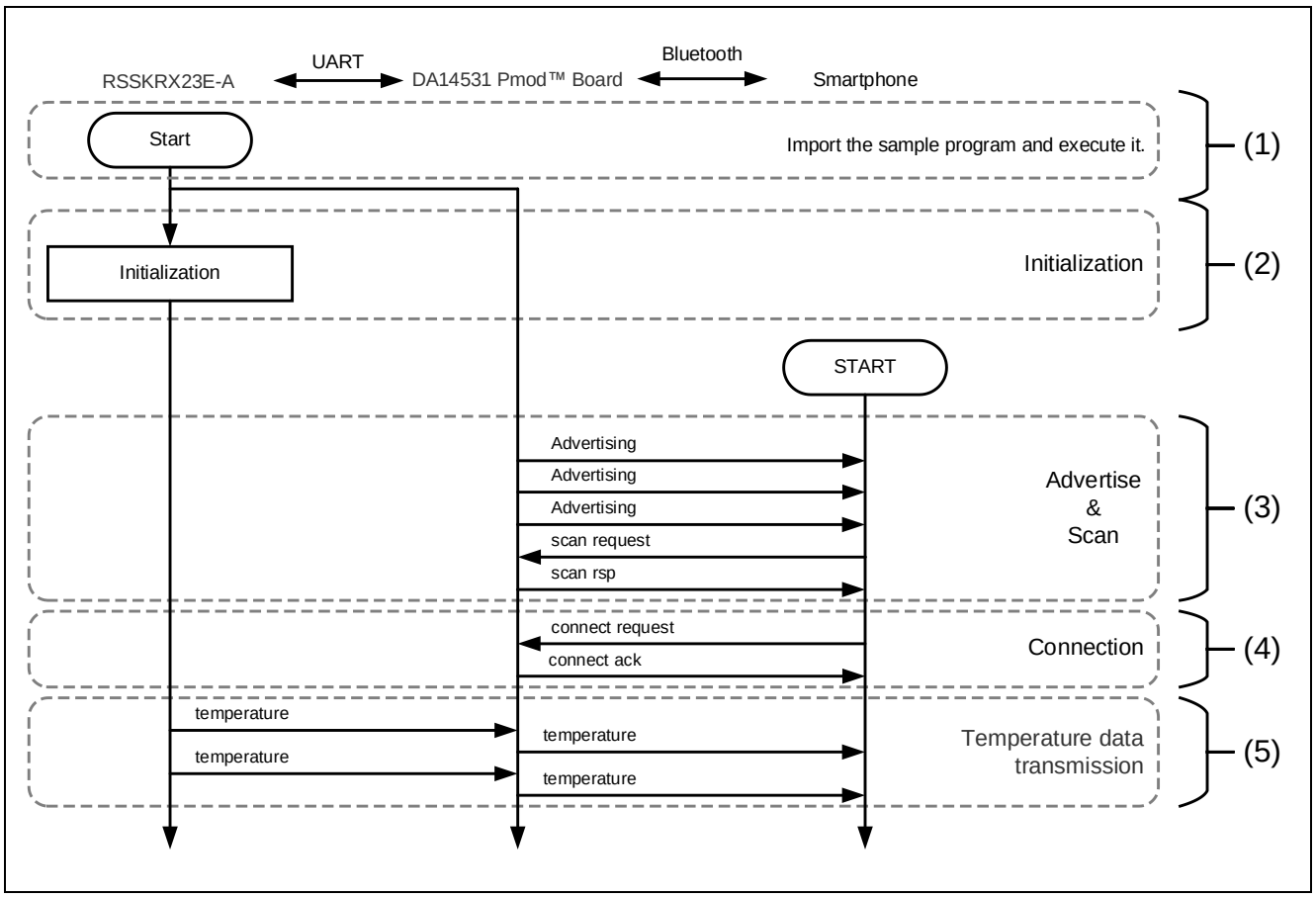

図 3-8　Bluetooth®デモの動作フロー

3.1.8　サンプルプログラム動作詳細

次の手順でデモを実行してください。

なお、DA14531 Pmod™ Board のファームウエアをアップデートした場合は、Renesas SmartConfig(app)を使用して、DA14531 Pmod™ Board のボーレートを 115200bps に再設定してください。

(1)　サンプルプログラムのインポートから実行までの手順
「4 プロジェクトの実行手順」に従って、プロジェクトを実行してください。
デバッガから電源が供給されると、LED2 が点灯します。

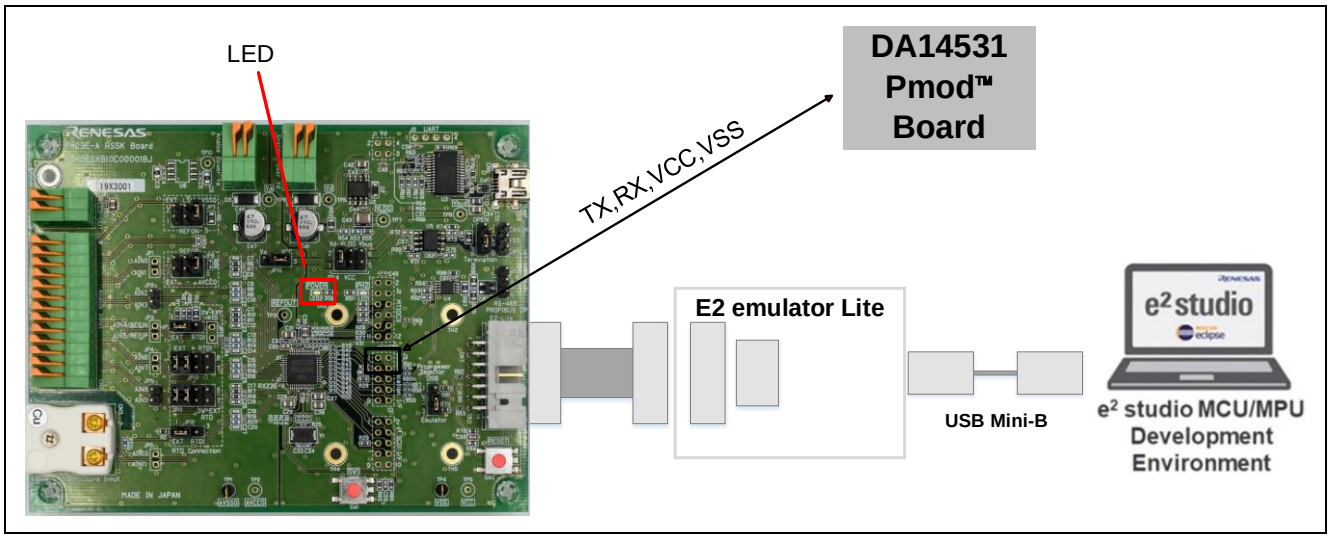

図 3-9　LED2（Power）の位置

(2)　Initialization
RSSKRX23E-A は、自動的に内部状態を初期化します。

(3)　Advertise & Scan
電源が投入されると DA14531 Pmod™ Board は、自動的にアドバタイズを開始します。ここでスマートフォンのアプリケーション（Renesas SmartConsole）を使用して、DA14531 Pmod™ Board をスキャンします。

(4) Connection
Renesas SmartConsole の画面で、DA14531 の Bluetooth 機器名(以下例では SPS_531)をタップすると、自動的に接続が行われます。

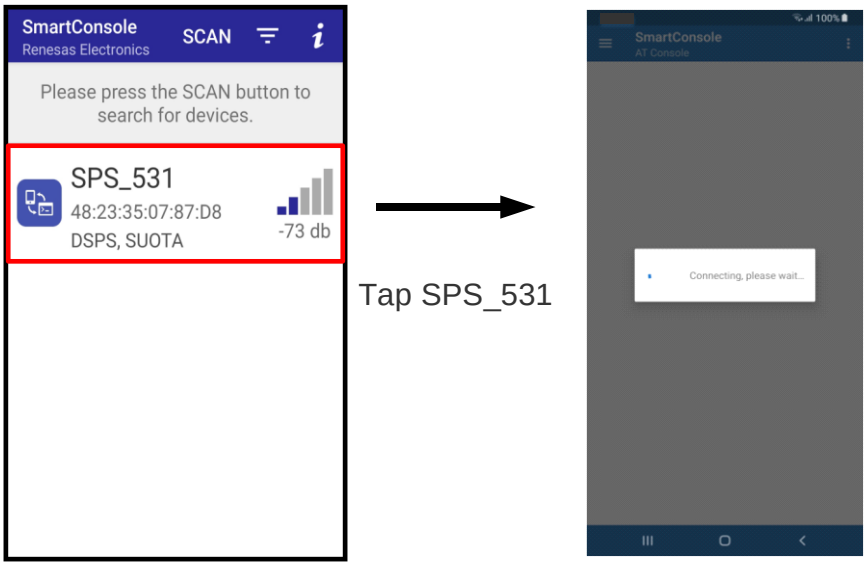

図 3-10　DA14531 Pmod™ Board のスキャン

(5) Temperature data transmission
接続が完了すると、自動的に RSSKRX23E-A からスマートフォンに温度データが送信されます。

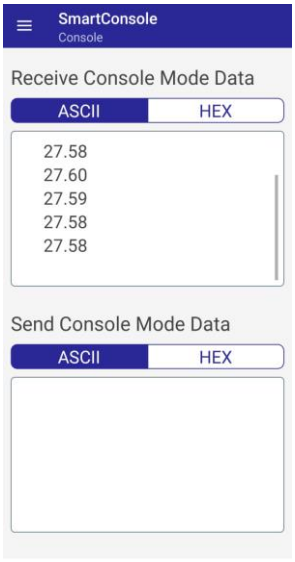

図 3-11　DA14531 Pmod™ Board のスキャン

3.2　Wi-Fi デモプログラム（r01an6677_rx23ea_wifi）

Wi-Fi デモを行うために、RSSKRX23E-A と DA16600 Pmod™ Board を接続します。

Wi-Fi デモを行うプロジェクトは、r01an6677_rx23ea_wifi です。本プロジェクトを実行するためには、ハードウェアを改造する必要があります。プロジェクトを実行する場合は、「3.2.5 ハードウェアの準備」に進んでください。本プロジェクトを実行するためには、ハードウェアを改造する必要があります。プロジェクトを実行する場合は、「3.2.5 ハードウェアの準備」に進んでください。

3.2.1　システム構成

本サンプルプログラムのシステム構成を以下に示します。

RSSKRX23E-A ボードの接続については、「RX23E-A グループ 熱電対を使用した温度計測例」アプリケーションノート（R01AN4747）の Page4 図 4-1 を参照してください。

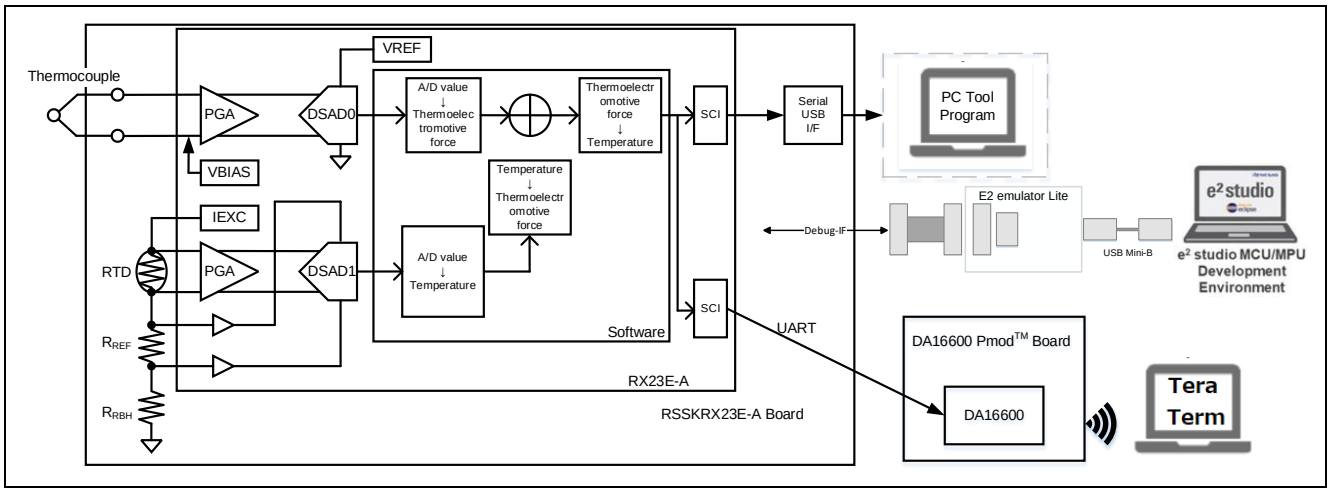

図 3-12　Wi-Fi デモのシステム構成

注：PC ツールは、本デモでは必要ありませんが、必要な場合は使用できます。

3.2.2　ソフトウェア構成

本サンプルプログラムのソフトウェア構成を示します。RSSKRX23E-A Board の青色の部分は流用元のサンプルプログラムから変更がない部分です。Wi-Fi や AWS との通信は、全て DA16600 Pmod™ Board が行います。

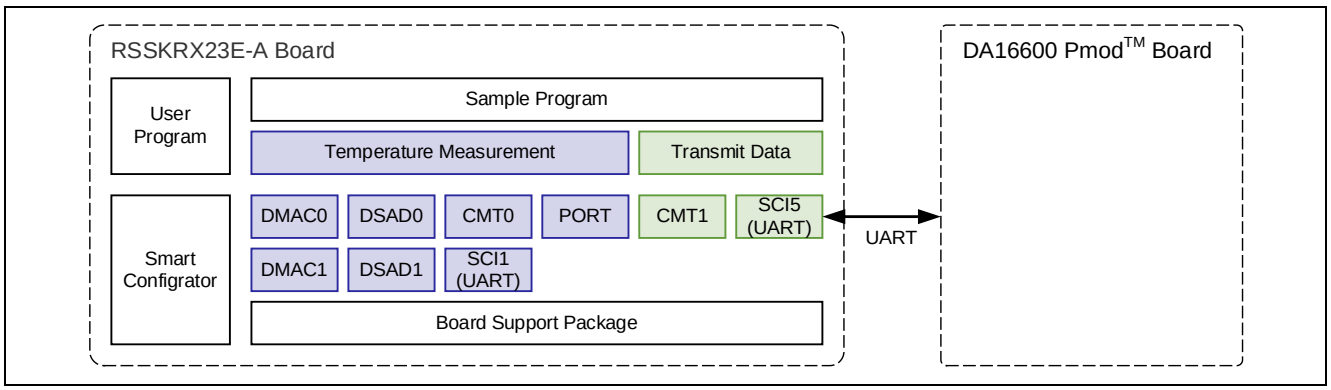

図 3-13　Wi-Fi デモのソフトウェア構成

3.2.3　概略フローチャート

本サンプルプログラムの概略フローチャートを以下に示します。

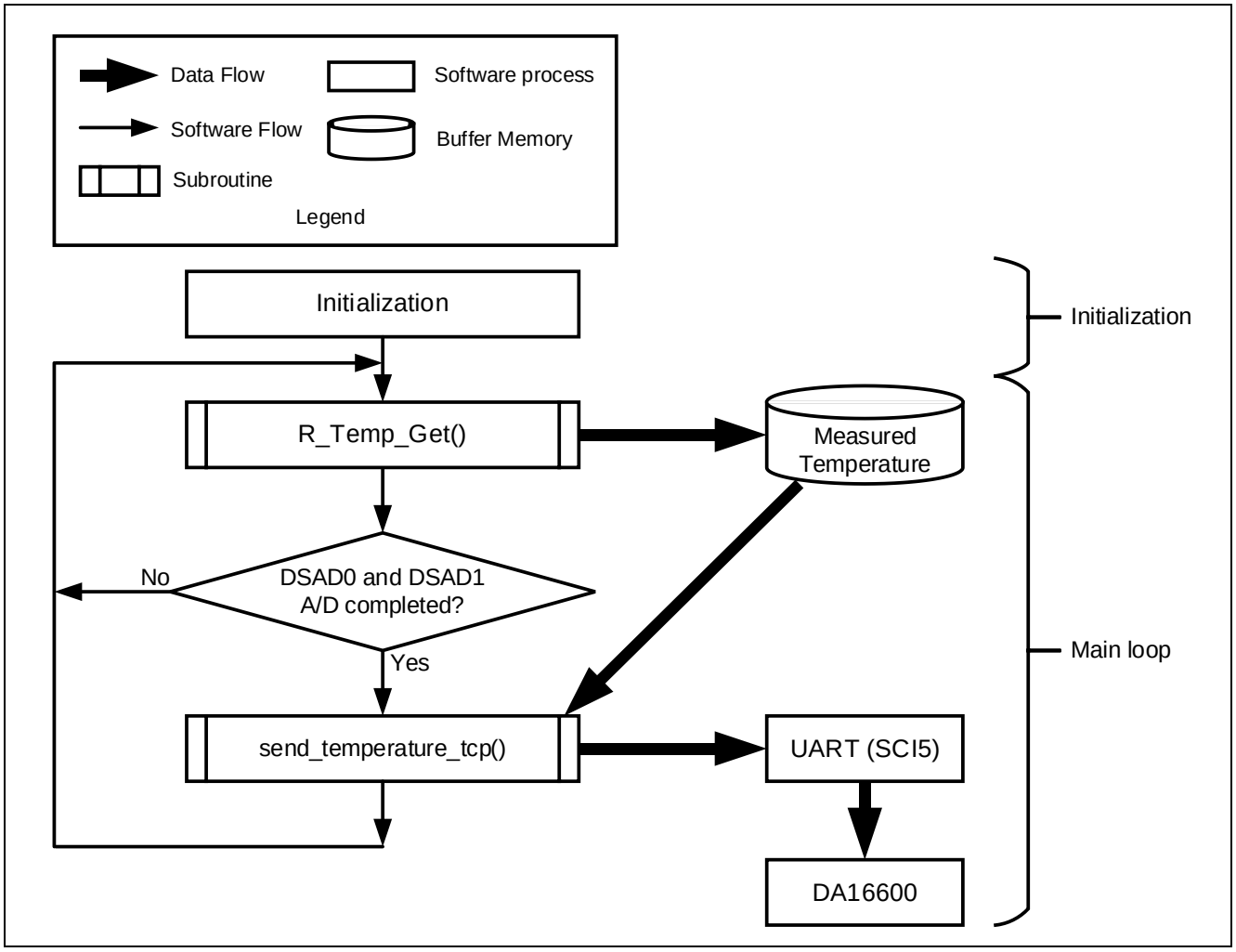

図 3-14　概略フローチャート

3.2.4　サンプルプログラムの構成

3.2.4.1　使用端子一覧

本サンプルプログラムの RX23E-A の使用端子一覧を以下に示します。

表 3-11　使用端子一覧

端子名	入出力	用途
PH2	出力	LED1 点灯制御
P26/TXD1	出力	UART1 送信端子
P30/RXD1	入力	UART1 受信端子
P31/CTS1#	入力	CTS 信号入力端子
AIN11	入力	熱電対+側入力端子
AIN10	入力	熱電対-側入力端子
AIN9	出力	RTD 励起電流出力端子
AIN7	入力	RTD +側入力端子
AIN6	入力	RTD -側入力端子
AIN5/REF1P	入力	RTD 測定 DSAD+側基準電圧
AIN4/REF1N	入力	RTD 測定 DSAD-側基準電圧
PH1/TXD5 注 1	出力	DA16600 Pmod™ Board の GPIOC7_TXD_HOST に接続
PH0/RXD5 注 1	入力	DA16600 Pmod™ Board の GPIOC6_RXD_HOST に接続
VCC 注 1	―	DA16600 Pmod™ Board に 3.3V を供給
VSS 注 1	―	DA16600 Pmod™ Board の VSS に接続

注 1.　ベースとする「RX23E-A グループ 熱電対を使用した温度計測例」アプリケーションノート（R01AN4747）から追加した端子

3.2.4.2　使用する周辺機能

本サンプルプログラムで使用する周辺機能を以下に示します。

表 3-12　使用する周辺機能一覧

周辺機能	用途	追加
AFE、DSAD0、DSAD1	熱電対、RTD の駆動(AFE)、熱電対の A/D 変換(DSAD0)、RTD の A/D 変換(DSAD1)	―
SCI1	PC ツールプログラムとの UART 通信	―
DMAC0	SCI1 の受信完了割り込みをトリガにデータ転送	―
DMAC3	SCI1 のバッファ空き割り込みをトリガにデータ転送	―
CMT0	SCI1 の通信タイムアウト検出	―
PH2	LED1 点灯制御	―
SCI5 注 1	DA16600 Pmod™ Board との UART 通信	yes
CMT1 注 1	温度データ送信のインターバル制御	yes

注 1.　ベースとする「RX23E-A グループ 熱電対を使用した温度計測例」アプリケーションノート（R01AN4747）から追加した周辺機能

3.2.4.3　周辺機能の設定

本サンプルプログラムで使用している周辺機能の設定はスマート・コンフィグレータのコード生成機能を用いています。スマート・コンフィグレータの設定条件を以下に示します。ベースとする「RX23E-A グループ 熱電対を使用した温度計測例」アプリケーションノート（R01AN4747）から追加した周辺機能を記載します。

表 3-13　SCI5 の設定

項目	設定
シリアル通信方式	調歩同期式
スタートビット検出設定	RXD5 端子の Low レベル
データ・ビット長	8 ビット
パリティ設定	禁止
ストップビット設定	1 ビット
データ転送方向設定	LSB ファースト
転送速度設定	• 転送クロック：内部クロック • ビットレート：115200bps • ビットレートモジュレーション機能有効 • SCK5 端子機能：SCK5 を使用しない
ノイズフィルタ設定	ノイズフィルタ無効
ハードウェアフロー制御設定	ハードウェアフロー制御設定：禁止
データ処理設定	送信データ処理：割り込みサービスルーチンで処理 受信データ処理：割り込みサービスルーチンで処理
割り込み設定	受信エラー割り込み許可 優先順位：レベル 15
コールバック機能設定	使用しない
入出力端子	• 出力：TXD5 (PH1) • 入力：RXD5 (PH0)

表 3-14　CMT1 の設定

項目	設定
クロック設定	PCLKB/512
コンペアマッチ設定	• インターバル時間：10ms • コンペアマッチ割り込みを許可 (CMI1) • 優先順位：レベル 15 (割り込み禁止)

3.2.4.4　ファイル構成

本サンプルプログラムのファイル構成を以下に示します。

表 3-15　ファイル構成

フォルダ名、ファイル名	説明
src	プログラム格納用フォルダ
├ main.c 注 1	メイン処理
├ r_ring_buffer_control_api.c	リングバッファ制御プログラム
├ r_ring_buffer_control_api.h	リングバッファ制御 API 定義
├ r_sensor_common_api.c	テーブル検索、直線補間処理プログラム
├ r_sensor_common_api.h	テーブル検索、直線補間処理 API 定義
├ r_thermocouple_api.c	熱電対計測演算プログラム、温度対熱起電力テーブル
├ r_thermocouple_api.h	熱電対計測演算 API 定義
├ r_rtd_api.c	測温抵抗体計測演算プログラム、温度対抵抗値テーブル
├ r_rtd_api.h	測温抵抗体計測演算 API 定義
├ r_communication_control_api.c	通信制御プログラム
├ r_communication_control_api.h	通信制御 API 定義
├ string_func.c 注 1	AT コマンド制御プログラム
├ string_func.h 注 1	AT コマンド制御 API 定義
└ smc_gen	スマート・コンフィグレータ生成
├ Config_AFE	
├ Config_CMT0	
├ Config_CMT1 注 1	
├ Config_DMAC0	
├ Config_DMAC3	
├ Config_DSAD0	
├ Config_DSAD1	
├ Config_PORT	
├ Config_SCI1	
├ Config_SCI5 注 1	
├ general	
├ r_bsp	
├ r_config	
└ r_pincfg	

注 1.　ベースとする「RX23E-A グループ 熱電対を使用した温度計測例」アプリケーションノート（R01AN4747）から追加、変更を加えたファイル

3.2.4.5　変数一覧

本サンプルプログラムで使用する変数一覧を以下に示します。

ベースとする「RX23E-A グループ 熱電対を使用した温度計測例」アプリケーションノート（R01AN4747）から追加した変数を記載します。追加した変数以外は、R01AN4747 を参照下さい。

表 3-16　サンプルプログラムで使用する変数一覧

変数名	型	内容
g_temp	volatile float	温度データ
g_rcv_end_flg	volatile uint8_t	DA16600 Pmod™ Board コマンド応答受信フラグ
g_send_flg	volatile uint8_t	温度データ送信完了フラグ
g_rcv_buf	uint8_t	受信データを格納するバッファ

3.2.4.6　定数一覧

ベースとする「RX23E-A グループ 熱電対を使用した温度計測例」アプリケーションノート（R01AN4747）から追加した定数はありません。定数一覧は、R01AN4747 を参照下さい。

3.2.4.7　関数一覧

本サンプルプログラムで使用する関数一覧を以下に示します。

ベースとする「RX23E-A グループ 熱電対を使用した温度計測例」アプリケーションノート（R01AN4747）から追加、変更を加えた関数を記載します。

表 3-17　サンプルプログラムで使用する関数一覧

関数名	概要	
main	メイン処理	変更
start_softap_mode	DA16600 Pmod™ Board を TCP 通信可能な状態に設定	追加
check_sci_rcv_end	DA16600 Pmod™ Board の応答を検出	追加
check_rcv_cmd	DA16600 Pmod™ Board から受信した内容をチェック	追加
reset_rcv_buf	受信バッファをクリア	追加
send_temperature_tcp	温度データを送信	追加

3.2.4.8　関数仕様

本サンプルプログラムの関数仕様を以下に示します。

[関数名]main

概要	メイン処理
ヘッダ	なし
宣言	void main (void)
説明	周辺機能を初期化します。熱電対を使用した温度計測、DA16600 Pmod™ Board の制御、温度データの送信を行います。
引数	なし
戻り値	なし
備考	なし

[関数名] start_softap_mode

概要	DA16600 Pmod™ Board を TCP 通信可能な状態に設定
ヘッダ	string_func.h
宣言	void start_softap_mode (void)
説明	DA16600 Pmod™ Board に AT コマンドを送信します。 DA16600 Pmod™ Board を soft-AP モードに設定し、TCP 通信可能な状態にします。
引数	なし
戻り値	なし
備考	なし

[関数名] check_sci_rcv_end

概要	DA16600 Pmod™ Board の応答を検出
ヘッダ	string_func.h
宣言	void check_sci_rcv_end (void)
説明	改行コードから、DA16600 Pmod™ Board の応答を検出します。
引数	なし
戻り値	なし
備考	なし

[関数名] check_rcv_cmd

概要	DA16600 Pmod™ Board から受信した内容をチェック
ヘッダ	string_func.h
宣言	void check_rcv_cmd (void)
説明	DA16600 Pmod™ Board から受信した内容をチェックします。
引数	なし
戻り値	なし
備考	なし

[関数名] reset_rcv_buf

概要	受信バッファをクリア
ヘッダ	string_func.h
宣言	void reset_rcv_buf (void)
説明	受信バッファをクリアします。
引数	なし
戻り値	なし
備考	なし

[関数名] send_temperature_tcp

概要	温度データを送信
ヘッダ	string_func.h
宣言	void send_temperature_tcp (void)
説明	送信バッファに送信用コマンドと温度データをセットし、DA16600 Pmod™ Board に送信します。
引数	なし
戻り値	なし
備考	なし

3.2.4.9　Wi-Fi デモのフローチャート

Wi-Fi デモの main 関数フローチャートを以下に示します。

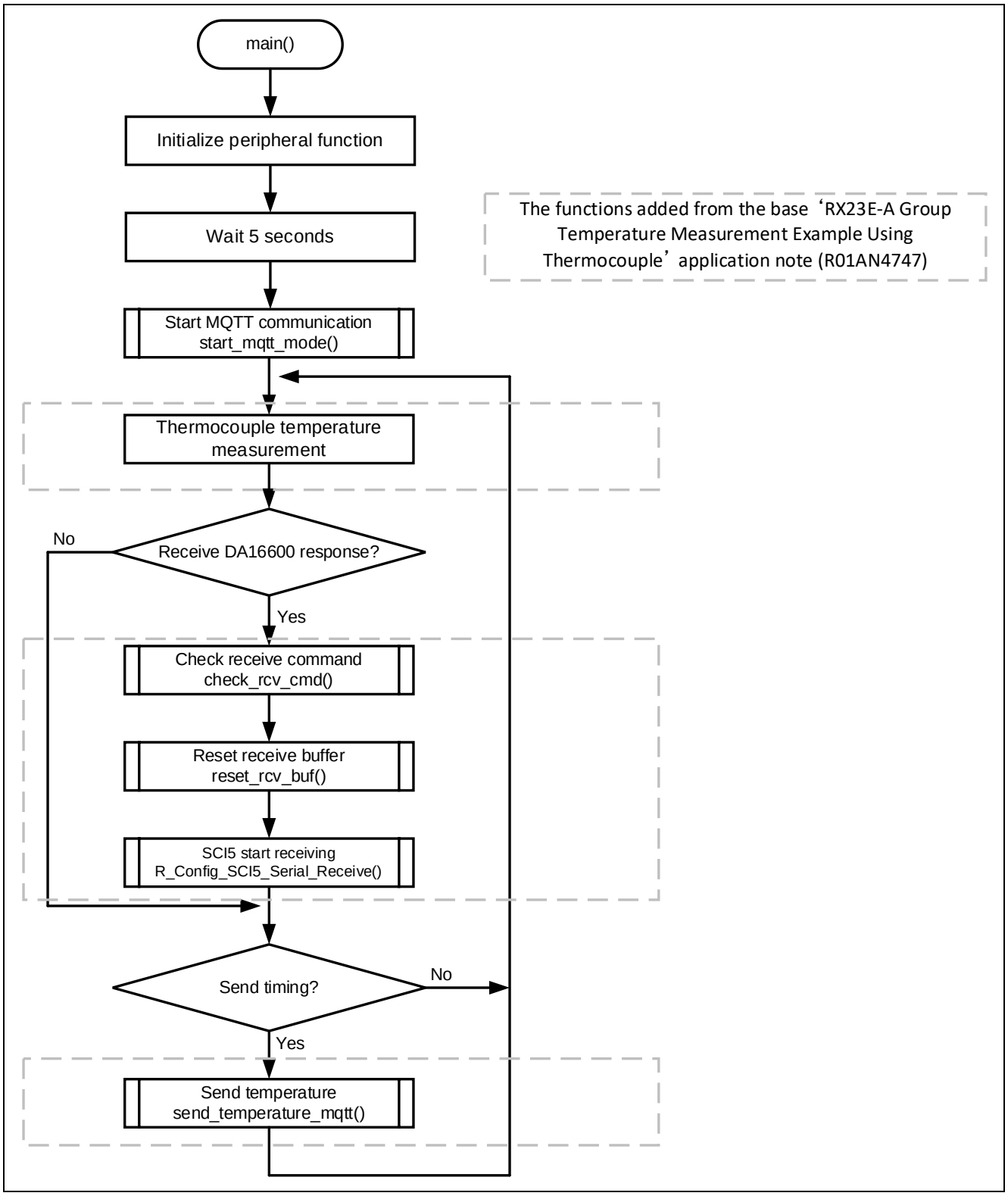

図 3-15　main 関数フローチャート

Wi-Fi デモの start_softap_mode 関数フローチャートを以下に示します。

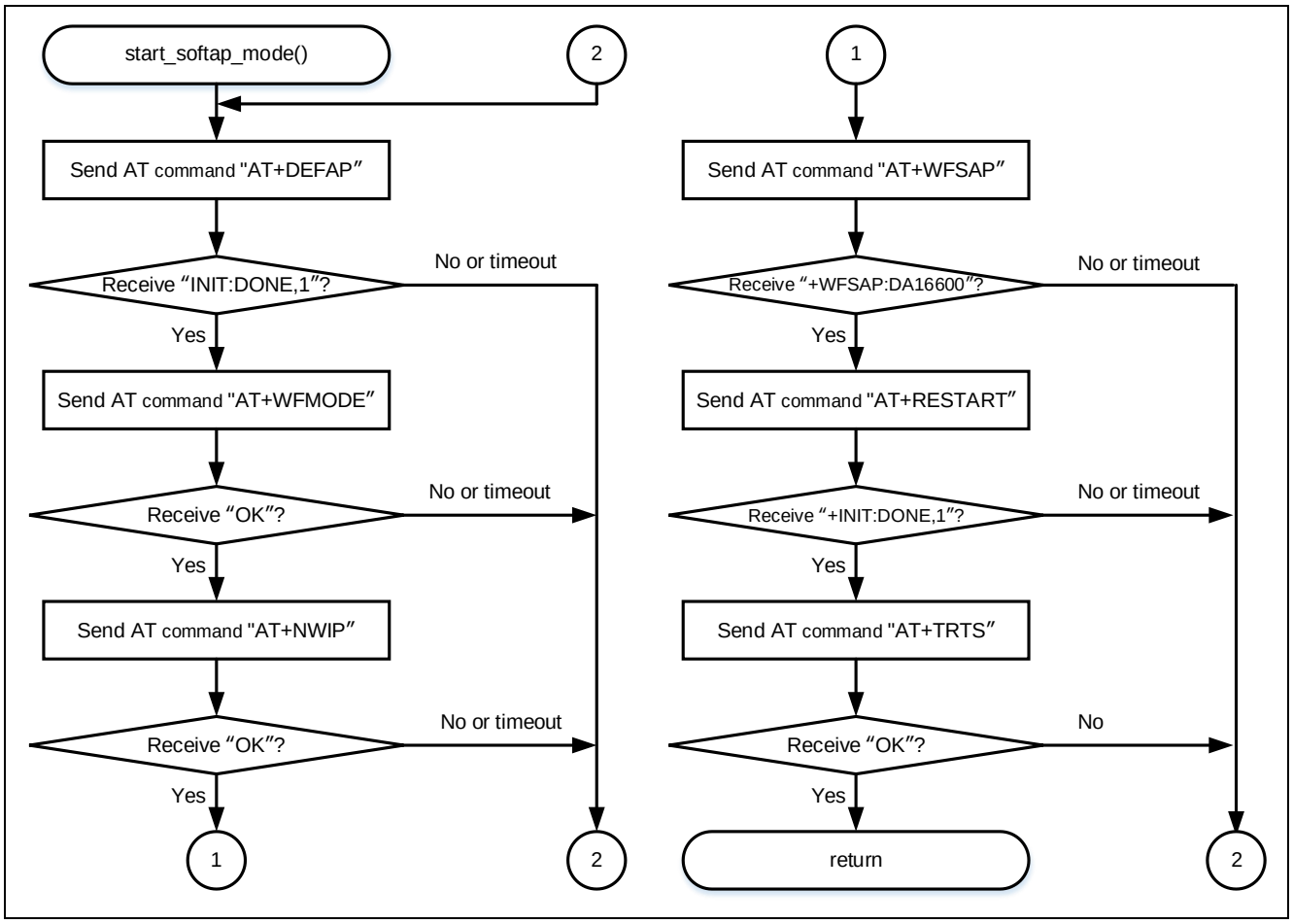

図 3-16　start_softap_mode 関数フローチャート

3.2.5　ハードウェアの準備

本アプリケーションノートでは、RSSKRX23E-A ボードの熱電対計測回路を使用します。使用方法の詳細は、「RSSKRX23E-A ユーザーズマニュアル」の「2.4 アナログ入力回路の使用方法」を参照してください。

RSSKRX23E-A と DA16600 Pmod™ Board を接続するために、RSSKRX23E-A ボードを改造する必要があります。

改造する端子一覧を以下に示します。ピン番号の詳細は、「RSSKRX23E-A ユーザーズマニュアル」を参照してください。

表 3-18　改造する端子一覧

ピン番号	MCU 端子番号	機能	入出力	説明
1	ー	VSS	出力	VSS 端子
2	ー	VCC	出力	VCC 端子 外部への電源供給に使用
3	23	PH1/TXD5	入出力	PH1/TXD5 端子
4	24	PH0/RXD5	入出力	PH0/RXD5 端子

3.2.5.1　チップ抵抗の除去

TXD5 と RXD5 を使用するために、チップ抵抗を除去する必要があります。除去するチップ抵抗は、R91 と R90 です。

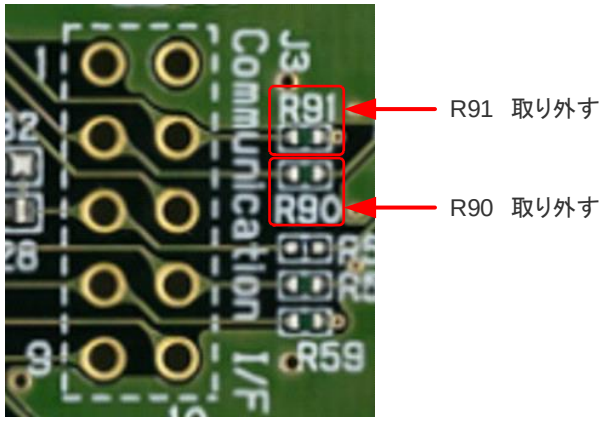

図 3-17　チップ抵抗の除去

3.2.5.2　ピンヘッダーの実装

VSS、VCC、TXD5、RXD5 を使用するために、ピンヘッダーを実装します。

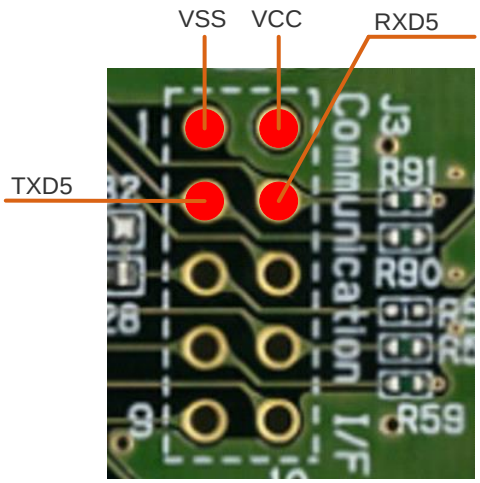

図 3-18　DA16600 Pmod™ Board との接続端子

3.2.5.3　RSSKRX23E-A と DA16600 Pmod™ Board を接続する

VSS、VCC、TXD5、RXD5 と DA16600 Pmod™ Board を以下のように接続します。

RSSKRX23E-A の TXD5 と DA16600 Pmod™ Board の TXD を接続することに注意してください。

表 3-19　接続表

RSSKRX23E-A		DA16600 Pmod™ Board		補足
ピン番号	端子名	ピン番号	信号	
―	―	1	CTS	OPEN
3	PH1/TXD5	2	TXD	
4	PH0/RXD5	3	RXD	
―	―	4	RTS	OPEN
1	VSS	5	GND	
2	VCC	6	VCC	
―	―	7	GPIO	OPEN
―	―	8	GPIO	OPEN
―	―	9	GPIO	OPEN
―	―	10	GPIO	OPEN
―	―	11	GND	OPEN
―	―	12	VCC	OPEN

3.2.6　ソフトウェアの準備

3.2.6.1　Tera Term の準備

Wi-Fi デモでは、RSSKRX23E-A が計測した温度データを、Windows PC に送信します。Windows PC でデータを確認するために、Windows PC に Tera Term をインストールする必要があります。

3.2.7　サンプルプログラムの動作概要

サンプルプログラムの動作概要を示します。サンプルプログラムを動作させるための詳細手順は、「3.2.8 サンプルプログラムの動作詳細」を参照してください。

(1) Start
サンプルプログラムをインポートして、実行します。

(2) Initialization
RSSKRX23E-A は自動的に初期化を行います。また、RSSKRX23E-A は自動的に DA16600 Pmod™ Board を Soft AP モードに設定します。

(3) Wi-Fi Connection
Windows PC から、Wi-Fi で DA16600 Pmod™ Board と接続します。

(4) Tera Term TCP/IP Connection
Tera Term から、TCP/IP で DA16600 Pmod™ Board に接続します。

(5) Temperature data translation
接続が完了すると、自動的に RSSKRX23E-A から Tera Term に温度データが送信されます。

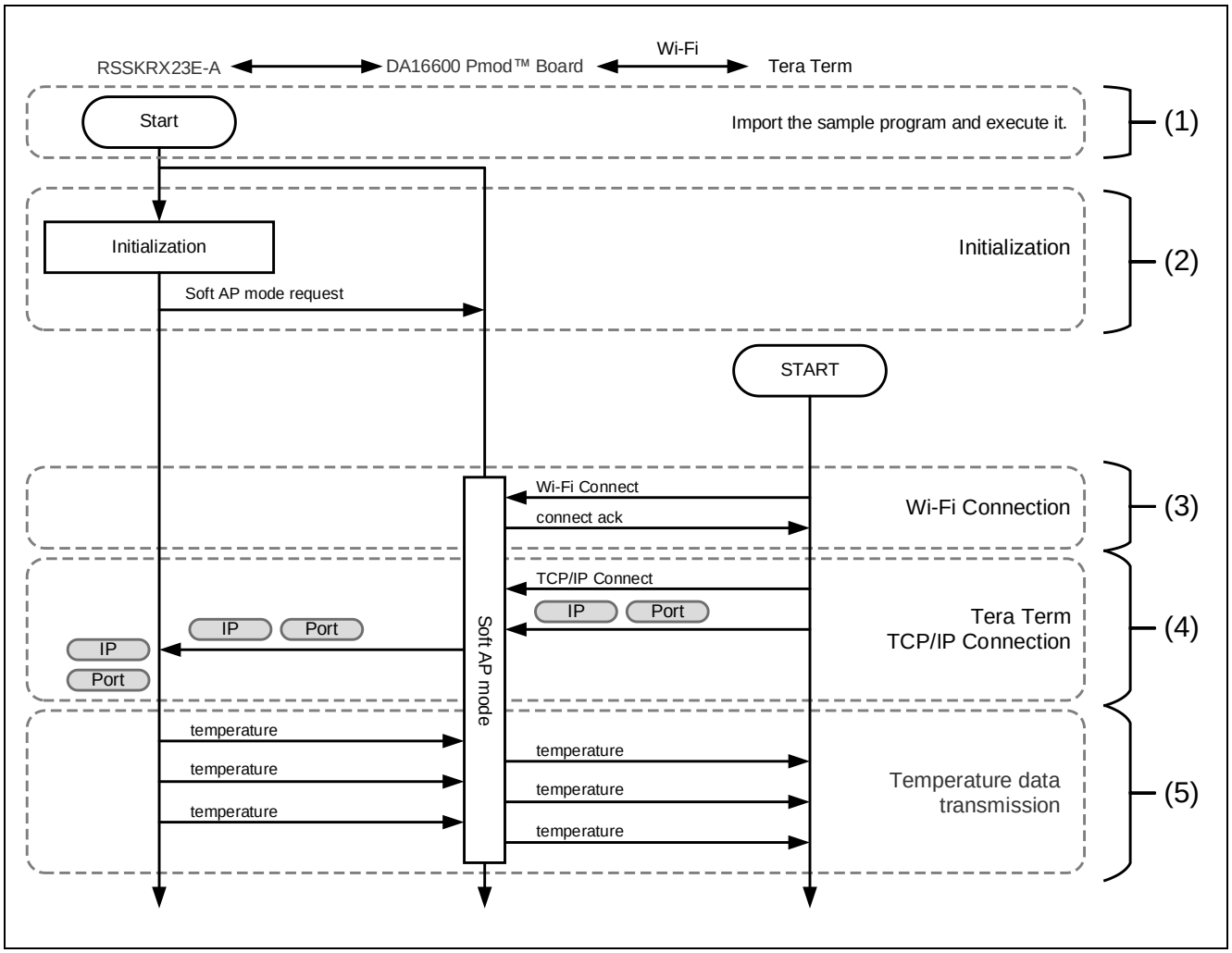

図 3-19　Wi-Fi デモの動作フロー

3.2.8　サンプルプログラムの動作詳細

次の手順でデモを実行してください。

なお、DA16600 Pmod™ Board は一部の設定を NVRAM に保持します。そのため、以前に別のデモを行っていた場合、前回のデモ動作が継続して実行される可能性があります。デバッガ上で数回リスタートを行うと、DA16600 の NVRAM に本デモの条件が書き込まれ、正常に動作する場合があります。デバッガ上でリスタートを繰り返してもうまく動作しない場合は、DA16600 Pmod™ Board のファクトリーリセットを行ってください。

(1)　サンプルプログラムのインポートから実行までの手順

「4 プロジェクトの実行手順」に従って、プロジェクトを実行してください。デバッガから電源が供給されると、LED2 が点灯します。

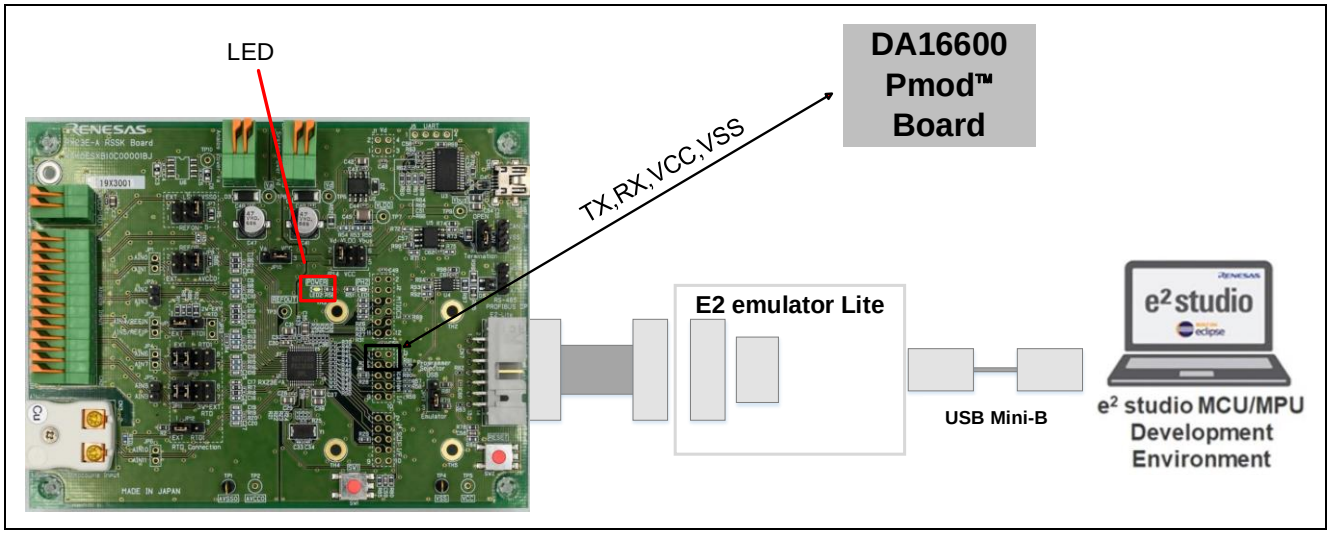

図 3-20　LED2（Power）の位置

(2)　Initialization

RSSKRX23E-A は自動的に初期化を行います。また、RSSKRX23E-A は自動的に DA16600 Pmod™ Board を Soft AP モードに設定します。ログは、Renesas Debug Virtual Console に出力されます。

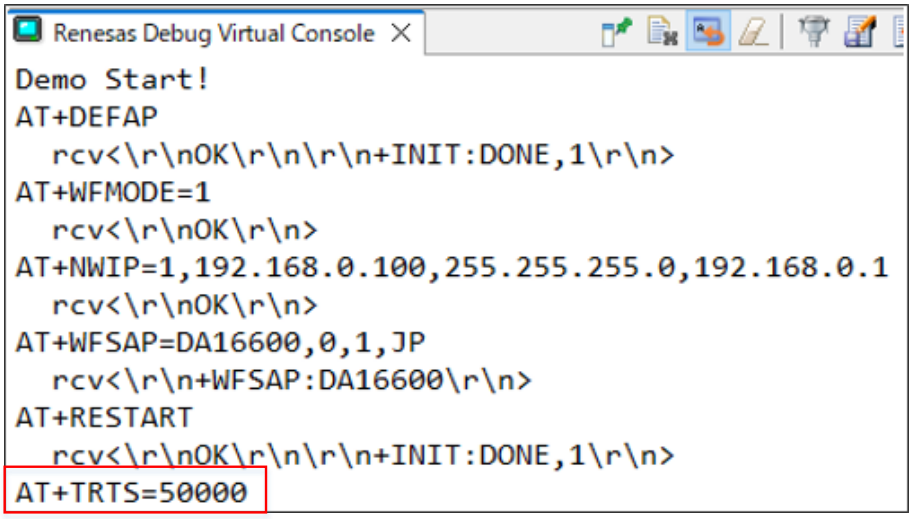

図 3-21　Soft AP mode

(3) Wi-Fi Connection

・Renesas Debug Virtual Console 上に「T+TRTS=50000」と表示されるのを待ちます。
・PC で、「設定」 → 「ネットワークとインターネット」→ 「Wi-Fi」の順に選択します。
・Wi-Fi を「オン」にすると、利用可能な Wi-Fi が検索されます。
・検索された Wi-Fi の中から、「DA16600 の SSID」を選択します。
注：ネットワークに接続する前に、ネットワークのパスワードの入力や、利用規約への同意を求められる場合があります。

図 3-22　Wi-Fi Connection

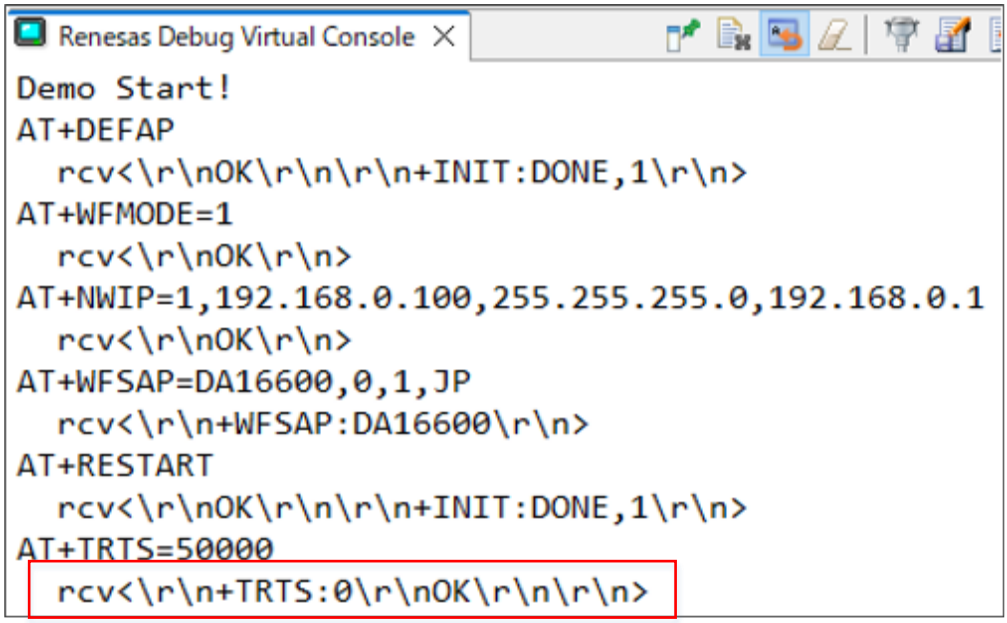

図 3-23　Wi-Fi 接続成功時のログ

(4) Tera Term TCP/IP Connection
DA16600 Pmod™ Board に Wi-Fi 接続後、Tera Term を起動し、以下の設定で TCP 通信を開始します。

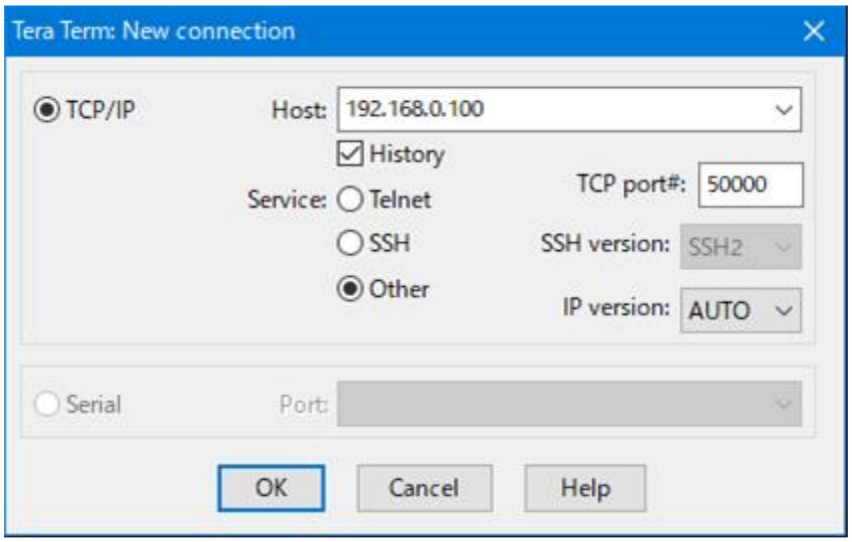

図 3-24　Tera Term 設定

(5) Temperature data transmission
TCP/IP の接続確立後、以下のように RSSKRX23E-A で計測した温度（単位は℃）が表示されます。

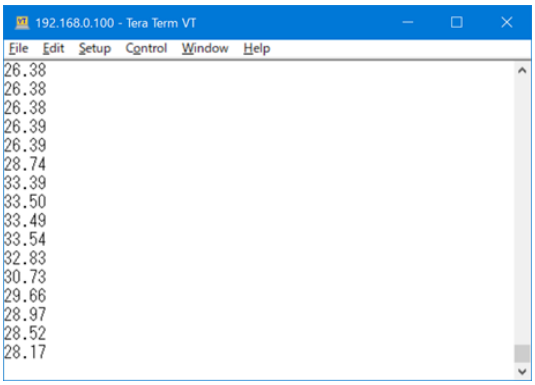

図 3-25　Tera Term での温度データ確認方法

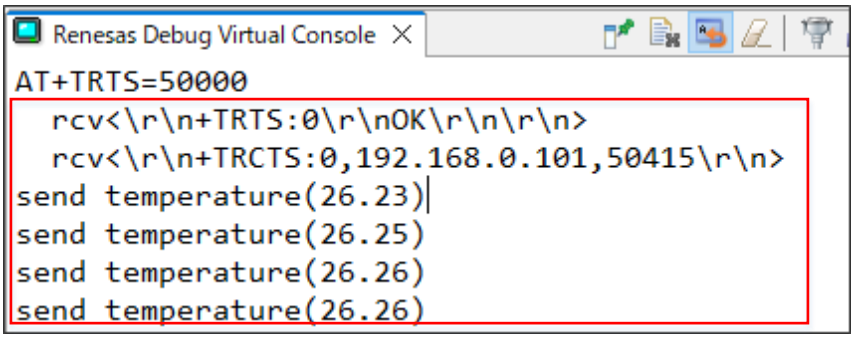

図 3-26　Renesas Debug Virtual Console ログの例

3.3　AWS デモプログラム（r01an6677_rx23ea_aws）

AWS デモを行うために、RSSKRX23E-A と DA16600 Pmod™ Board を接続します。

Wi-Fi デモを行うプロジェクトは、r01an6677_rx23ea_aws です。本プロジェクトを実行するためには、ハードウェアを改造する必要があります。プロジェクトを実行する場合は、「3.3.5 ハードウェアの準備」に進んでください。

AWS にデータを送信するためには、AWS での事前準備が必要です。AWS のアカウントは、お客様ご自身で準備していただく必要があります。AWS のアカウントをご準備いただいた後、「モノの登録」や、「証明書の発行」など、いくつかの手続きが必要になります。これらの手続きについては、「3.3.6 AWS 準備」を参照してください。

「モノの登録」や「証明書の発行」に関する詳細については、「AWS クラウドと FFT を応用した故障検知/動作解析デモンストレーション」の、「3.3 ネットワーク環境と証明書の発行」を参考にしてください。

3.3.1　システム構成

本サンプルプログラムのシステム構成を以下に示します。

RSSKRX23E-A ボードの接続については、「RX23E-A グループ 熱電対を使用した温度計測例」アプリケーションノート（R01AN4747）の Page4 図 4-1 を参照してください。

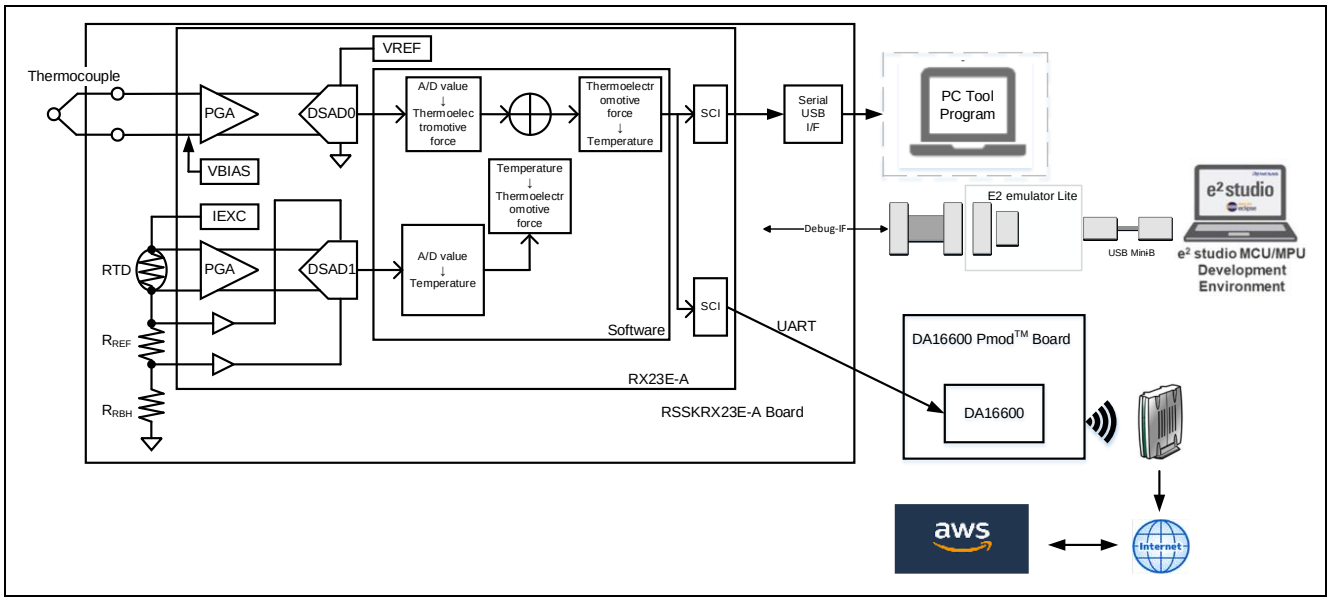

図 3-27　AWS デモのシステム構成

注：PC ツールは、本デモでは必要ありませんが、必要な場合は使用できます。

3.3.2　ソフトウェア構成

本サンプルプログラムのソフトウェア構成を示します。RSSKRX23E-A Board の青色の部分は流用元のサンプルプログラムから変更がない部分です。Wi-Fi や AWS との通信は、全て DA16600 Pmod™ Board が行います。

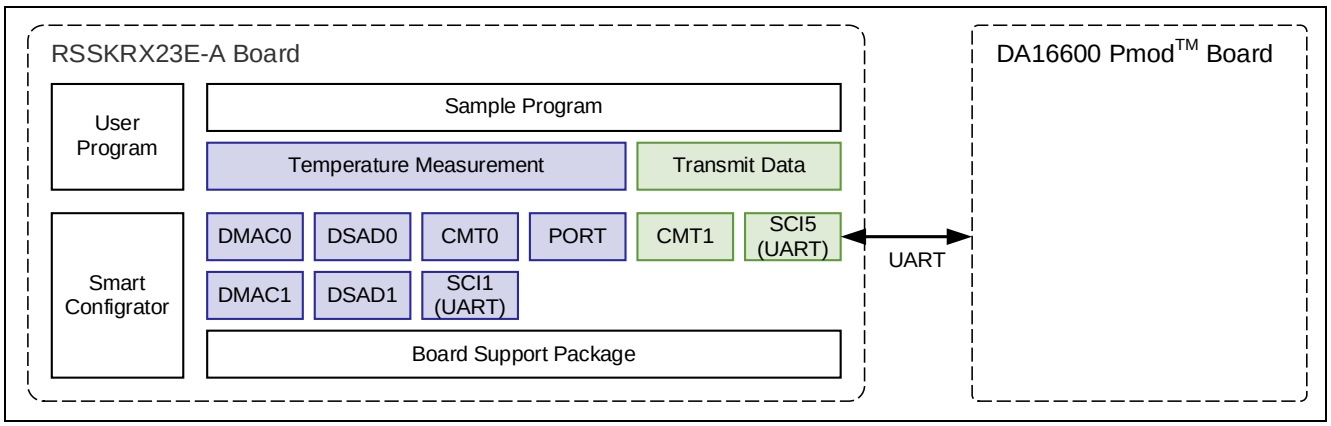

図 3-28　Wi-Fi デモのソフトウェア構成

3.3.3　概略フローチャート

本サンプルプログラムの概略フローチャートを以下に示します。

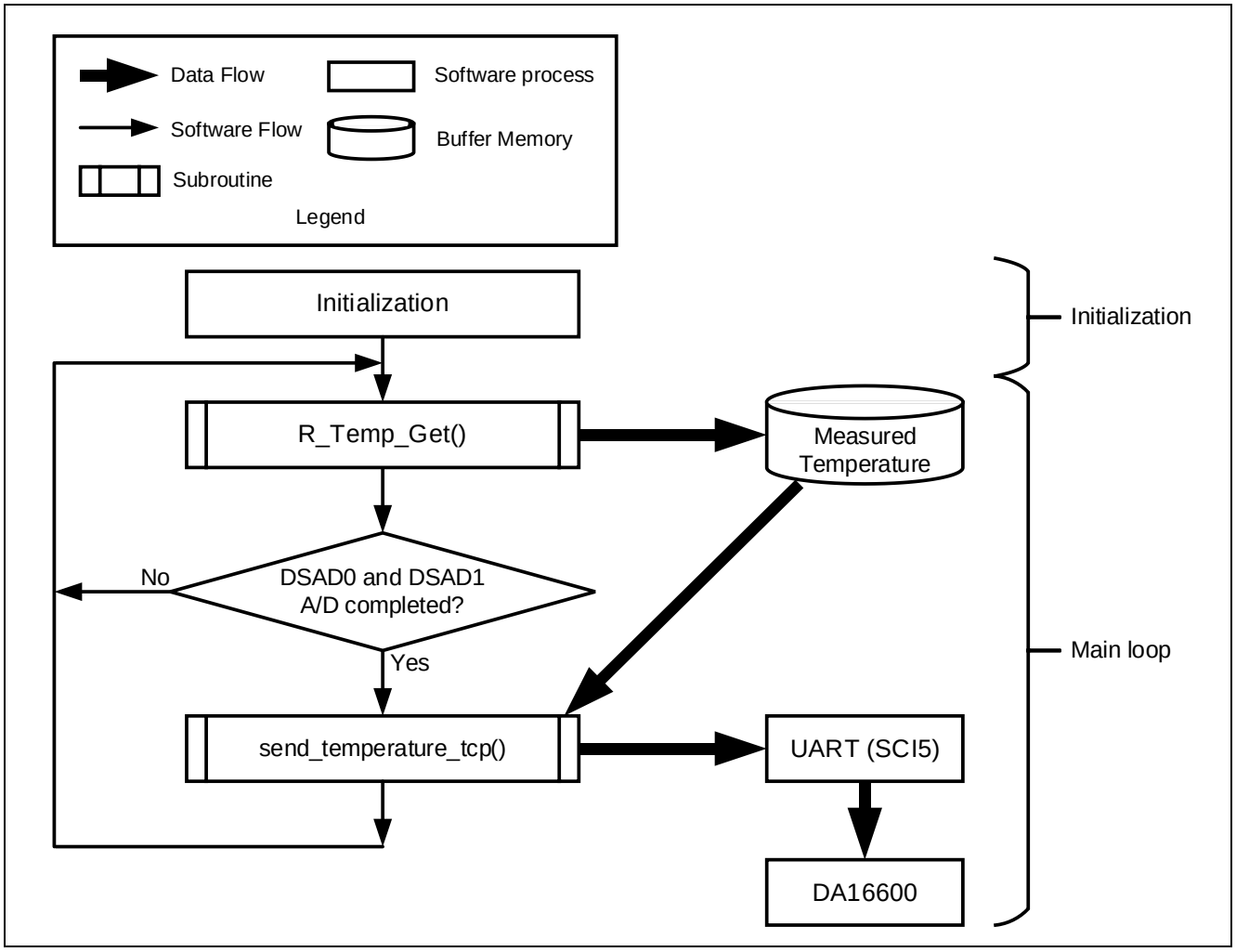

図 3-29　概略フローチャート

3.3.4　サンプルプログラムの構成

3.3.4.1　使用端子一覧

本サンプルプログラムの RX23E-A の使用端子一覧を以下に示します。

表 3-20　使用端子一覧

端子名	入出力	用途
PH2	出力	LED1 点灯制御
P26/TXD1	出力	UART1 送信端子
P30/RXD1	入力	UART1 受信端子
P31/CTS1#	入力	CTS 信号入力端子
AIN11	入力	熱電対+側入力端子
AIN10	入力	熱電対-側入力端子
AIN9	出力	RTD 励起電流出力端子
AIN7	入力	RTD +側入力端子
AIN6	入力	RTD -側入力端子
AIN5/REF1P	入力	RTD 測定 DSAD+側基準電圧
AIN4/REF1N	入力	RTD 測定 DSAD-側基準電圧
PH1/TXD5 注 1	出力	DA16600 Pmod™ Board の GPIOC7_TXD_HOST に接続
PH0/RXD5 注 1	入力	DA16600 Pmod™ Board の GPIOC6_RXD_HOST に接続
VCC 注 1	―	DA16600 Pmod™ Board に 3.3V を供給
VSS 注 1	―	DA16600 Pmod™ Board の VSS に接続

注 1.　ベースとする「RX23E-A グループ 熱電対を使用した温度計測例」アプリケーションノート（R01AN4747）から追加した端子

3.3.4.2　使用する周辺機能

本サンプルプログラムで使用する周辺機能を以下に示します。

表 3-21　使用する周辺機能一覧

周辺機能	用途	追加
AFE、DSAD0、DSAD1	熱電対、RTD の駆動(AFE)、熱電対の A/D 変換(DSAD0)、RTD の A/D 変換(DSAD1)	―
SCI1	PC ツールプログラムとの UART 通信	―
DMAC0	SCI1 の受信完了割り込みをトリガにデータ転送	―
DMAC3	SCI1 のバッファ空き割り込みをトリガにデータ転送	―
CMT0	SCI1 の通信タイムアウト検出	―
PH2	LED1 点灯制御	―
SCI5 注 1	DA16600 Pmod™ Board との UART 通信	yes
CMT1 注 1	温度データ送信のインターバル制御	yes

注 1.　ベースとする「RX23E-A グループ 熱電対を使用した温度計測例」アプリケーションノート（R01AN4747）から追加した周辺機能

3.3.4.3　周辺機能の設定

本サンプルプログラムで使用している周辺機能の設定はスマート・コンフィグレータのコード生成機能を用いています。スマート・コンフィグレータの設定条件を以下に示します。ベースとする「RX23E-A グループ 熱電対を使用した温度計測例」アプリケーションノート（R01AN4747）から追加した周辺機能を記載します。

表 3-22　SCI5 の設定

項目	設定
シリアル通信方式	調歩同期式
スタートビット検出設定	RXD5 端子の Low レベル
データ・ビット長	8 ビット
パリティ設定	禁止
ストップビット設定	1 ビット
データ転送方向設定	LSB ファースト
転送速度設定	• 転送クロック：内部クロック • ビットレート：115200bps • ビットレートモジュレーション機能有効 • SCK5 端子機能：SCK5 を使用しない
ノイズフィルタ設定	ノイズフィルタ無効
ハードウェアフロー制御設定	ハードウェアフロー制御設定：禁止
データ処理設定	送信データ処理：割り込みサービスルーチンで処理 受信データ処理：割り込みサービスルーチンで処理
割り込み設定	受信エラー割り込み許可 優先順位：レベル 15
コールバック機能設定	使用しない
入出力端子	• 出力：TXD5 (PH1) • 入力：RXD5 (PH0)

表 3-23　CMT1 の設定

項目	設定
クロック設定	PCLKB/512
コンペアマッチ設定	• インターバル時間：10ms • コンペアマッチ割り込みを許可 (CMI1) • 優先順位：レベル 15 (割り込み禁止)

3.3.4.4　ファイル構成

本サンプルプログラムのファイル構成を以下に示します。

表 3-24　ファイル構成

フォルダ名、ファイル名	説明
src	プログラム格納用フォルダ
├ main.c 注 1	メイン処理
├ r_ring_buffer_control_api.c	リングバッファ制御プログラム
├ r_ring_buffer_control_api.h	リングバッファ制御 API 定義
├ r_sensor_common_api.c	テーブル検索、直線補間処理プログラム
├ r_sensor_common_api.h	テーブル検索、直線補間処理 API 定義
├ r_thermocouple_api.c	熱電対計測演算プログラム、温度対熱起電力テーブル
├ r_thermocouple_api.h	熱電対計測演算 API 定義
├ r_rtd_api.c	測温抵抗体計測演算プログラム、温度対抵抗値テーブル
├ r_rtd_api.h	測温抵抗体計測演算 API 定義
├ r_communication_control_api.c	通信制御プログラム
├ r_communication_control_api.h	通信制御 API 定義
├ r_mqtt_config.c 注 1	MQTT 通信用設定情報定義
├ r_mqtt_config.h 注 1	MQTT 通信用設定情報変数宣言
├ string_func.c 注 1	AT コマンド制御プログラム
├ string_func.h 注 1	AT コマンド制御 API 定義
└ smc_gen	スマート・コンフィグレータ生成
├ Config_AFE	
├ Config_CMT0	
├ Config_CMT1 注 1	
├ Config_DMAC0	
├ Config_DMAC3	
├ Config_DSAD0	
├ Config_DSAD1	
├ Config_PORT	
├ Config_SCI1	
├ Config_SCI5 注 1	
├ general	
├ r_bsp	
├ r_config	
└ r_pincfg	

注 1.　ベースとする「RX23E-A グループ 熱電対を使用した温度計測例」アプリケーションノート（R01AN4747）から追加、変更を加えたファイル

3.3.4.5　変数一覧

本サンプルプログラムで使用する変数一覧を以下に示します。

ベースとする「RX23E-A グループ 熱電対を使用した温度計測例」アプリケーションノート（R01AN4747）から追加した変数を記載します。追加した変数以外は、R01AN4747 を参照下さい。

表 3-25　サンプルプログラムで使用する変数一覧

変数名	型	内容
g_temp	volatile float	温度データ
g_rcv_end_flg	volatile uint8_t	DA16600 Pmod™ Board コマンド応答受信フラグ
g_send_flg	volatile uint8_t	温度データ送信完了フラグ
g_rcv_buf	uint8_t	受信データを格納するバッファ
g_mqtt_root_certificate_pem	uint8_t	ルート証明書
g_mqtt_certificate_pem_cert	uint8_t	コード署名証明書
g_mqtt_private_pem_key	uint8_t	プライベートキー
g_mqtt_broker_endpoint	uint8_t	AWS エンドポイント
g_mqtt_broker_port	uint8_t	MQTT broker ポート番号
g_mqtt_subscriber	uint8_t	AWS モノの名前
g_mqtt_publisher	uint8_t	MQTT トピック
g_wifi_ssid	uint8_t	アクセスポイントの SSID
g_wifi_password	uint8_t	アクセスポイントのパスワード
g_root_certificate_pem_size	uint16_t	ルート証明書の文字数
g_certificate_pem_cert_size	uint16_t	コード署名証明書の文字数
g_private_pem_key_size	uint16_t	プライベートキーの文字数
g_broker_endpoint_size	uint16_t	AWS エンドポイントの文字数
g_broker_port_size	uint16_t	MQTT broker ポート番号の文字数
g_subscriber_size	uint16_t	AWS モノの名前の文字数
g_publisher_size	uint16_t	MQTT トピックの文字数
g_wifi_size	uint16_t	アクセスポイントの SSID の文字数
g_password_size	uint16_t	アクセスポイントのパスワードの文字数

3.3.4.6　定数一覧

ベースとする「RX23E-A グループ 熱電対を使用した温度計測例」アプリケーションノート（R01AN4747）から追加した定数はありません。定数一覧は、R01AN4747 を参照下さい。

3.3.4.7　関数一覧

本サンプルプログラムで使用する関数一覧を以下に示します。

ベースとする「RX23E-A グループ 熱電対を使用した温度計測例」アプリケーションノート（R01AN4747）から追加、変更を加えた関数を記載します。

表 3-26　サンプルプログラムで使用する関数一覧

関数名	概要	
main	メイン処理	変更
start_mqtt_mode	DA16600 Pmod™ Board を MQTT 通信可能な状態に設定	追加
check_sci_rcv_end	DA16600 Pmod™ Board の応答を検出	追加
check_rcv_cmd	DA16600 Pmod™ Board から受信した内容をチェック	追加
reset_rcv_buf	受信バッファをクリア	追加
send_temperature_mqtt	温度データを送信	追加

3.3.4.8　関数仕様

本サンプルプログラムの関数仕様を以下に示します。

[関数名]main

概要	メイン処理
ヘッダ	なし
宣言	void main (void)
説明	周辺機能を初期化します。熱電対を使用した温度計測、DA16600 Pmod™ Board の制御、温度データの送信を行います。
引数	なし
戻り値	なし
備考	なし

[関数名] start_mqtt_mode

概要	DA16600 Pmod™ Board を MQTT 通信可能な状態に設定
ヘッダ	string_func.h
宣言	void start_mqtt_mode (void)
説明	DA16600 Pmod™ Board に AT コマンドを送信します。 DA16600 Pmod™ Board を STA モードに設定し、MQTT 通信可能な状態にします。
引数	なし
戻り値	なし
備考	なし

[関数名] check_sci_rcv_end

概要	DA16600 Pmod™ Board の応答を検出
ヘッダ	string_func.h
宣言	void check_sci_rcv_end (void)
説明	改行コードから、DA16600 Pmod™ Board の応答を検出します。
引数	なし
戻り値	なし
備考	なし

[関数名] reset_rcv_buf

概要	受信バッファをクリア
ヘッダ	string_func.h
宣言	void reset_rcv_buf (void)
説明	受信バッファをクリアします。
引数	なし
戻り値	なし
備考	なし

[関数名] send_temperature_mqtt

概要	温度データを送信
ヘッダ	string_func.h
宣言	void send_temperature (void)
説明	送信バッファに送信用コマンドと温度データをセットし、DA16600 Pmod™ Board に送信します。
引数	なし
戻り値	なし
備考	なし

3.3.4.9　AWS デモのフローチャート

AWS デモの main 関数フローチャートを示します。

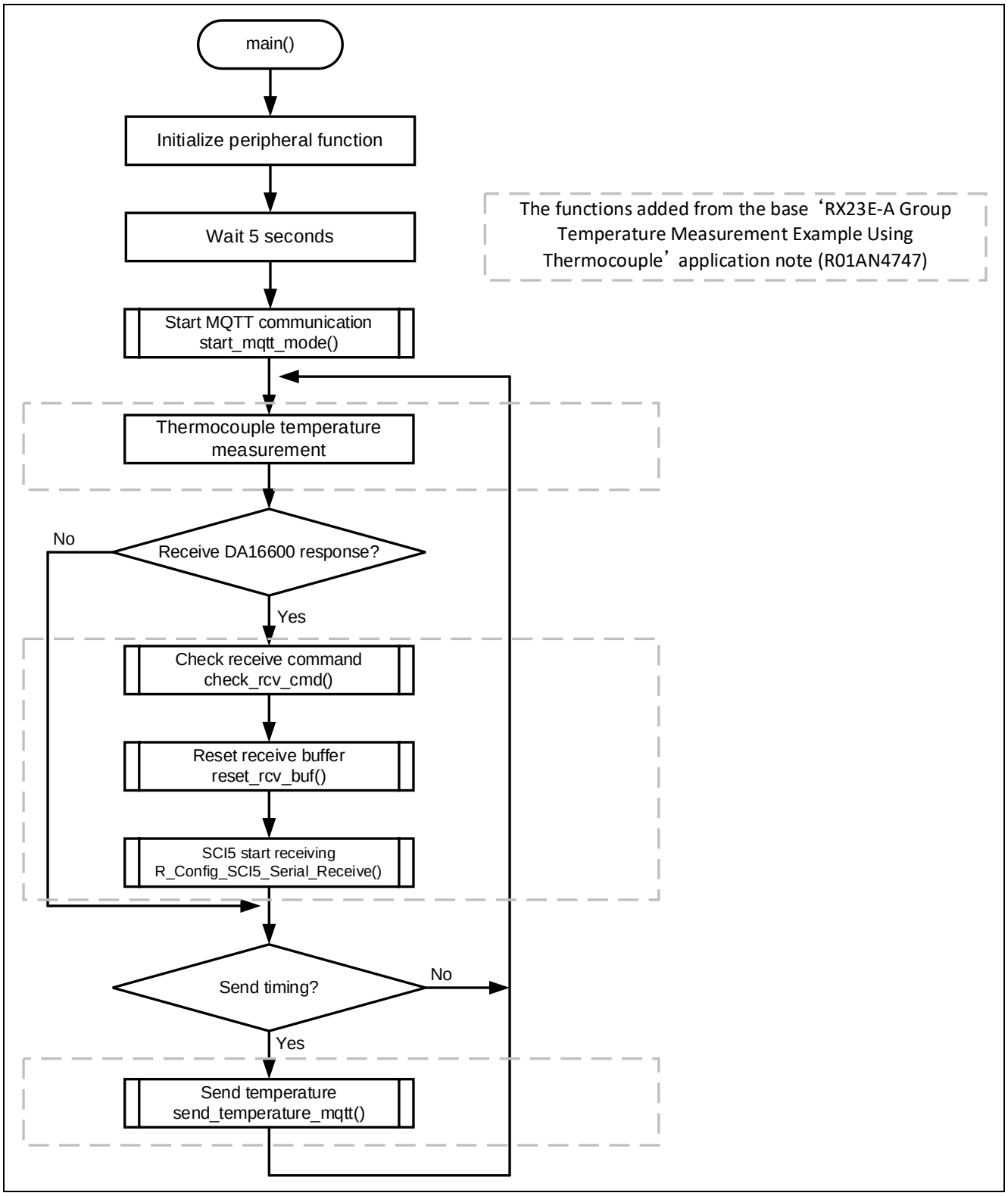

図 3-30　main 関数フローチャート

AWS デモの start_mqtt_mode 関数フローチャートを示します。

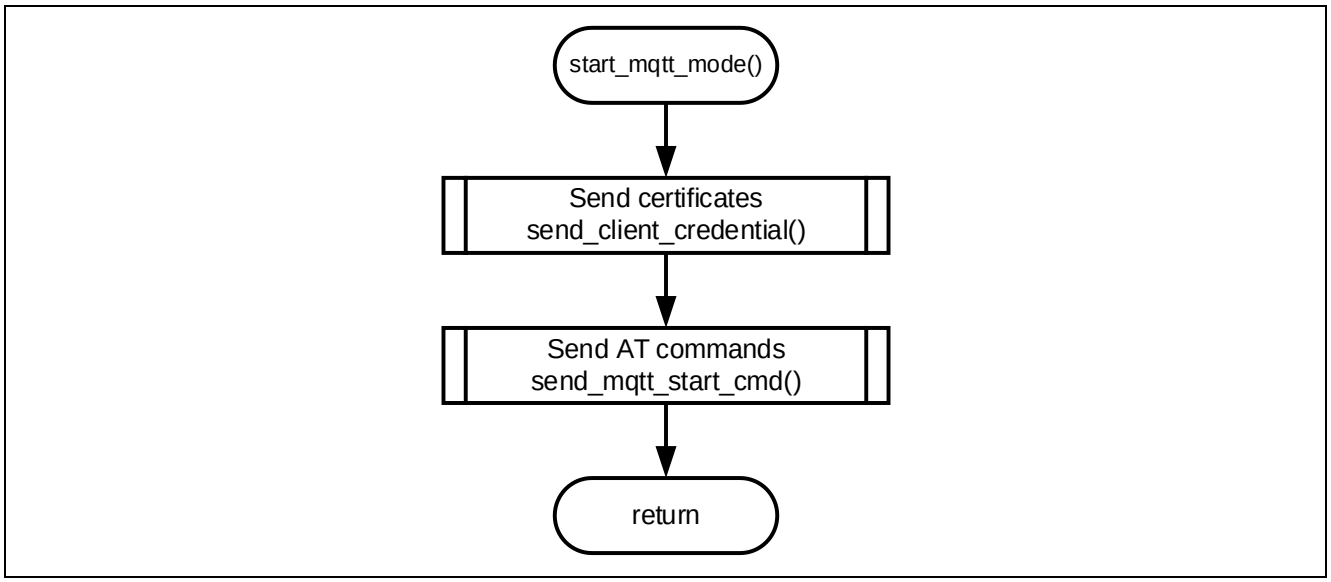

図 3-31　start_mqtt_mode 関数フローチャート

AWS デモの send_client_credential 関数フローチャートを示します。

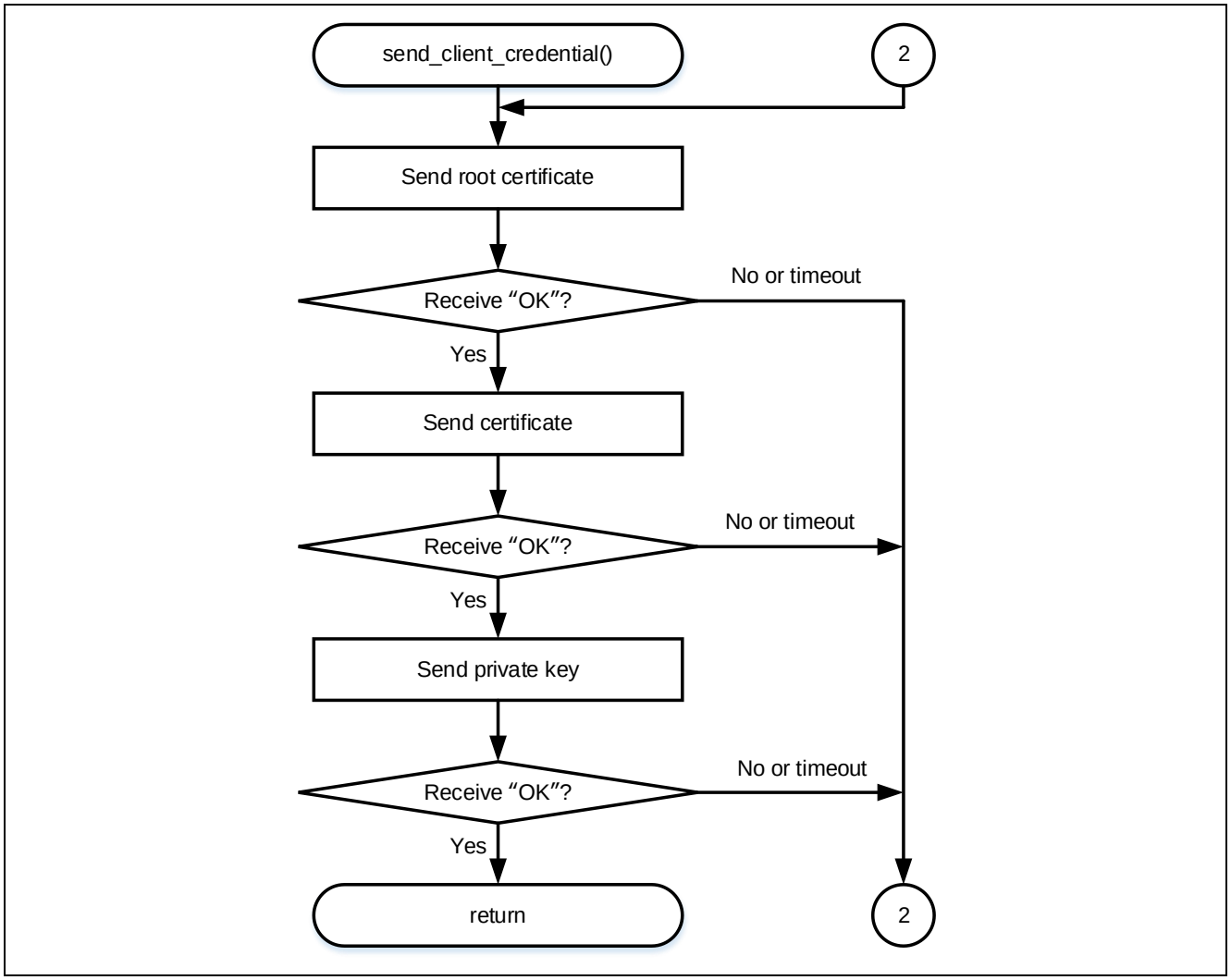

図 3-32　send_client_credential 関数フローチャート

AWS デモの send_mqtt_start_cmd 関数フローチャートを示します。

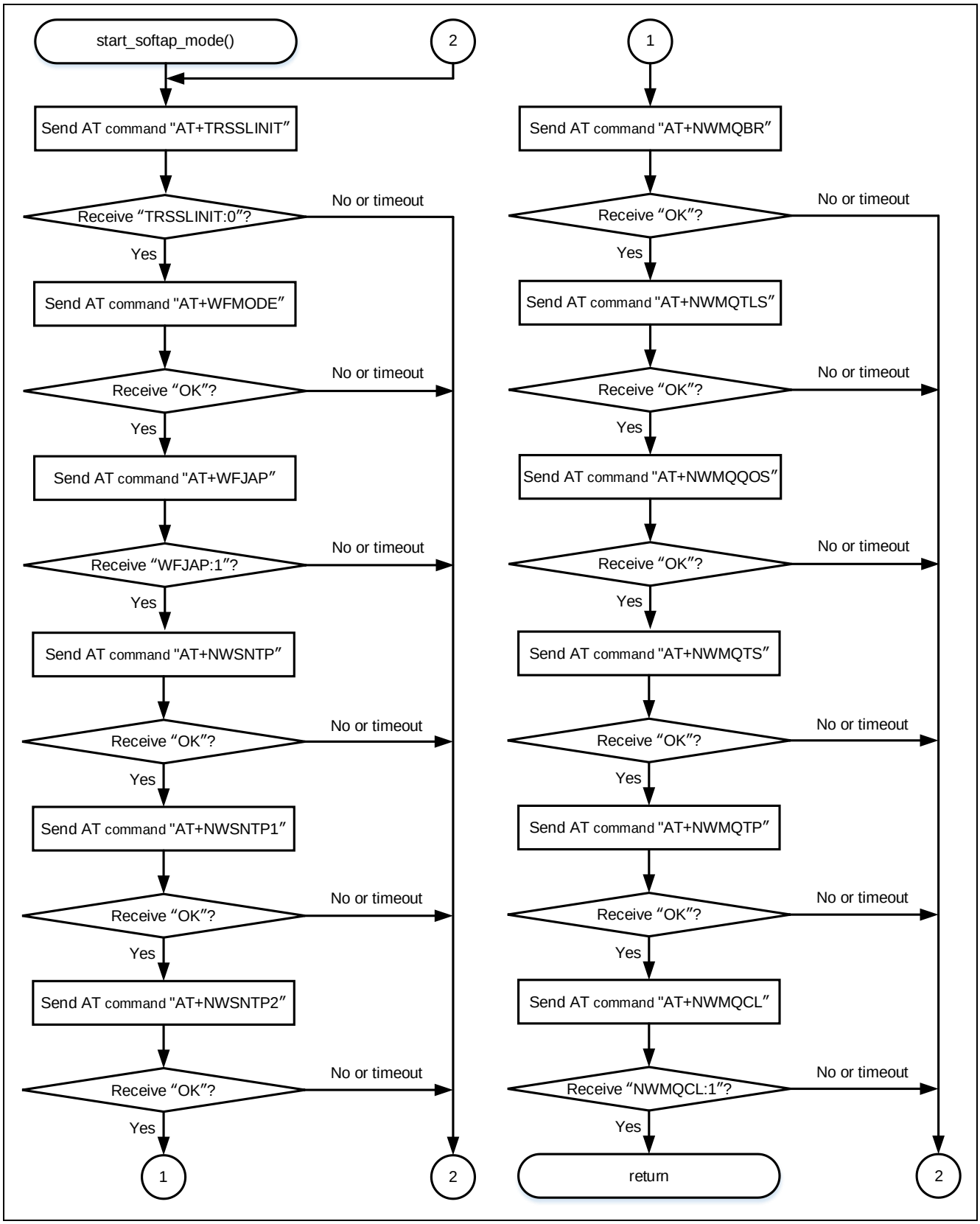

図 3-33　send_mqtt_start_cmd 関数フローチャート

3.3.5　ハードウェアの準備

本アプリケーションノートでは、RSSKRX23E-A ボードの熱電対計測回路を使用します。使用方法の詳細は、「RSSKRX23E-A ユーザーズマニュアル」の「2.4 アナログ入力回路の使用方法」を参照してください。

RSSKRX23E-A と DA16600 Pmod™ Board を接続するために、RSSKRX23E-A ボードを改造する必要があります。

改造する端子一覧を以下に示します。ピン番号の詳細は、「RSSKRX23E-A ユーザーズマニュアル」を参照してください。

表 3-27　改造する端子一覧

ピン番号	MCU 端子番号	機能	入出力	説明
1	ー	VSS	出力	VSS 端子
2	ー	VCC	出力	VCC 端子 外部への電源供給に使用
3	23	PH1/TXD5	入出力	PH1/TXD5 端子
4	24	PH0/RXD5	入出力	PH0/RXD5 端子

3.3.5.1　チップ抵抗の除去

TXD5 と RXD5 を使用するために、チップ抵抗を除去する必要があります。除去するチップ抵抗は、R91 と R90 です。

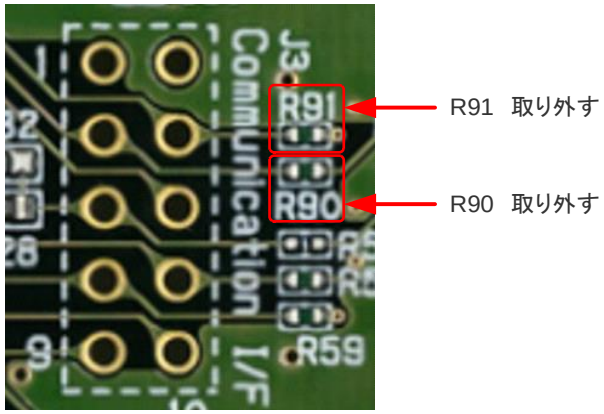

図 3-34　チップ抵抗の除去

3.3.5.2　ピンヘッダーの実装

VSS、VCC、TXD5、RXD5 を使用するために、ピンヘッダーを実装します。

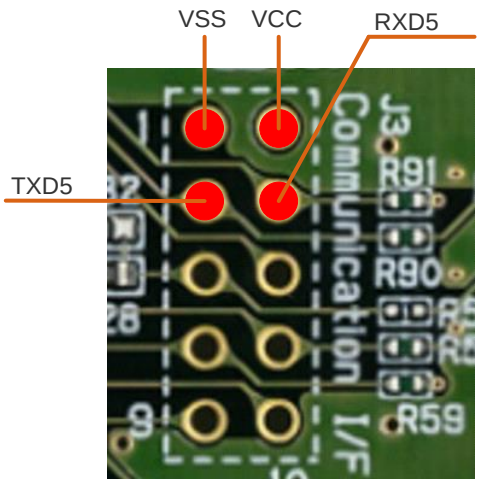

図 3-35　DA16600 Pmod™ Board との接続端子

3.3.5.3　RSSKRX23E-A と DA16600 Pmod™ Board と接続する

VSS、VCC、TXD5、RXD5 と DA16600 Pmod™ Board を以下のように接続します。

RSSKRX23E-A の TXD5 と DA16600 Pmod™ Board の TXD を接続することに注意してください。

表 3-28　接続表

RSSKRX23E-A		DA16600 Pmod™ Board		補足
ピン番号	端子名	ピン番号	信号	
―	―	1	CTS	OPEN
3	PH1/TXD5	2	TXD	
4	PH0/RXD5	3	RXD	
―	―	4	RTS	OPEN
1	VSS	5	GND	
2	VCC	6	VCC	
―	―	7	GPIO	OPEN
―	―	8	GPIO	OPEN
―	―	9	GPIO	OPEN
―	―	10	GPIO	OPEN
―	―	11	GND	OPEN
―	―	12	VCC	OPEN

3.3.6　AWS 準備

以下のチュートリアルを参考に AWS の設定をしてください。

- デバイスを AWS IoT に登録する
  デバイスを AWS IoT に登録する · renesas/amazon-freertos Wiki · GitHub

注：「AWS IoT のエンドポイントを確認する」まで行ってください。

- ソースコードに AWS 接続情報を入力する
  {rx23ea_thermocouple_aws/src/r_mqtt_config.c} にある 5 つの変数を設定してください。

- g_mqtt_broker_endpoint[]　➡「3.3.6 AWS 準備」で確認した エンドポイント の名前
- g_mqtt_suscriber[]　➡「3.3.6 AWS 準備」で登録した モノ の名前
- g_mqtt_publisher[]　➡「3.3.6 AWS 準備」で登録した モノ の名前/send
- g_wifi_ssid　➡ 接続するアクセスポイントの SSID
- g_wifi_password　➡ 接続するアクセスポイントのパスワード

(上記の変数は、下図のように、”　”の中に入力してください。)

```
uint8_t  g_mqtt_broker_endpoint[] = "[redacted]iot.ap-northeast-1.amazonaws.com";
uint16_t g_broker_endpoint_size   = sizeof(g_mqtt_broker_endpoint);

uint8_t  g_mqtt_subscriber[] = "[redacted]";
uint16_t g_subscriber_size   = sizeof(g_mqtt_subscriber);

uint8_t  g_mqtt_publisher[] = "[redacted]/send";
uint16_t g_publisher_size   = sizeof(g_mqtt_publisher);

uint8_t  g_wifi_ssid[] = "[redacted]";
uint16_t g_wifi_size   = sizeof(g_wifi_ssid);

uint8_t  g_wifi_password[] = "[redacted]";
uint16_t g_password_size   = sizeof(g_wifi_password);
```

図 3-36　r_mqtt_config.c

3.3.6.1　ルート CA 証明書とデバイス証明書とプライベートキーをソースコードに反映する

{rx23ea_thermocouple_aws/src/r_mqtt_config.c} にある 3 つの変数を以下のように反映してください。

- g_mqtt_root_cetificate_pem[] ➡図 3-37 に従い、ダウンロードしたルート CA 証明書
- g_mqtt_cetificate_pem_cert[] ➡図 3-38 に従い、ダウンロードしたデバイス証明書
- g_mqtt_private_pem_key[]　➡図 3-39 に従い、ダウンロードしたプライベートキー

注：以下に注意してください。
　　最終行を除く各行の最後に「\n」が必要
　　各行はダブルクォーテーションで囲われること
　　最終行を除く各行の最終がバックスラッシュで終わっていること

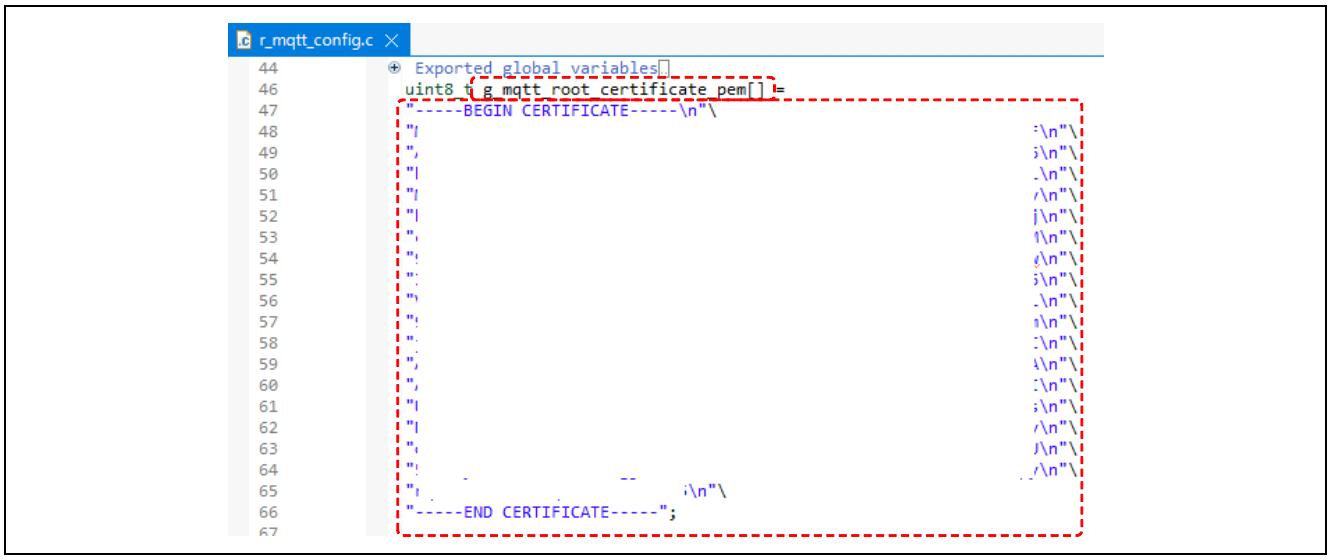

図 3-37　ルート CA 証明書の反映

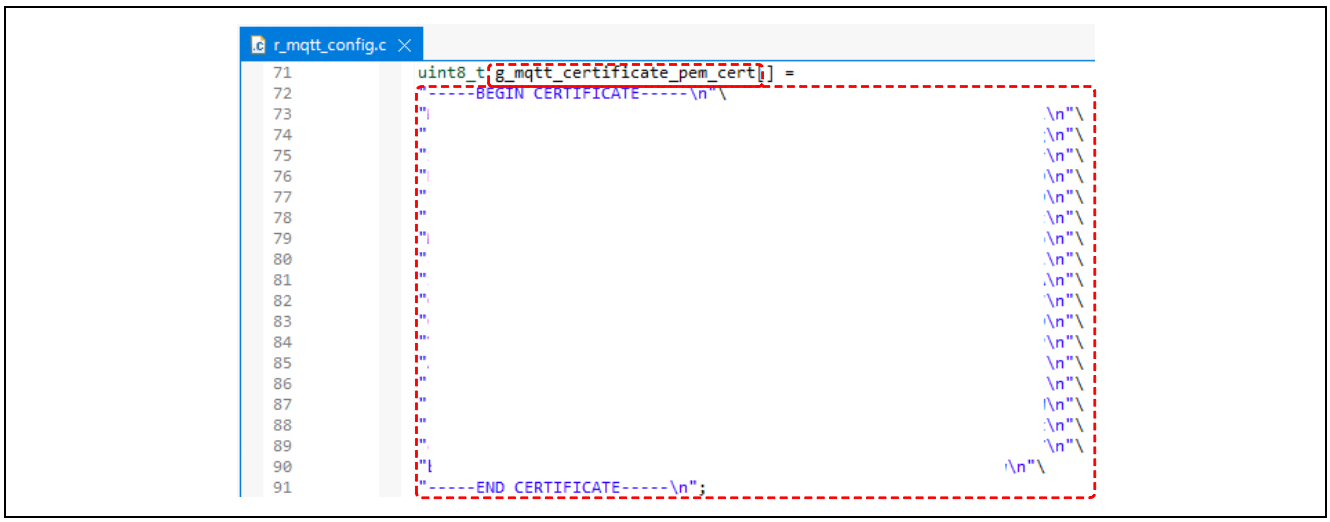

図 3-38　デバイス証明書の反映

図 3-39　プライベートキーの反映

3.3.7　サンプルプログラムの動作概要

サンプルプログラムの動作概要を示します。サンプルプログラムを動作させるための詳細手順は、「3.3.8 サンプルプログラムの動作詳細」を参照してください。

(1) Start
サンプルプログラムをインポートして、実行します。

(2) Initialization
RSSKRX23E-A は自動的に初期化を行います。

(3) Storage of Information
RSSKRX23E-A は自動的に DA16600 Pmod™ Board を STA（Station）モードに設定します。また、RSSKRX23E-A は自動的に DA16600 Pmod™ Board に証明書やプライベートキーを書き込みます。

(4) Wi-Fi Connection
RSSKRX23E-A は、証明書などを書き込み後、自動的に DA16600 Pmod™ Board に Wi-Fi 接続要求を行います。

(5) AWS Connection
RSSKRX23E-A は、Wi-Fi に接続されたことを確認後、DA16600 Pmod™ Board に AWS 接続要求を行います。

(6) MQTT Connection
RSSKRX23E-A は、AWS に接続されたことを確認後、DA16600 Pmod™ Board に MQTT 接続要求を行います。

(7) Temperature data translation
RSSKRX23E-A は、MQTT に接続されたことを確認後、DA16600 Pmod™ Board に温度データを送信します。

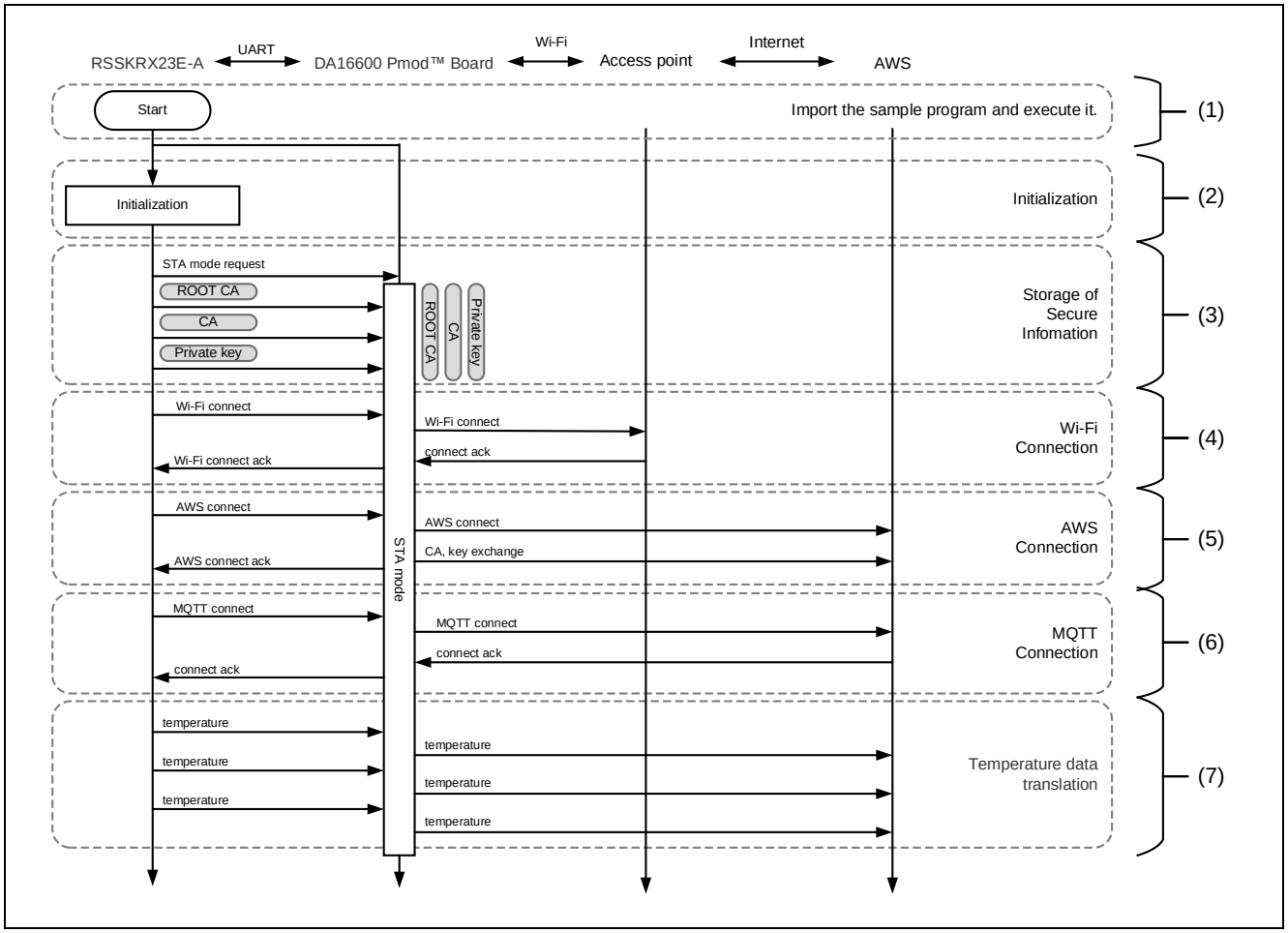

図 3-40　AWS デモの動作フロー

3.3.8　サンプルプログラムの動作詳細

次の手順でデモを実行してください。

なお、DA16600 Pmod™ Board は一部の設定を NVRAM に保持します。そのため、以前に別のデモを行っていた場合、前回のデモ動作が継続して実行される可能性があります。デバッガ上で数回リスタートを行うと、DA16600 の NVRAM に本デモの条件が書き込まれ、正常に動作する場合があります。デバッガ上でリスタートを繰り返してもうまく動作しない場合は、DA16600 Pmod™ Board のファクトリーリセットを行ってください。

(1)　サンプルプログラムのインポートから実行までの手順
「4 プロジェクトの実行手順」に従って、プロジェクトを実行してください。デバッガから電源が供給されると、LED2 が点灯します。

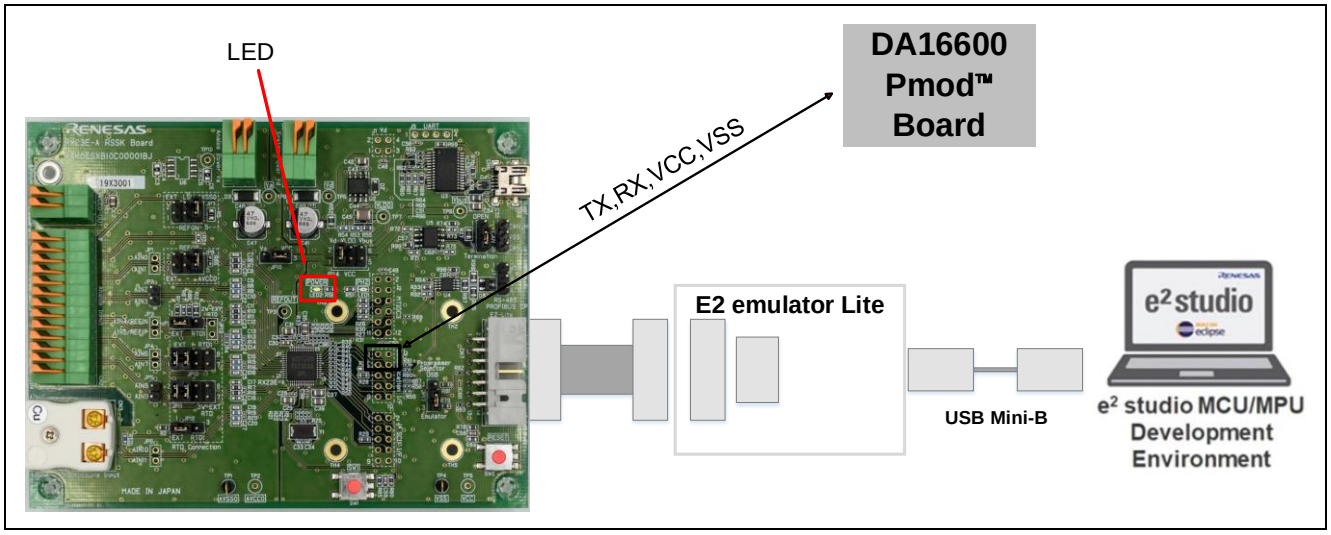

図 3-41　LED2（Power）の位置

(2)　Initialization
RSSKRX23E-A は自動的に初期化を行います。
この後、AWS にデータを転送するまでの全ての動作は、RSSKRX23E-A が自動で行います。

(3)　Storage of Information
RSSKRX23E-A は自動的に DA16600 Pmod™ Board を STA モードに設定します。その後、RSSKRX23E-A は自動的に、証明書と秘密鍵を DA16600 Pmod™ Board に書き込みます。

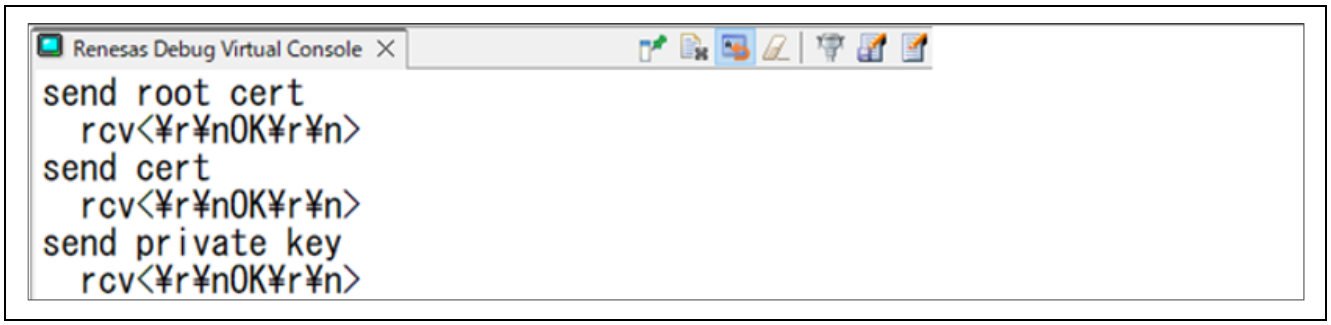

図 3-42　証明書と秘密鍵のログ

(4) Wi-Fi Connection
RSSKRX23E-A は、自動的に DA16600 Pmod™ Board に Wi-Fi 接続要求を行います。

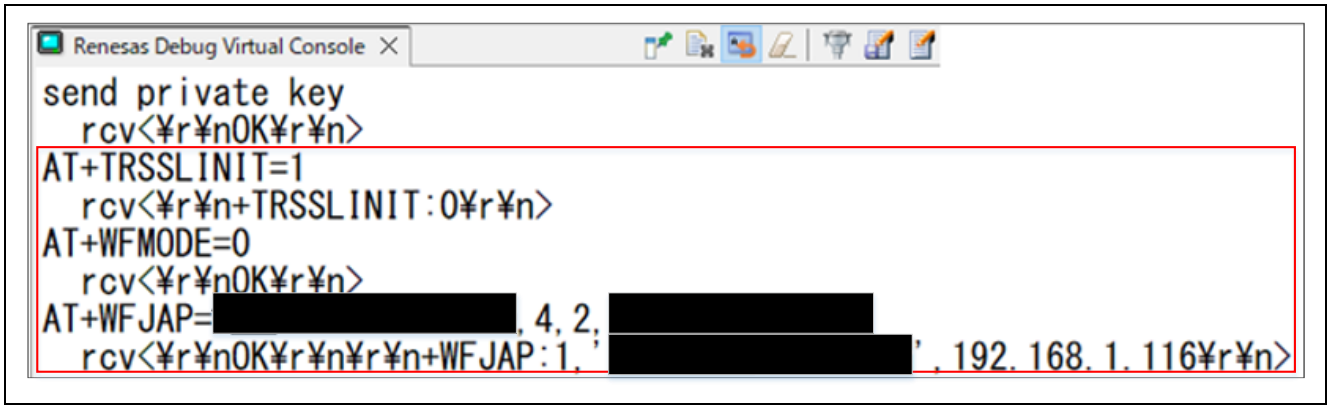

図 3-43　Wi-Fi 要求のログ

(5) RSSKRX23E-A は、自動的に DA16600 Pmod™ Board に AWS 接続要求を行います。
(6) RSSKRX23E-A は、自動的に DA16600 Pmod™ Board に MQTT 接続要求を行います。

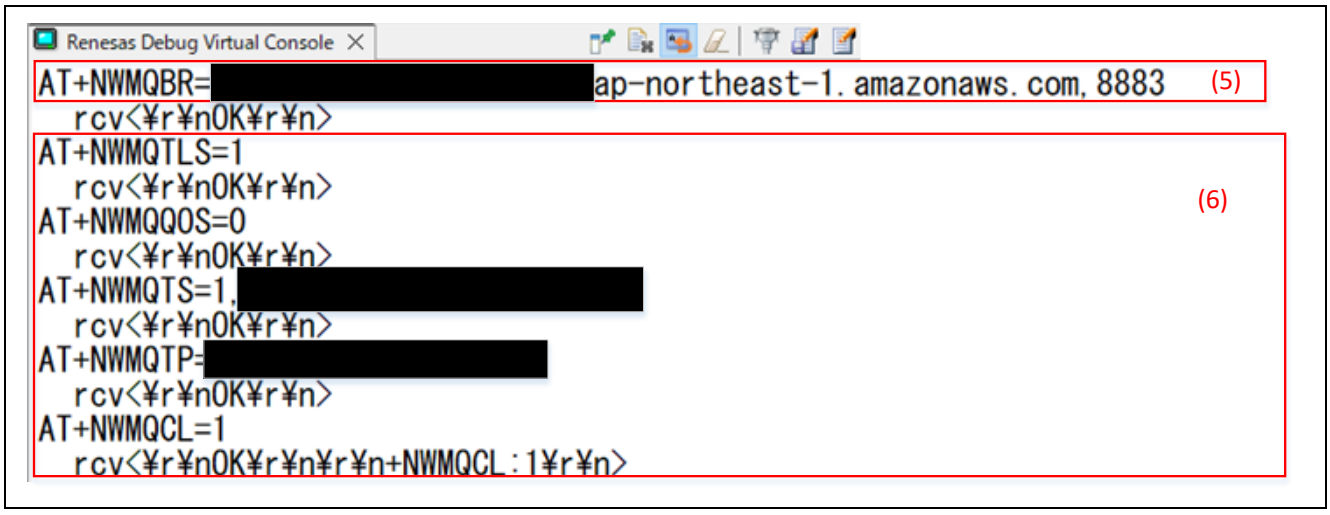

図 3-44　AWS と MQTT のログ

(7) RSSKRX23E-A は、自動的に DA16600 Pmod™ Board に温度データを送信します。

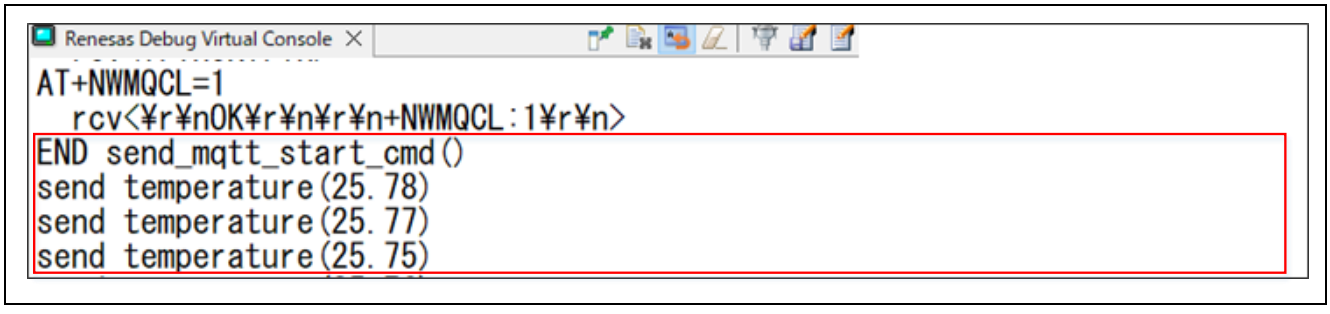

図 3-45　AWS へ温度データを送信

3.3.8.1　AWS で温度データを確認する方法

AWS で温度データを確認する方法を示します。

「IoT Core」 → 「テスト」 → 「MQTT テストクライアント」 → 「トピックをサブスクライブする」のタブを開きます。「トピックのフィルター」にモノの名前を入力します。”/#”は、ワイルドカードです。サブスクライブボタンをクリックすると、受信した温度データを表示します。

図 3-46　AWS での温度データ確認方法

4.　プロジェクトの実行手順

サンプルプログラムは e² studio のプロジェクト形式で提供しています。本章では、e² studio および CS+ へプロジェクトをインポートする方法を示します。インポート完了後、ビルドおよびデバッガの設定を確認してください。

4.1　e² studio での手順

4.1.1　e² studio でのインポート方法

e² studio でご使用になる際は、以下の手順で e² studio にインポートしてください。

なお、e² studio で管理するプロジェクトのフォルダ名、およびそのフォルダに至るファイルパスには、空白文字の他、半角カナ文字、全角文字、半角記号(特に'$','#','%') が混じらないようにしてください。

（使用する e² studio のバージョンによっては画面が異なる場合があります。）

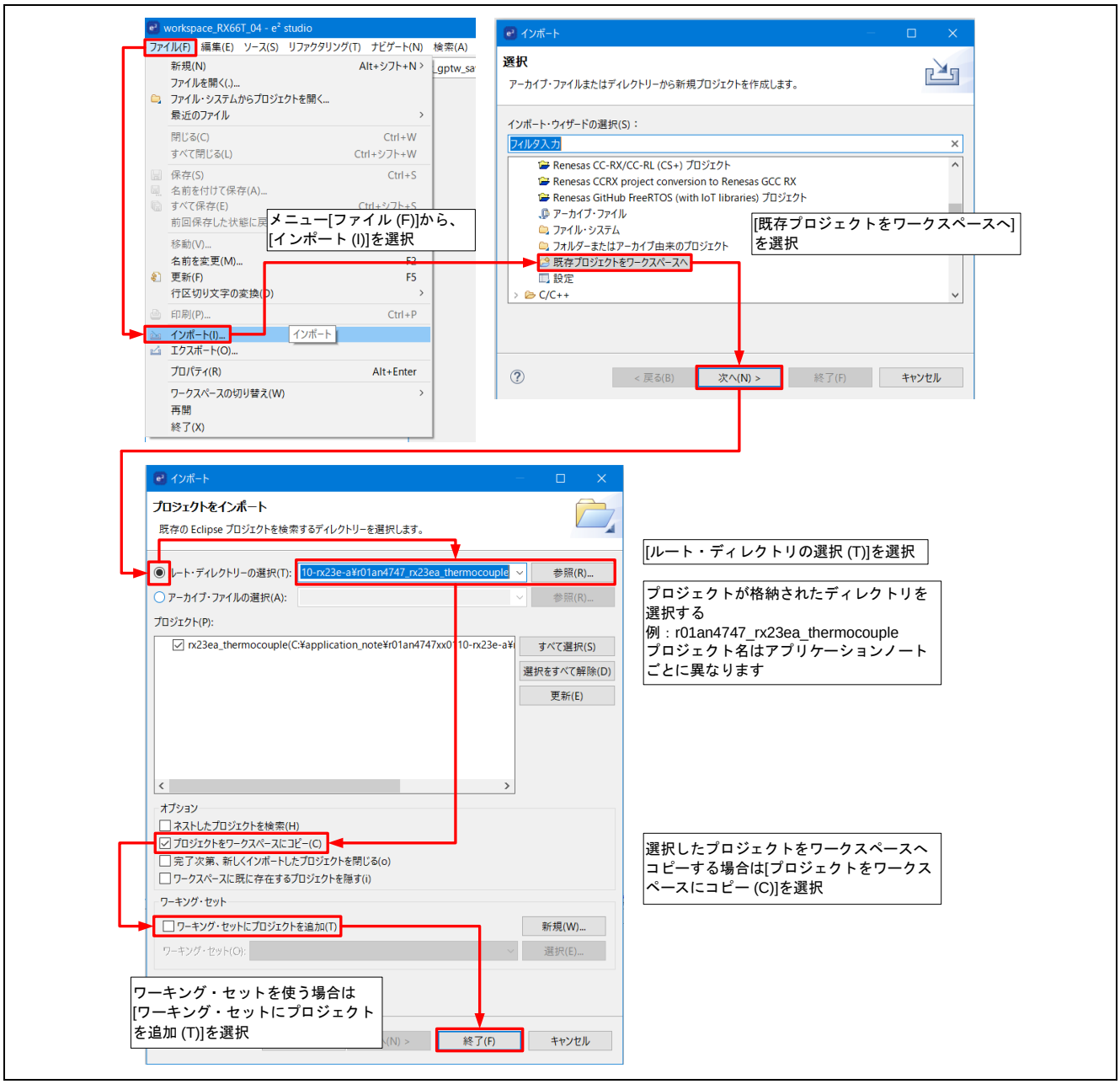

図 4-1　プロジェクトを e² studio にインポートする方法

4.1.2　ビルドオプションの設定

1. プロジェクト名を右クリックし、メニュー表示→「プロパティ」を選択します。
2. 「C/C++ビルド」→「設定」の「Toolchain」タブをクリックして、ツールチェーンとバージョンを確認してください。

—・ツールチェーン：Renesas CC-RX
—・バージョン　：v3.05.00

4.1.3　プロジェクトのビルド

1. プロジェクト・エクスプローラでプロジェクトを右クリックし、「プロジェクトのビルド」を選択します。
2. ビルドが開始され、「コンソール」にビルドの状況が表示されるので”Build Finished”というメッセージが表示されたらビルド完了です。

4.1.4　デバッグ

1. 「実行」→「デバッグの構成…」を選択し、「デバッグ構成」ウィンドウを開きます。
2. 「デバッグ構成」ウィンドウで“プロジェクト名称(HardwareDebug)” デバッグ構成の表示を展開し、既存のデバッグ構成をクリックします。
3. 「debugger」→「Connection Settings」タブに切り替え、下図の設定となっていることを確認してください。
4. 「デバッグ」をクリックすると、RX23E-A にプログラムがダウンロードされます。

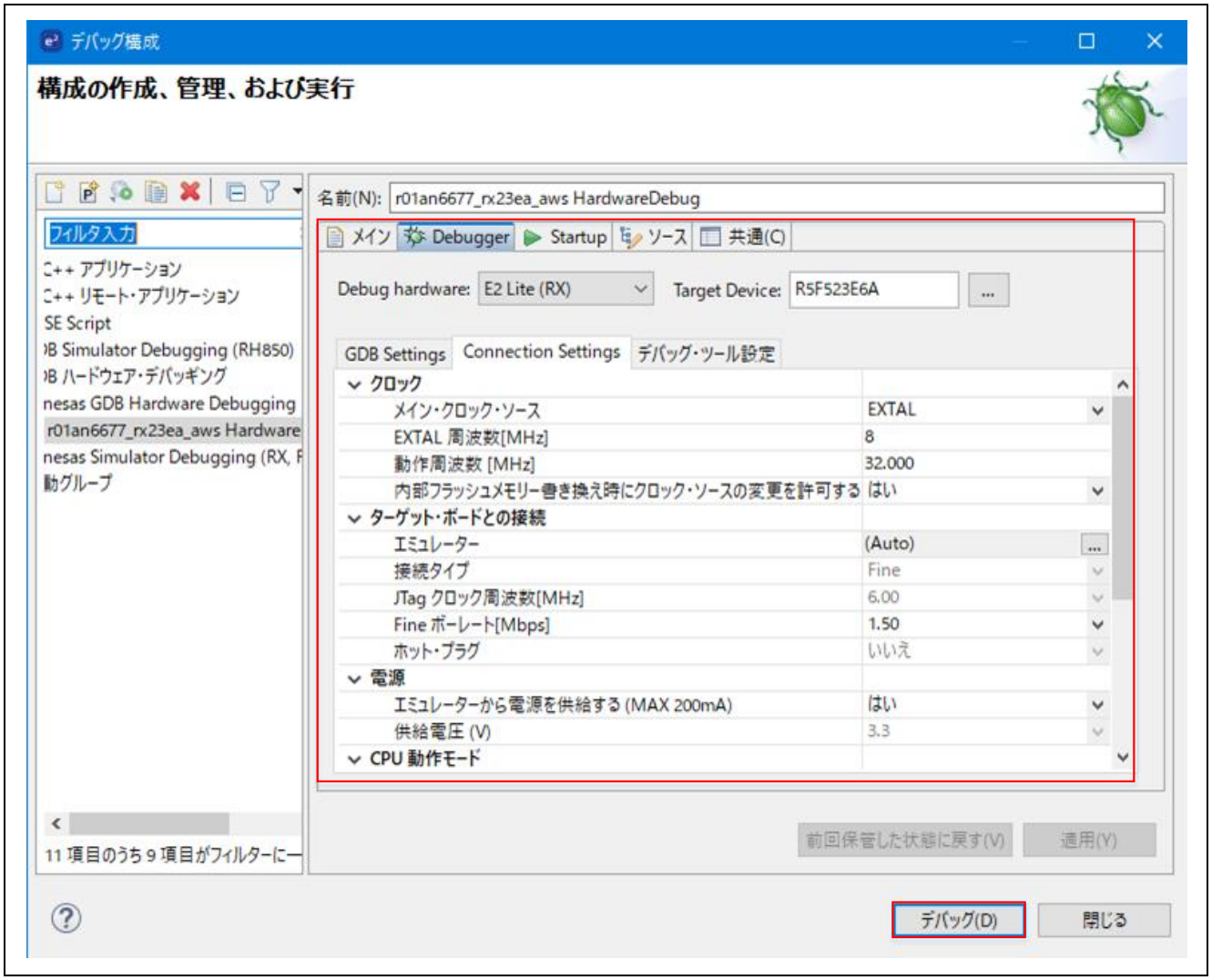

図 4-2　Debug screen settings pickup

5. 「Renesas Views」→「デバッグ」→「Renesas Debug Virtual Console」を選択します。

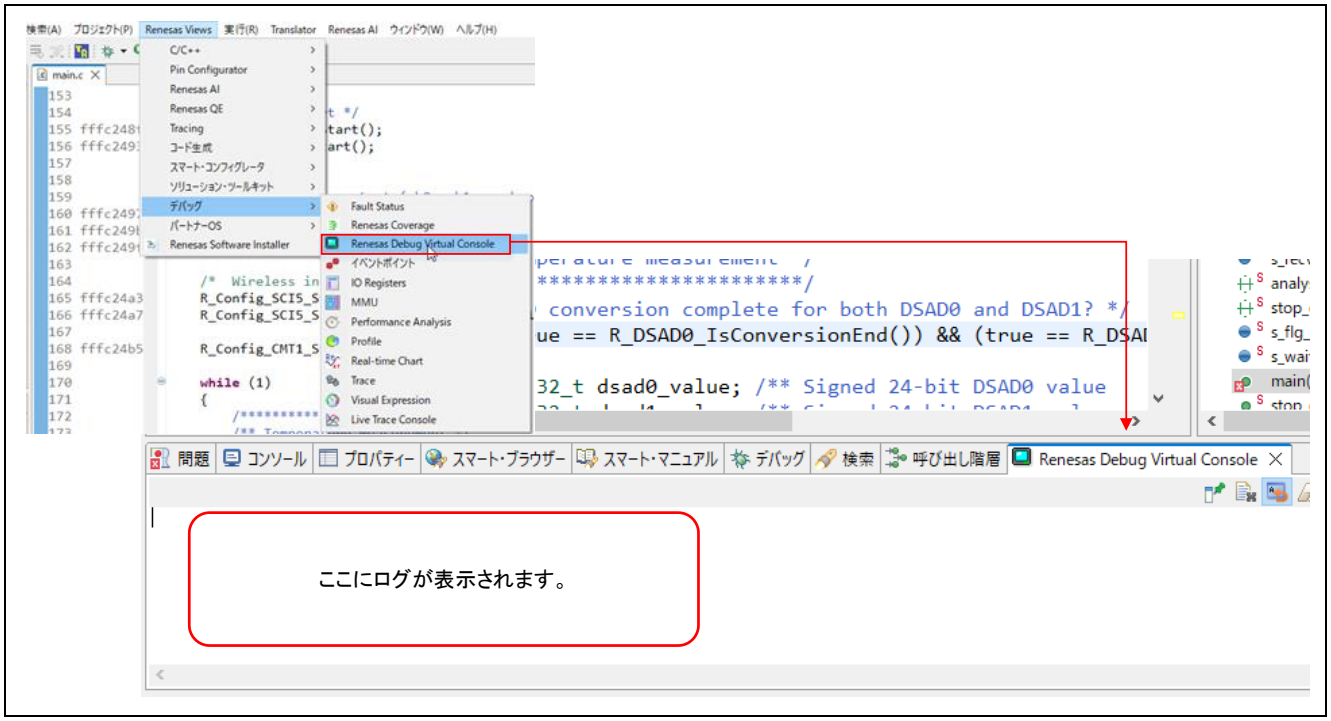

図 4-3　Renesas Debug Virtual Console

4.1.5　実行

1.  タンをクリック、または「F8」キー入力でデモプロジェクトを実行します。

その他、デバッグ画面の操作方法は以下ユーザーズマニュアルの 5.4 章を参照してください。

―・統合開発環境 e2studio ユーザーズマニュアル 入門ガイド (R20UT4204)

4.2　CS+ での手順

CS+ でご使用になる際は、以下の手順で CS+ にインポートしてください。

なお、CS+で管理するプロジェクトのフォルダ名、およびそのフォルダに至るファイルパスには、 空白文字の他、半角カナ文字、全角文字、半角記号(特に'$','#','%') が混じらないようにしてください。

（使用する CS+ のバージョンによっては画面が異なる場合があります。）

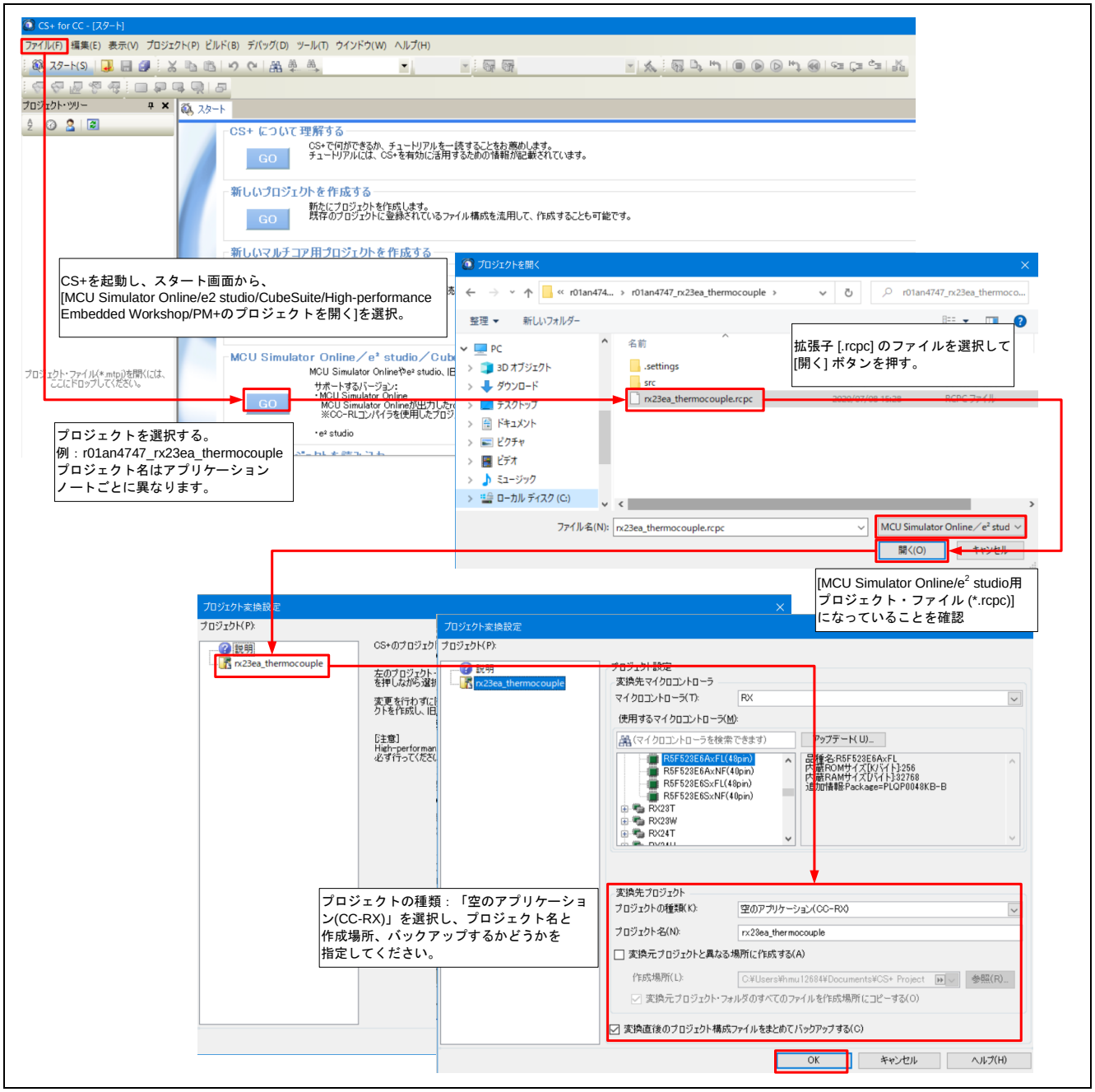

図 4-4　プロジェクトを CS+ にインポートする方法

4.2.1　プロジェクトのビルド

1. 「ビルド」→「ビルド・プロジェクト」を選択します。
2. ビルドが開始され、「コンソール」にビルドの状況が表示されるので”ビルド終了”というメッセージが表示されたらビルド完了です。

4.2.2　デバッグ

1. RX シミュレータを右クリックし、「使用するデバッグツール」→「RX E2 Lite」を選択します。
2.表示されたプロパティ画面が図 4-5 の通りであることを確認します。

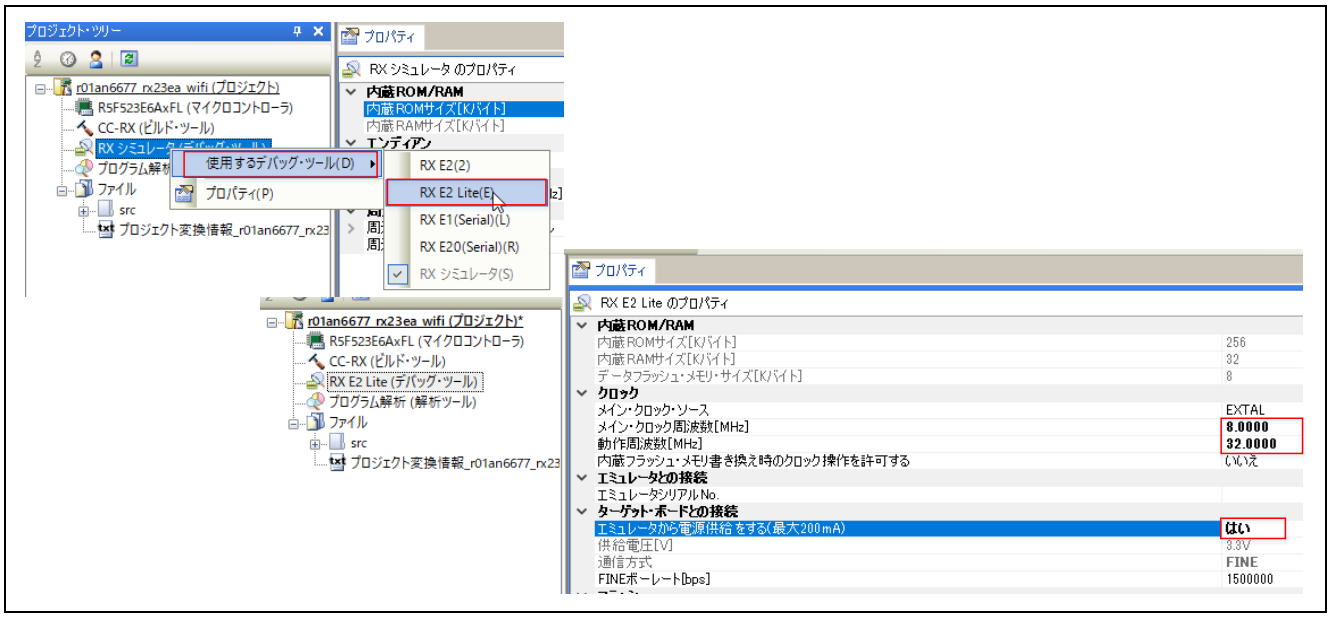

図 4-5　デバッグ設定

4.2.3　実行

1.  ボタンをクリックして、プログラムをダウンロードします。
2. 「Renesas Views」→「デバッグ」→「Renesas Debug Virtual Console」を選択します。
3.  ボタンをクリックして、プログラムを実行します。

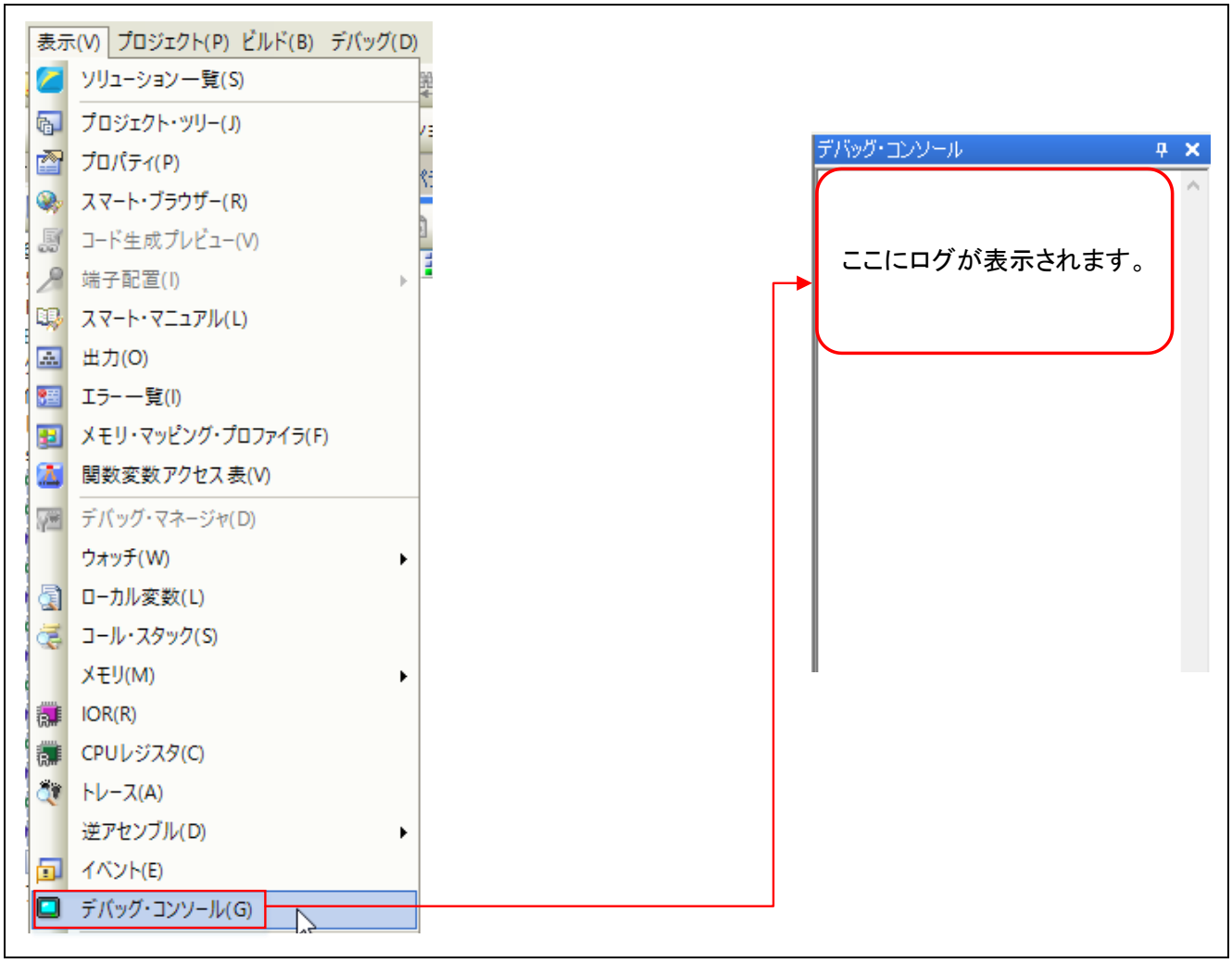

図 4-6　Renesas Debug Virtual Console(CS+)

5.　トラブルシューティング

5.1　AWS に接続できない

AWS に接続できない原因は様々です。主な原因は、「AWS のエンドポイントが間違っている」や「証明書やプライベートキーが間違っている」などが挙げられます。これらを正確に確認するためには、DA16600 Pmod™ Board のデバッグポート経由でログを確認します。ここではログの確認方法、およびいくつかのトラブルシュートの方法を示します。

5.1.1　DA16600 Pmod™ Board のログを確認する方法

5.1.1.1　必要部品

DA16600 Pmod™ Board のログを確認するためには、以下のモジュールが必要です。

- Pmod USBUART（DIGILENT 製）
  https://digilent.com/reference/pmod/pmodusbuart/start?redirect=1

5.1.1.2　接続方法

Pmod USBUART と DA16600 Pmod™ Board を以下のように接続してください。RSSKRX23E-A の接続はそのままとしてください。ただし、Pmod USBUART の接続を行う際には、RSSKRX23E-A の電源は切断してください。

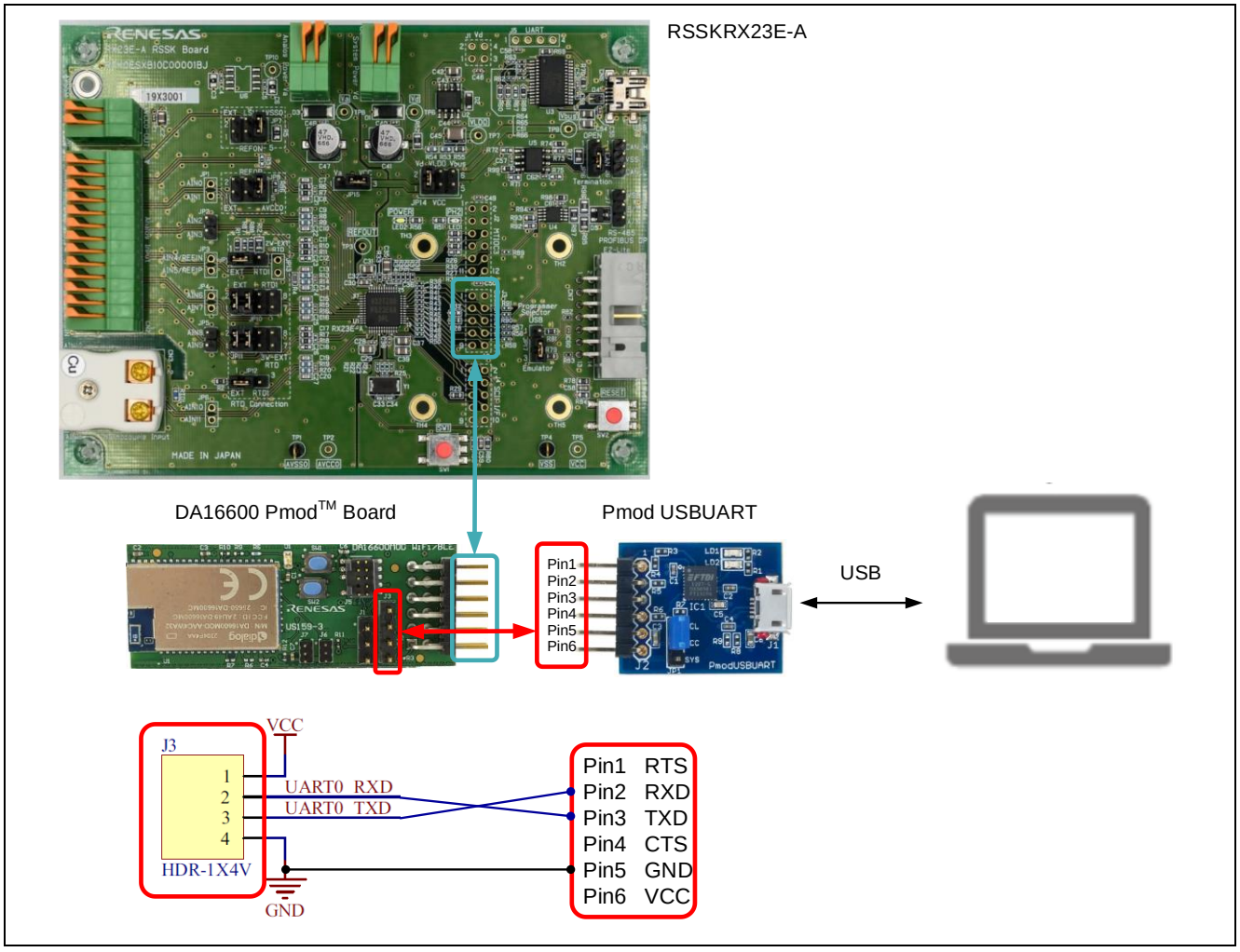

図 5-1　Pmod USBUART と DA16600 Pmod™ Board の接続方法

5.1.1.3　Tera Term を起動する

RSSKRX23E-A の電源を投入後、Tera Term を起動します。「Serial」にチェックを入れて、Pmod USBUART が接続されている COM ポートを選択してください。

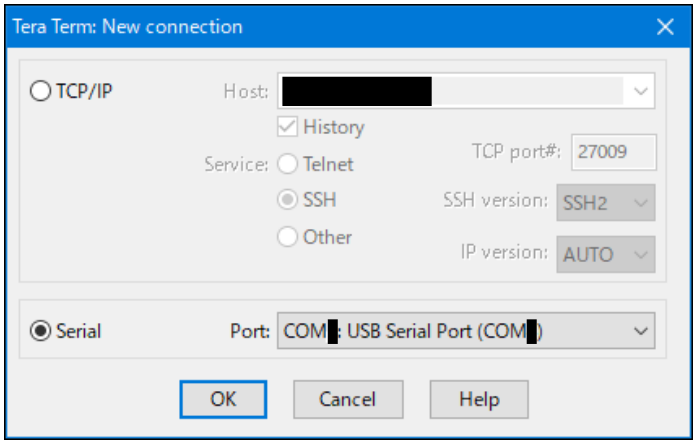

図 5-2　Tera Term の起動方法

5.1.1.4　Tera Term の Terminal を設定する

Tera Term のウインドウから、「Setup」 → 「Terminal setup」を開きます。以下のように設定してください。

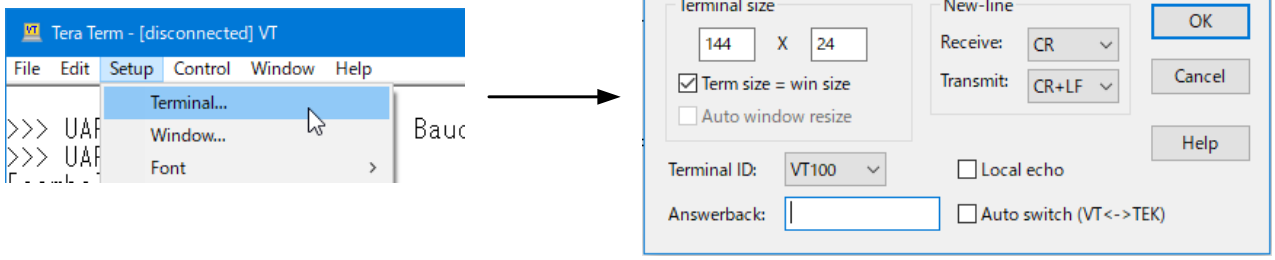

図 5-3　Tera Term の「Terminal setup」画面

5.1.1.5　Tera Term の Serial を設定する

Tera Term のウインドウから、「Setup」 →「Serial port setup and connection」を開きます。ボーレートは、DA16600 Pmod™ Board のデバッグポートの設定に合わせる必要があります。以下のように設定してください。

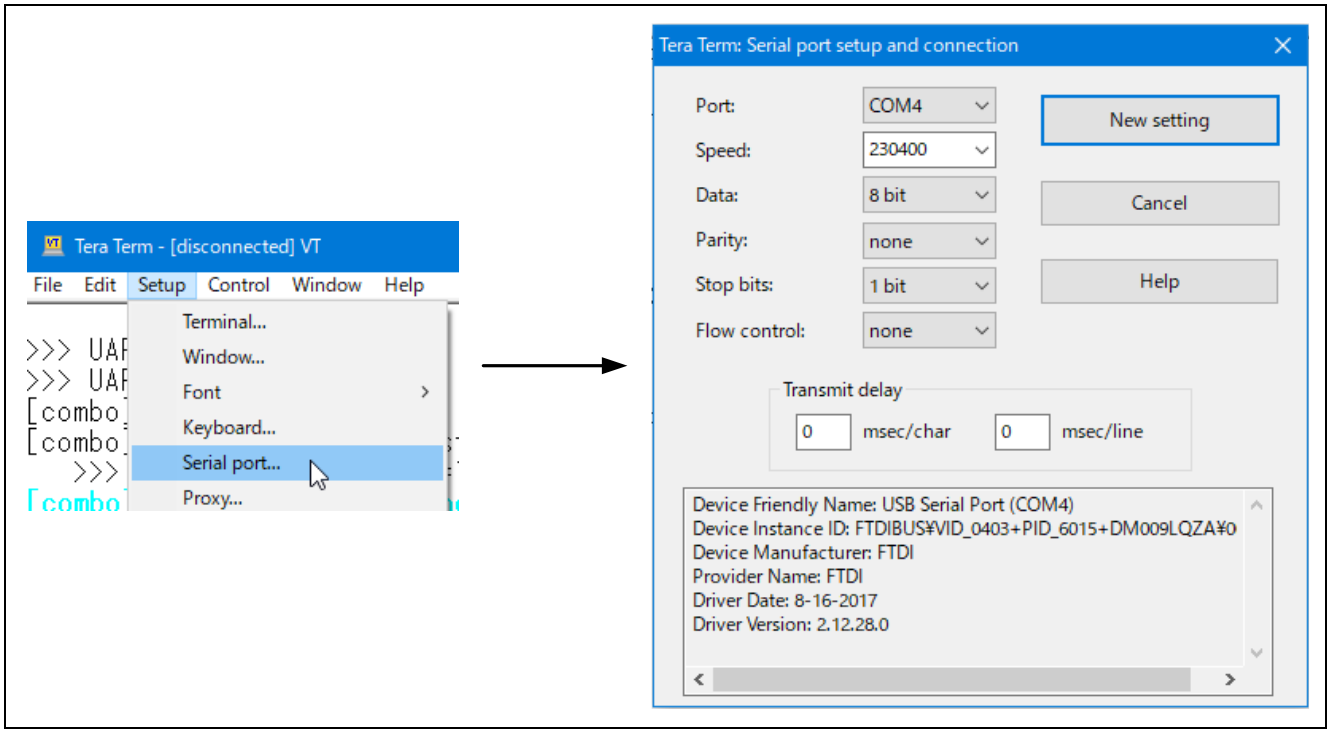

図 5-4　Tera Term の「Serial port setup and connection」画面

5.1.1.6　動作を確認する

RX23E-A の電源を入れ、正しくログが表示されることを確認してください。代表的なログを以下に示します。

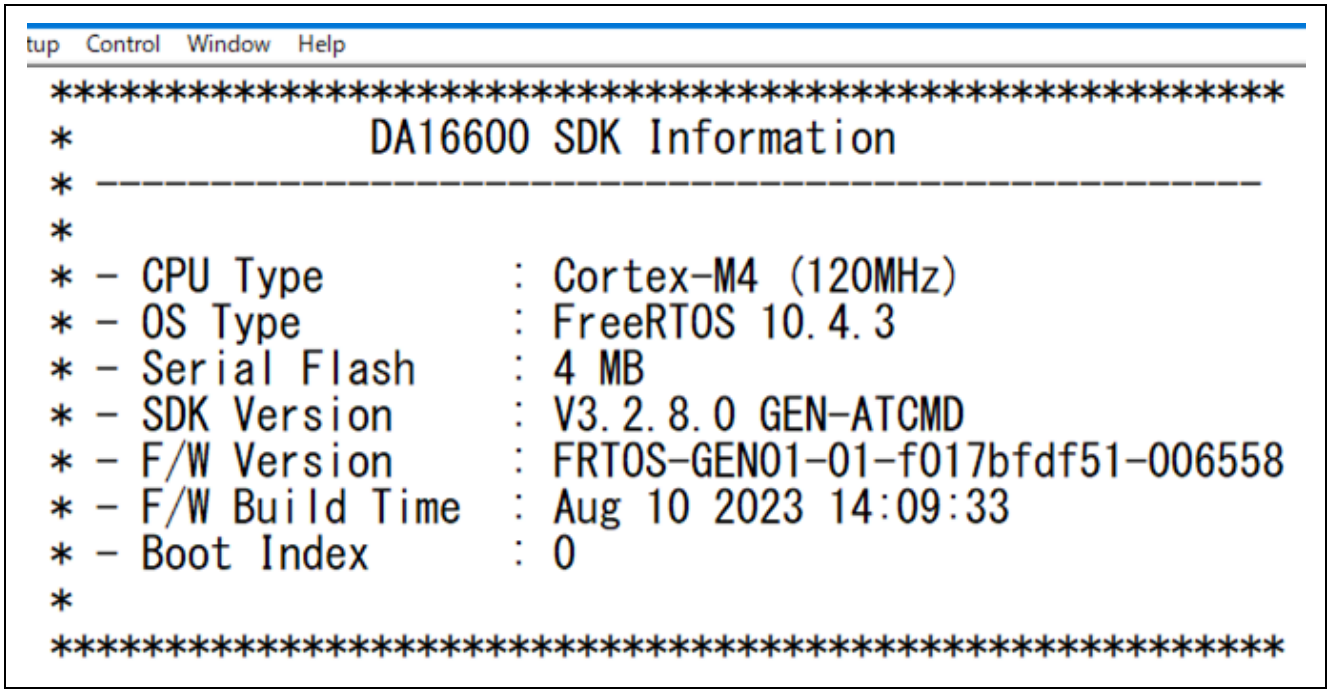

図 5-5　ログの例

5.1.2　DA16600 Pmod™ Board のファームウェアバージョンを更新する

本アプリケーションノートは、DA16600_IMG_FreeRTOS_ATCMD_UART2_EVK_v3.2.8.0_4 で動作を確認しています。ファームウェアのバージョンが異なる場合、以下に従ってファームウェアを更新してください。

5.1.2.1　ファームウェアをダウンロードする

以下に接続し、DA16200 DA16600 FreeRTOS SDK Image v3.2.8.0 をダウンロードします。

https://www.renesas.com/jp/ja/products/interface-connectivity/wireless-communications/wi-fi/low-power-wi-fi/da16600mod-ultra-low-power-wi-fi-bluetooth-low-energy-combo-modules-battery-powered-iot-devices#document

図 5-6　ファームウェアのダウンロード

5.1.2.2　ダウンロードしたファイルを解凍する

ダウンロードした zip ファイルを、任意のフォルダに解凍してください。解凍後、さらに DA16600_IMG_FreeRTOS_ATCMD_UART2_EVK_v3.2.8.0_4MB.zip を解凍してください。

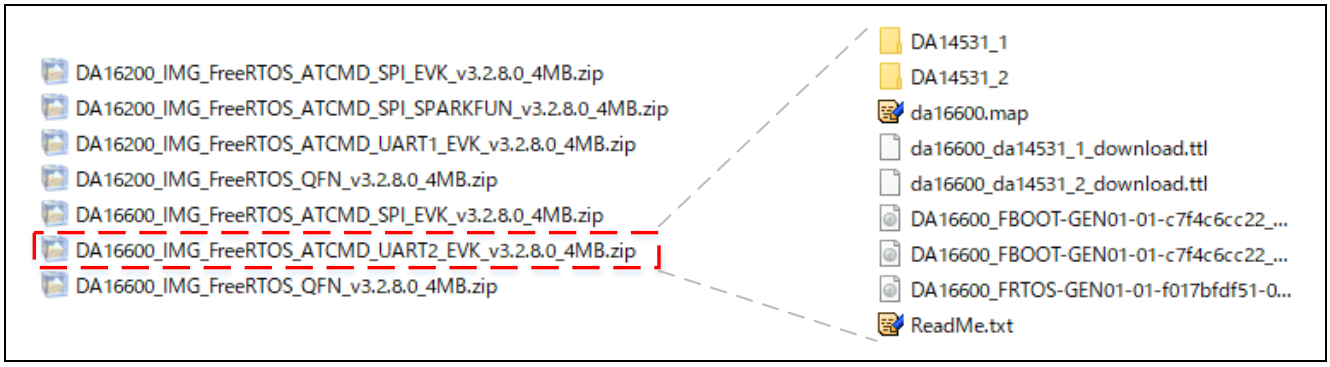

図 5-7　ファームウェアの解凍

5.1.2.3　Pmod USBUART を接続する

Pmod USBUART と DA16600 Pmod™ Board を以下の通り接続してください。

DA16600 Pmod™ Board の Pmod 端子は、全てオープンとしてください。Pmod USBUART と PC の接続は、最後に行ってください。

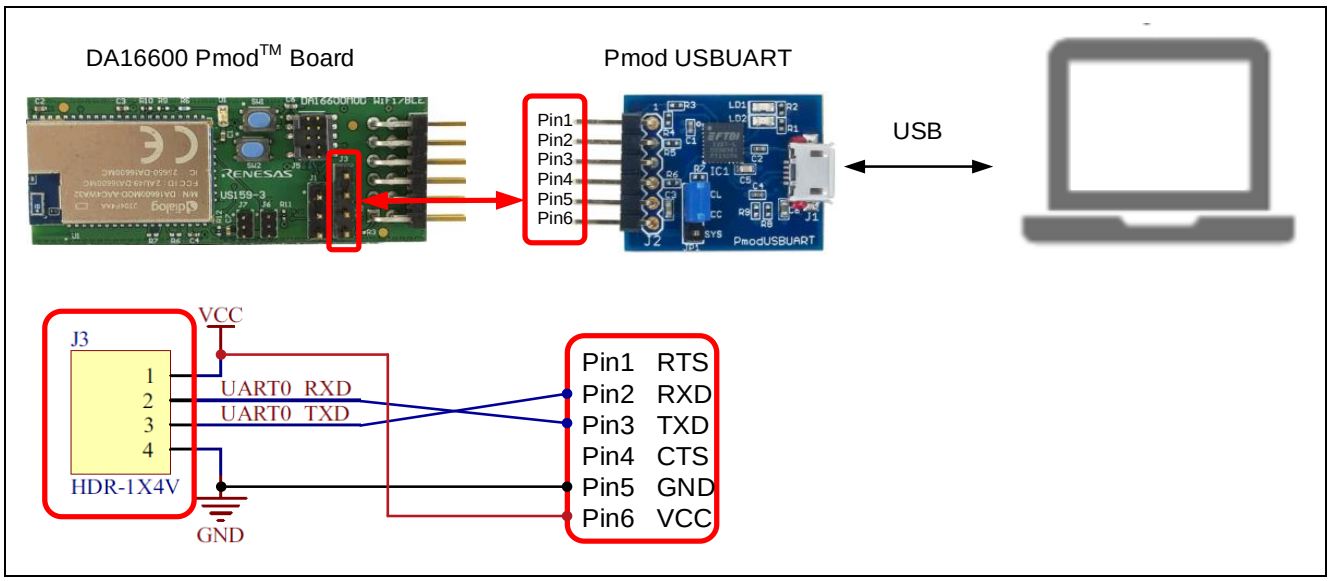

図 5-8　Pmod USBUART と DA16600 Pmod™ Board の接続方法

5.1.2.4　電源を供給する

Pmod USBUART と PC を USB で接続すると、Pmod USBUART と DA16600 Pmod™ Board に電源が供給されます。

5.1.2.5　Tera Term を起動する

RSSKRX23E-A の電源を投入後、Tera Term を起動します。「Serial」にチェックを入れて、Pmod USBUART が接続されている COM ポートを選択してください。

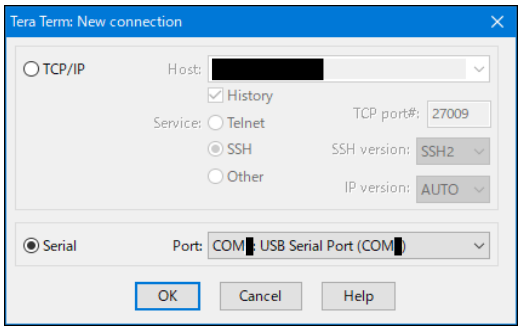

図 5-9　Tera Term の起動方法

5.1.2.6　Tera Term の Terminal を設定する

Tera Term のウインドウから、「Setup」 → 「Terminal setup」を開きます。以下のように設定してください。

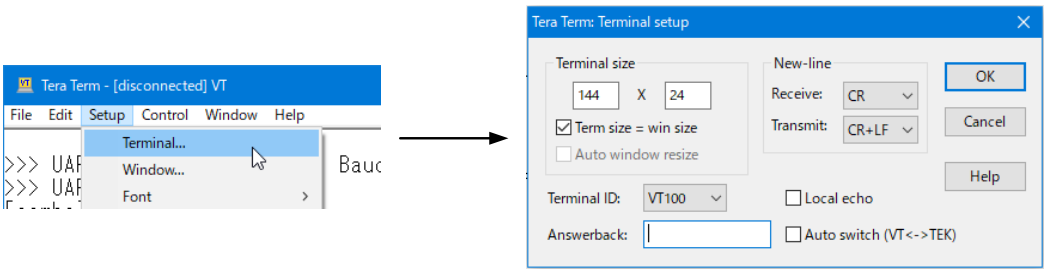

図 5-10　Tera Term の「Terminal setup」画面

5.1.2.7　Tera Term の Serial を設定する

Tera Term のウインドウから、「Setup」 → 「Serial port setup and connection」を開きます。ボーレートは、DA16600 Pmod™ Board のデバッグポートの設定に合わせる必要があります。以下のように設定してください。

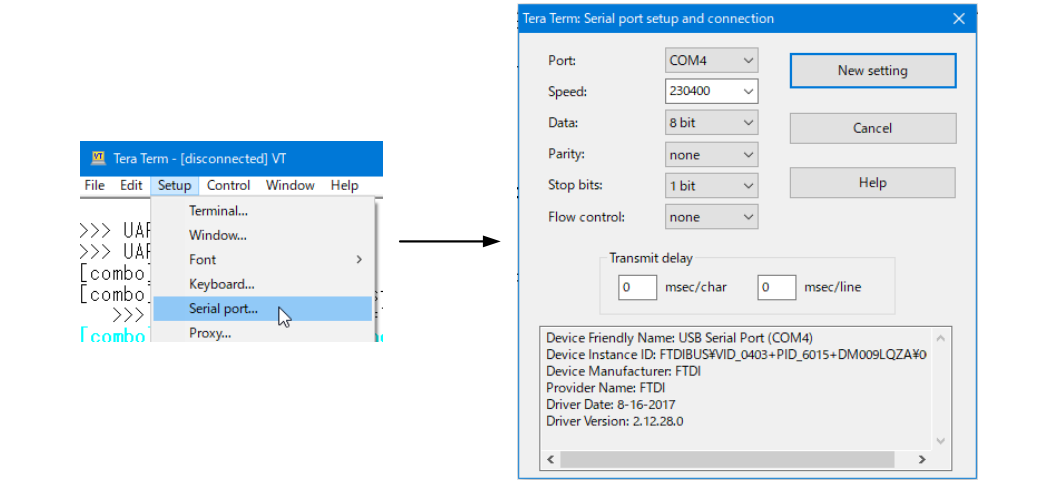

図 5-11　Tera Term の Serial port setup and connection 画面

5.1.2.8　ファームウェアを更新する

Tera Term のウインドウから、「Control」 → 「Macro」を開きます。「5.1.2.2 ダウンロードしたファイルを解凍する」で解凍したフォルダの「da16600_da14531_1_download.ttl」を選択し、「開く」をクリックします。続いて、「Confirm」画面で、「AT25SL321」を選択し、「OK」をクリックすると、Tera Term 上で、自動的にコマンドが実行され、ファームウェアのアップデートが行われます。

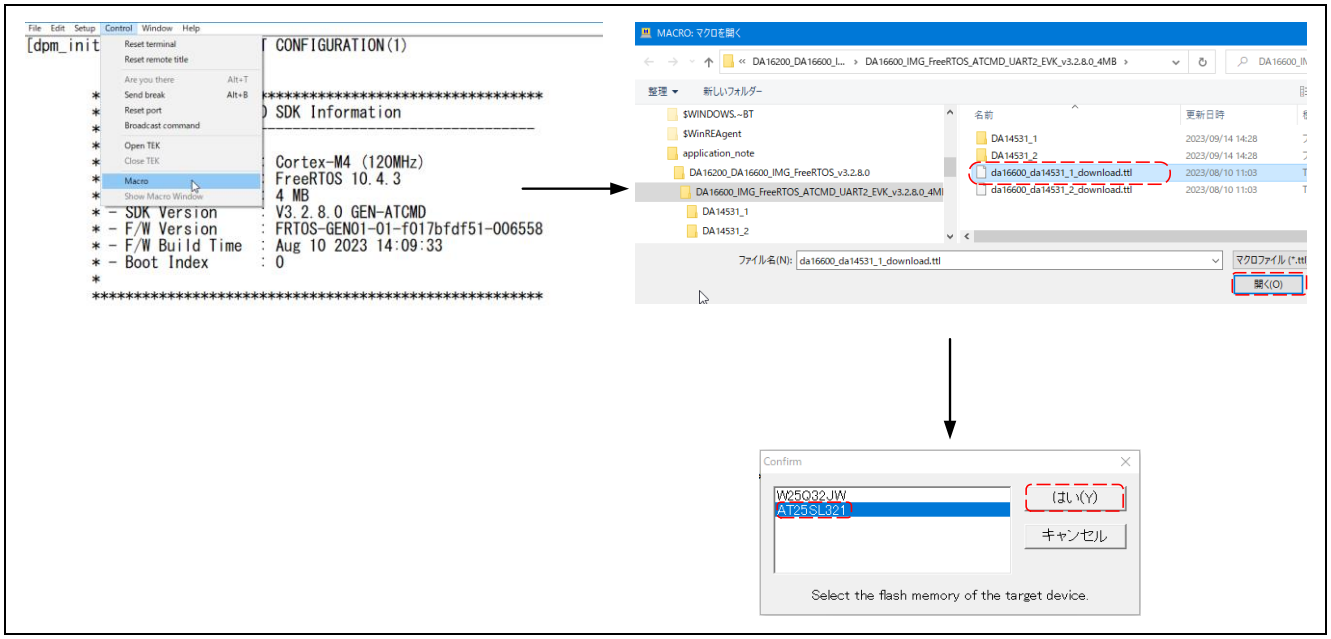

図 5-12　ファームウェアの更新

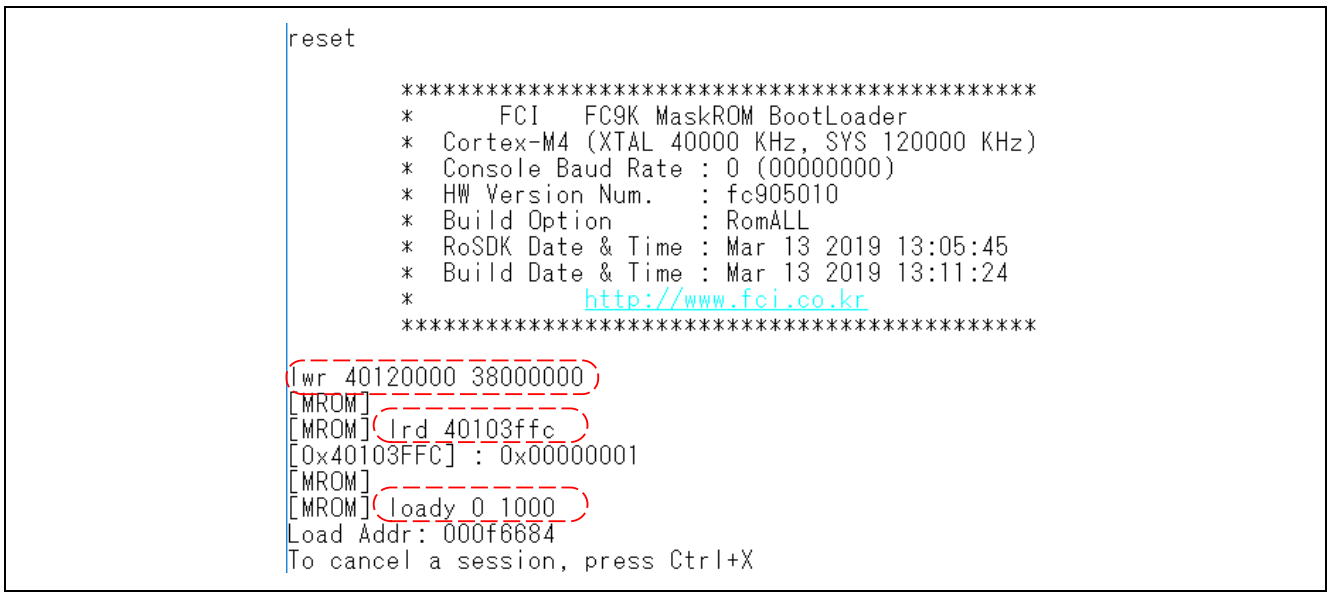

図 5-13　ファームウェアの更新中の Tera Term 画面

5.1.2.9　バージョンを確認する

ファームウェアがアップデートされた後、ファームウェアのバージョンが v3.2.8.0 になっていることを確認してください。

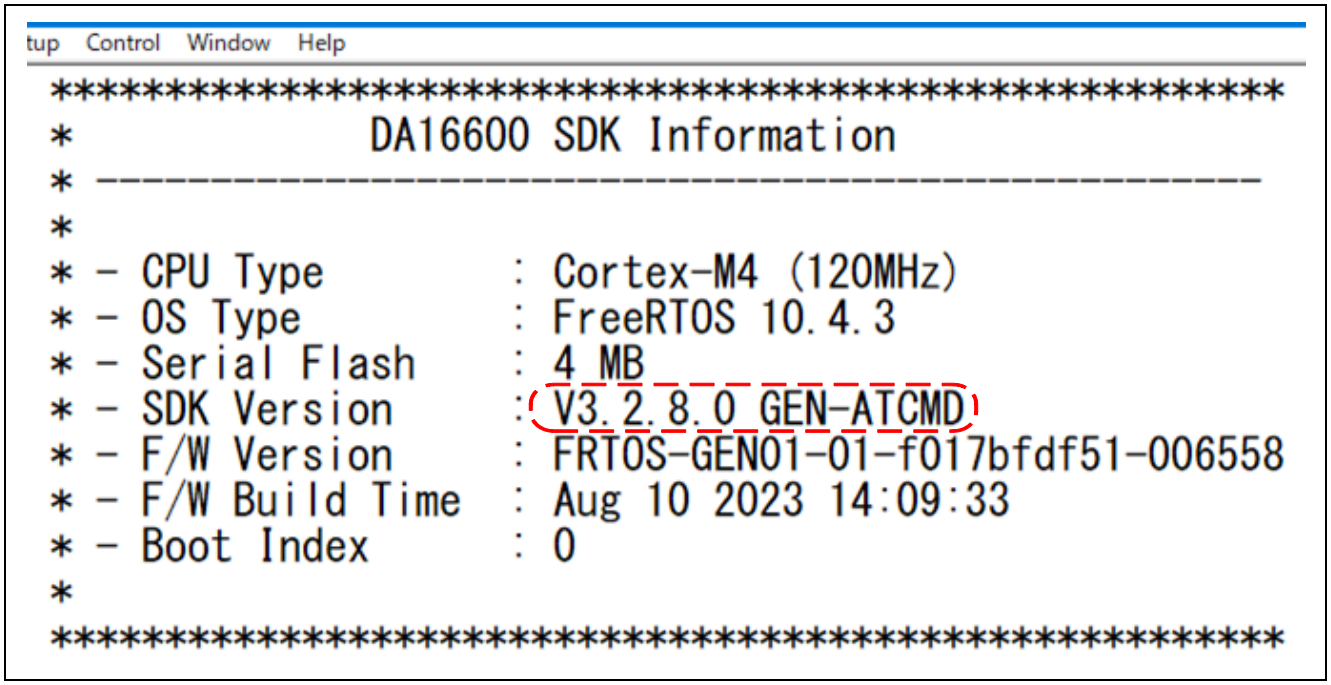

図 5-14　ファームウェアのバージョン確認

5.1.3　Fail to establish tls-sess(0x7200)が表示される場合

以下のように Fail to establish tls-sess(0x7200)が表示される場合について説明します。

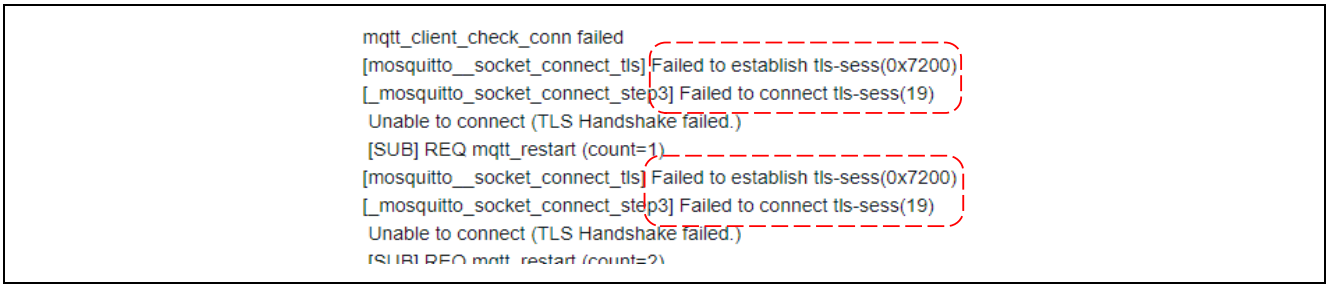

図 5-15　Fail to establish tls-sess(0x7200)が表示される場合

5.1.3.1　MQTT 用バッファを再設定する

Fail to establish tls-sess(0x7200)が表示される場合、MQTT 用のバッファを再設定します。以下のコマンドを実行し、再設定を行ってください。

```
setenv MQTT_TLS_INCOMING 16384
setenv MQTT_TLS_OUTGOING 16384
```

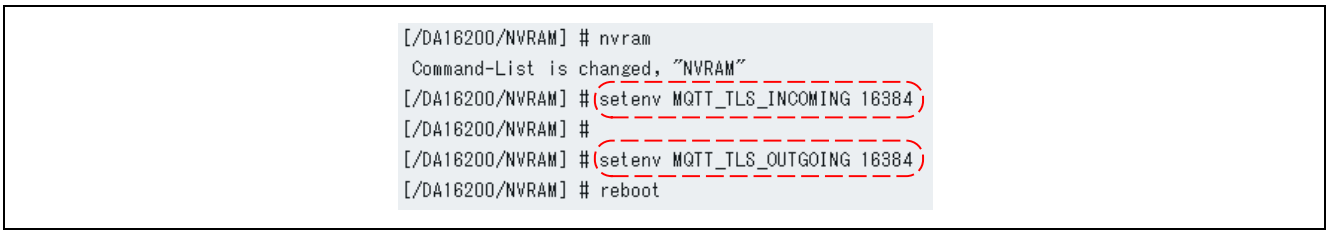

図 5-16　MQTT バッファ再設定

5.1.3.2　設定結果の確認する

以下のコマンドを実行し、再設定されていることを確認してください。

```
mqtt_config status
```

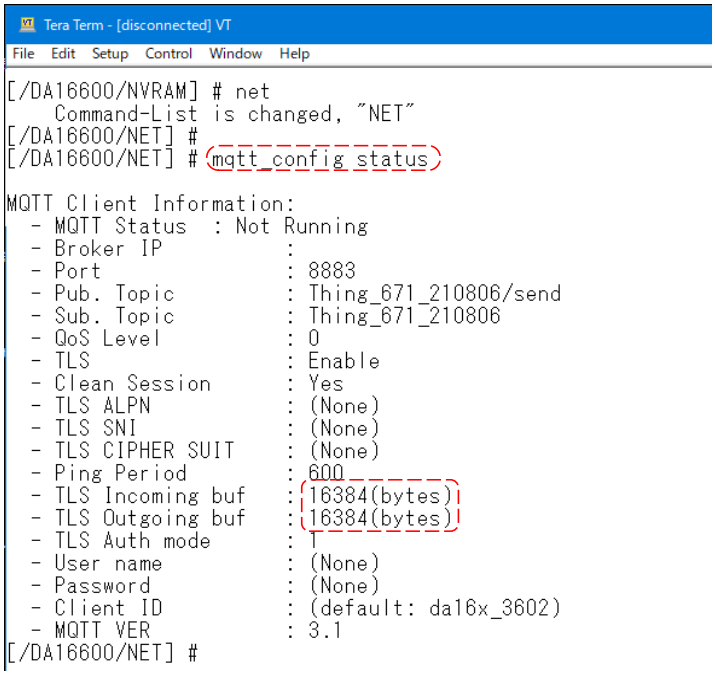

図 5-17　MQTT バッファ再設定確認結果

6. 参考資料

- RX23E-A グループ ユーザーズマニュアル ハードウェア編（R01UH0801）
- RSSKRX23E-A ユーザーズマニュアル（R20UT4542）
- RX23E-A グループ 熱電対を使用した温度計測例（R01AN4747）
- US159-DA14531EVZ Evaluation Board Manual（R15UZ0004）
- US159-DA 16600 EVZ Evaluation Board Manual（R15UZ0006）
- GATTBrowser for Windows Windows アプリケーション取扱説明書（R01AN6230）
- User Manual DA16200 DA16600 AT Command（UM-WI-003）
- AWS クラウドと FFT を応用した故障検知/動作解析デモンストレーション（R01AN5366）

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	Sep.27.23	ー	初版

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。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、お客様の責任において、お客様の機器・システムを設計ください。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. 当社製品または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. 当社製品を組み込んだ製品の輸出入、製造、販売、利用、配布その他の行為を行うにあたり、第三者保有の技術の利用に関するライセンスが必要となる場合、当該ライセンス取得の判断および取得はお客様の責任において行ってください。
5. 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。
6. 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。
7. あらゆる半導体製品は、外部攻撃からの安全性を 100%保証されているわけではありません。当社ハードウェア／ソフトウェア製品にはセキュリティ対策が組み込まれているものもありますが、これによって、当社は、セキュリティ脆弱性または侵害（当社製品または当社製品が使用されているシステムに対する不正アクセス・不正使用を含みますが、これに限りません。）から生じる責任を負うものではありません。当社は、当社製品または当社製品が使用されたあらゆるシステムが、不正な改変、攻撃、ウイルス、干渉、ハッキング、データの破壊または窃盗その他の不正な侵入行為（「脆弱性問題」といいます。）によって影響を受けないことを保証しません。当社は、脆弱性問題に起因しまたはこれに関連して生じた損害について、一切責任を負いません。また、法令において認められる限りにおいて、本資料および当社ハードウェア／ソフトウェア製品について、商品性および特定目的との合致に関する保証ならびに第三者の権利を侵害しないことの保証を含め、明示または黙示のいかなる保証も行いません。
8. 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。
9. 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。
10. 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。
11. 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。
12. 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。
13. 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。
14. 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.5.0-1 2020.10)

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。