

Renesas RA ファミリ

クイックスタートガイド : EK-RA8D1 Modbus TCP

要旨

本書は、RA マイコン評価ボードを使用して Modbus 通信を評価するためのクイックスタートガイドです。Modbus プロトコルは、Modicon Inc. (Schneider Electric SA.) がプログラマブル ロジックコントローラー (PLC) 向けに開発した通信プロトコルであり、その仕様は公開されています。詳細についてはプロトコル仕様書 (PI-MBUS-300 Rev.J) を参照してください。

対象デバイス

EK-RA8D1

目次

1. 概要.....	2
1.1 略語/定義.....	2
1.2 参考資料.....	2
2. 特徴.....	3
3. サンプルアプリケーションの構成.....	4
3.1 FSP Drivers.....	4
3.2 Modbus プロトコルスタック.....	4
3.3 ModbusDemoApplication.exe.....	4
4. 動作環境.....	5
5. EK ボードの設定と接続.....	5
6. Modbus サンプルプログラムの実行.....	6
7. 評価ツールを用いた Modbus 通信デモ.....	10
7.1 IP アドレスの設定.....	10
7.2 Modbus デモアプリケーションの設定.....	11
7.2.1 デモアプリケーションの仕様.....	12
8. Appendix.....	13
8.1 Appendix A: DHCP Mode.....	13
8.2 Appendix B: User-defined function.....	14
8.2.1 Register callback function.....	14
8.2.2 Register function code.....	14
8.2.3 User-Defined Functions.....	14
8.3 Appendix C: デバッカ構成.....	16
改訂記録.....	20

1. 概要

本書は、EK-RA8D1 評価ボード上で動作する Modbus プロトコルスタックのドキュメントであり、プロトコルスタックを利用したアプリケーションを開発・実装するための機能概要、アプリケーションプログラミングインターフェース(API)、アプリケーションサンプルについて説明しています。
このパッケージは、イーサネットベースの Modbus TCP プロトコルをサポートします。

このクイック スタート ガイドでは次の内容が提供されます。

- ・ クイック スタート サンプル プロジェクトを実行する手順。
- ・ フレキシブル ソフトウェア パッケージ (FSP) および e² studio 統合開発環境 (IDE) を使用して、クイック スタート サンプル プロジェクトをインポート、変更、および構築する手順。
- ・ マスターとの接続と簡単なデモの操作手順

1.1 略語/定義

表 1.1 略語/定義

Index	Abbreviations /Definitions	Description
1	IP	Internet Protocol
2	TCP	Transmission Control Protocol
3	USB	Universal Serial Bus
4	PC	Personal Computer
5	SW	Switch
6	EWARM	Embedded Workbench® for ARM
7	LED	Light Emitting Diode
8	TCP	Transmission Control Protocol
9	Wireshark	Free packet capture tool to check packets flowing on LAN

1.2 参考資料

Modbus に関する技術情報は Modbus 組織サイトから入手でき、EK-RA8D1 に関する情報はルネサスから入手できます。

表 1.2 技術資料

Index	Technical Inputs
1	r01ds0416xx0110-ra8d1.pdf
2	r01uh0995xx0110-ra8d1.pdf
3	r20ut5205eg0100_ek-ra8d1_user_manual.pdf
4	Modicon Modbus Protocol Reference Guide Rev.J
5	Modbus Application Protocol Specification V1.1b3

2. 特徴

EK-RA8D1 の Modbus プロトコルスタックを使用することにより、Modbus TCP アプリケーションの迅速かつ簡単な開発が可能になります。

初期化 API がサポートする Modbus 関数コードを指定します。このスタックには次の 9 つの機能コードが実装できます。

- 1(0x01) – Read coils
- 2(0x02) – Read discrete input
- 3(0x03) – Read holding registers
- 4(0x04) – Read input registers
- 5(0x05) – Write single coil
- 6(0x06) – Write single register
- 15(0x0F) – Write multiple coils
- 16(0x10) – Write multiple registers
- 23(0x17) – Read/Write multiple registers

Modbus について詳細は以下のサイトをご覧ください。

<http://www.modbus.org>

* アップデートによりバージョン番号が異なる場合があります。最新のマニュアルをご参照ください。

3. サンプルアプリケーションの構成

このサンプル アプリケーションは 3 つのブロックで構成されています

- FreeRTOS+TCP を含む FSP ドライバー
- RTOS と TCP/IP スタックを使用した Modbus プロトコルスタックのサンプルプログラム
- Modbus プロトコルスタックを使用したアプリケーションサンプルプログラム

3.1 FSP Drivers

- e2studio/ra
このディレクトリには、以下の Modbus プロトコルスタックに必要なコードが含まれています。
 - BSP (Board support package) for EK-RA8D1,
 - HAL (Hardware abstraction layer) drivers,
 - FreeRTOS and FreeRTOS+TCP.

FreeRTOS は、マイクロコントローラー用のリアルタイムオペレーティング システム (RTOS) のオープンソースソフトウェアです。
詳細については、以下のリンクを参照してください。

<https://www.freertos.org/index.html>

<https://aws.amazon.com/freertos/>

- e2studio/ra_gen
このディレクトリには、以下の Modbus プロトコルスタックに必要なコードが含まれています。
 - HAL (Hardware abstraction layer) drivers,
- e2studio/ra_cfg
このディレクトリには、以下の Modbus プロトコルスタックに必要なコードが含まれています。
 - BSP (Board support package) for EK-RA8D1,
 - FreeRTOS and FreeRTOS+TCP.

3.2 Modbus プロトコルスタック

- e2studio/Modbus
 - Modbus プロトコルスタックと TCP/IP 通信を実装しています。
 - このディレクトリ内のコードはタスク生成、消去を行います。
ユーザーはコードに触れる必要はありません。
- e2studio/Modbus_user
 - ユーザーは、Modbus 関数コードの独自の実装を Modbus プロトコルスタックに登録できます。
 - このディレクトリ内のコードは、Modbus プロトコルスタックを初期化と、Modbus プロトコルスタック API を使用して、ユーザーが Modbus 関数コード実装を登録する方法に関する例です。

3.3 ModbusDemoApplication.exe

この実行ファイルは、TCP/IP 通信用の TCP デモサーバーです。
これを使用して、Modbus プロトコルスタックのデモを実行できます。

4. 動作環境

本書のサンプルプログラムは以下の環境で動作します。

表 4.1 動作環境

項目	概要
Board	EK-RA8D1 v1.0
CPU	Cortex®-M85
Operating frequency	CPU clock (CPUCLK): 480 MHz
Operating voltage	3.3 V
Device requirements	R7FA8D1AHECBD <ul style="list-style-type: none"> Flash memory Capacity: 2 MB SDRAM Capacity: 1 MB
Integrated development environment	e ² Studio 2023-10 以降
MCU software package	FSP(Flexible Software Package) v5.1.0
Emulator	J-LINK OB
Communications protocol	Modbus TCP
Master tool	ModbusDemoApplication.exe: Modbus 評価プログラム

5. EK ボードの設定と接続

以下に示すように、PC を EK-RA8D1 ボードに接続します。

電源は USB micro-B ケーブルをボードに接続することで供給されます。

Modbus TCP 通信の場合は RJ45 コネクタを使用し、LAN ケーブルで PC に接続します。

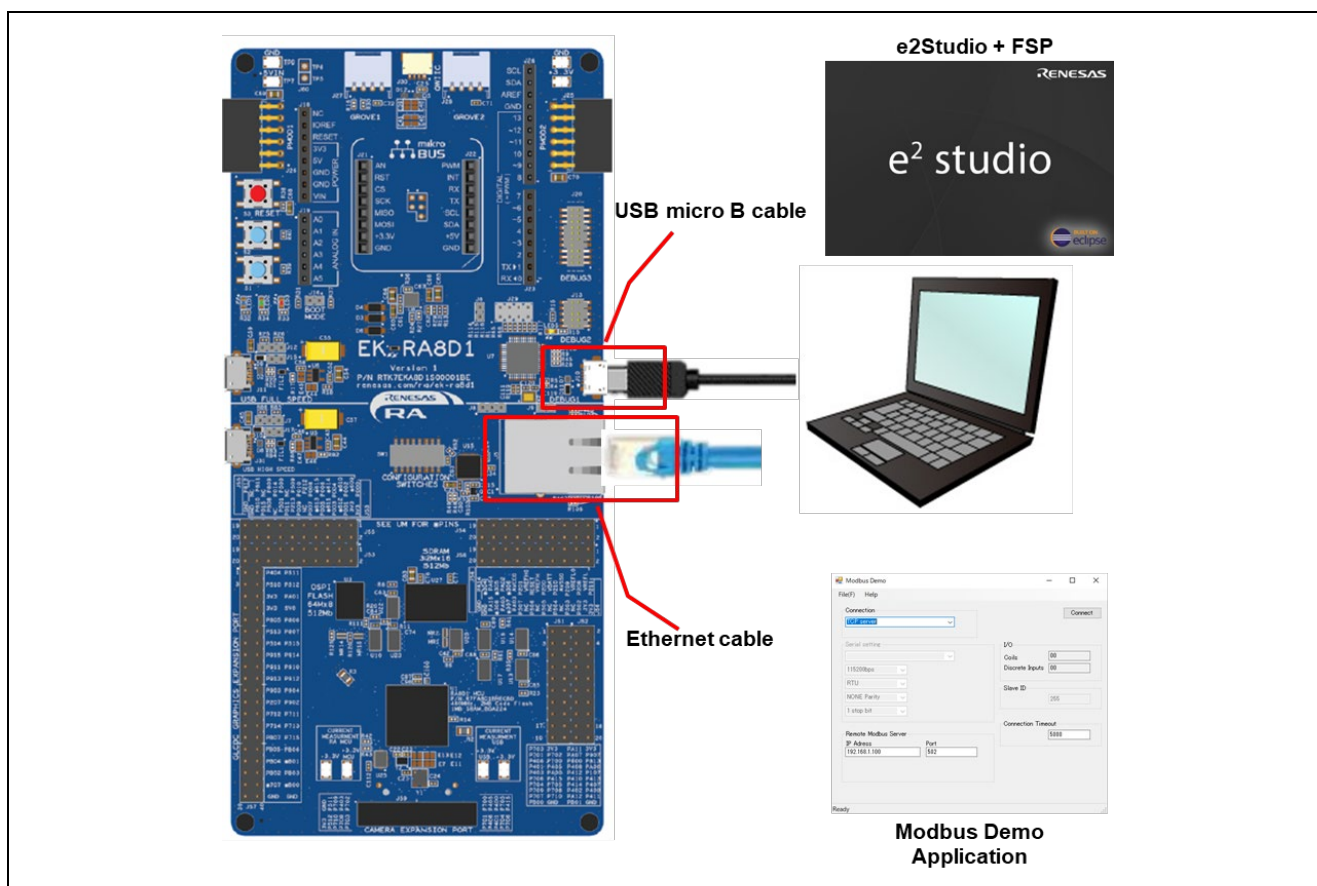


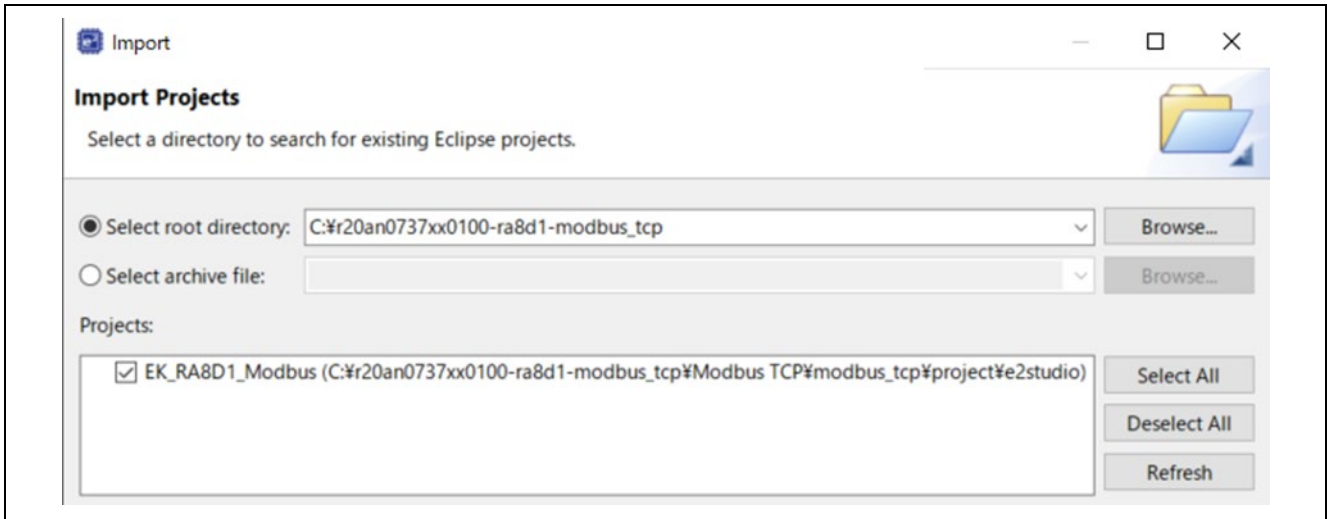
図 5: EK-RA8D1 board 接続設定

6. Modbus サンプルプログラムの実行

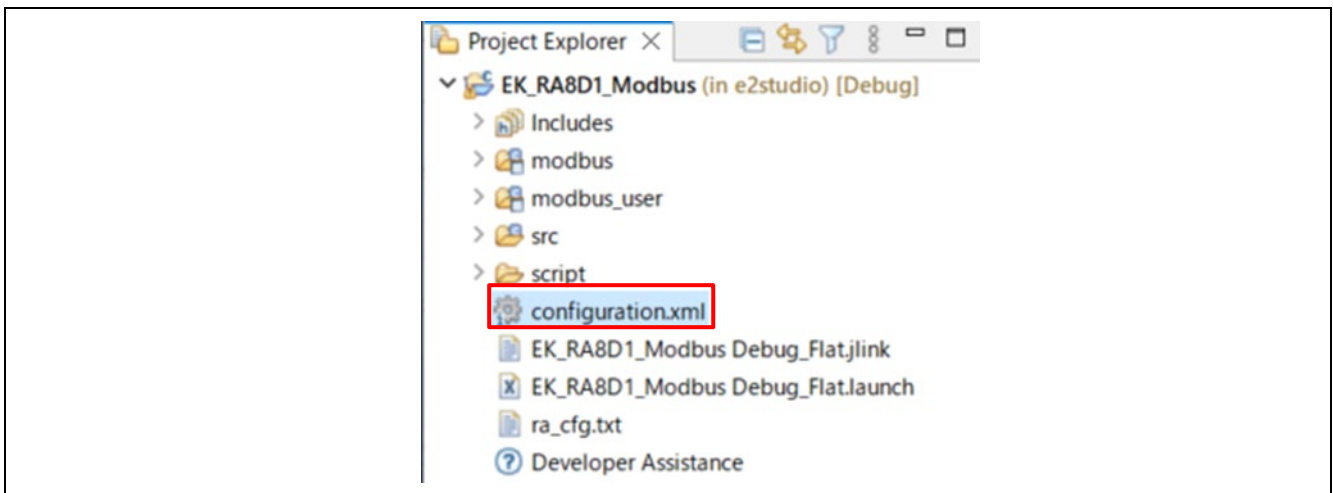
この章では、Modbus サンプルアプリケーションと通信する手順について説明します。

事前に「第4章 動作環境」、「第5章 ボードの設定と接続」を参照し、ツールのインストールとハードウェアの接続を完了してください。

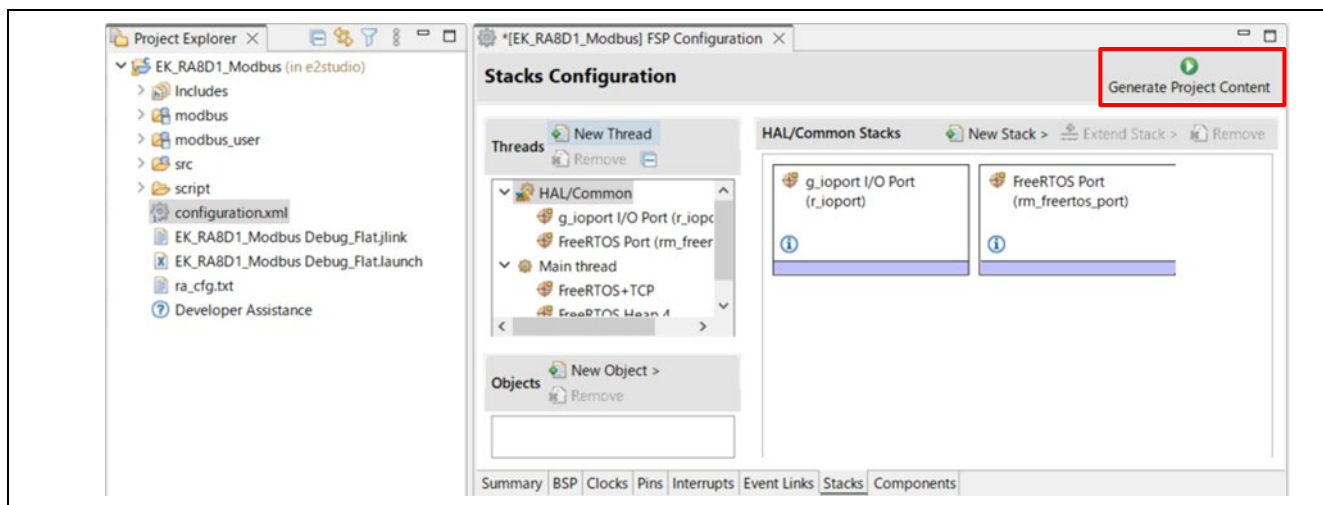
1. サンプルプロジェクトをインポートします。e² studio を起動し、[File] → [Import] → [Existing Projects into Workspace]の順で選択します。
"select root directory"にチェックを入れ、“r20an0737xx0100-ra8d1-modbus_tcp” folder → [Finish]を選択します。



2. 「EK_RA8D1_Modbus」プロジェクトの「configuration.xml」を開きます

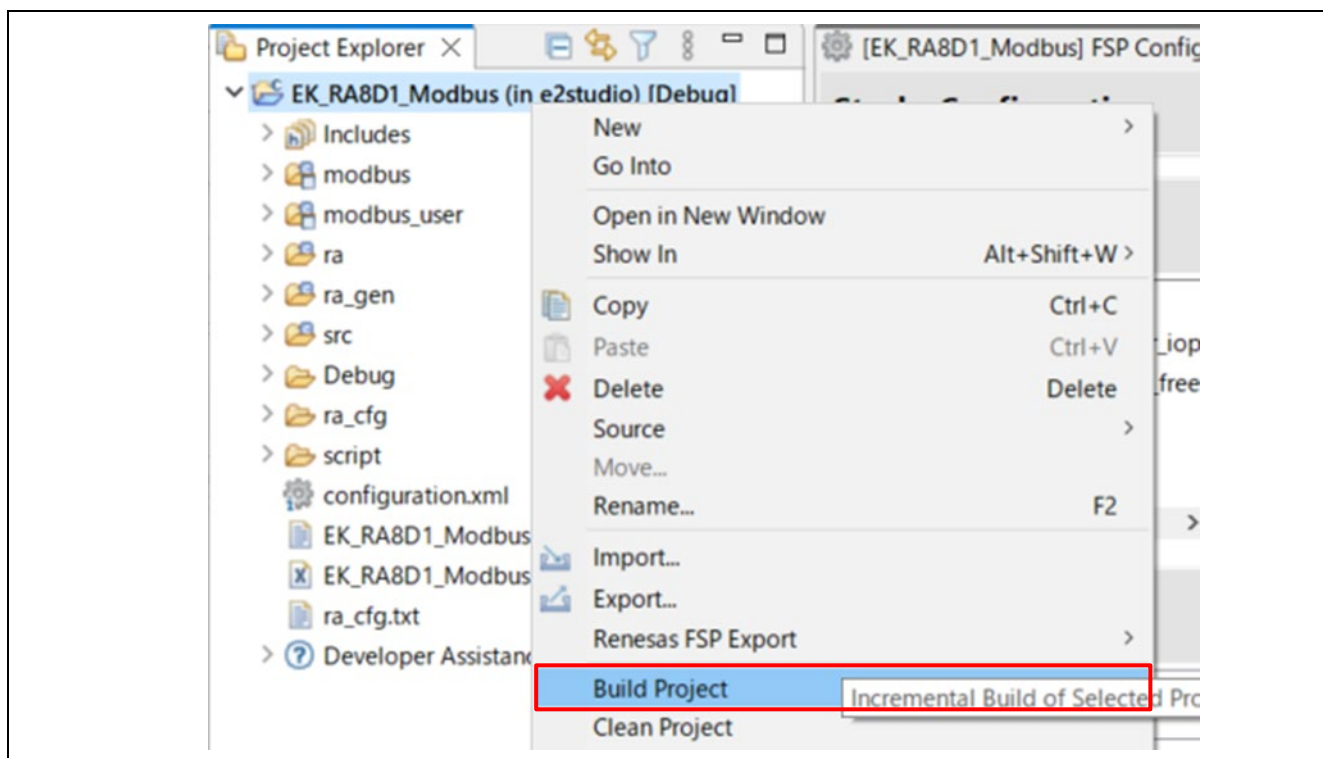


3. 「プロジェクトコンテンツの生成」でコードを生成します。



4. ビルドの実行

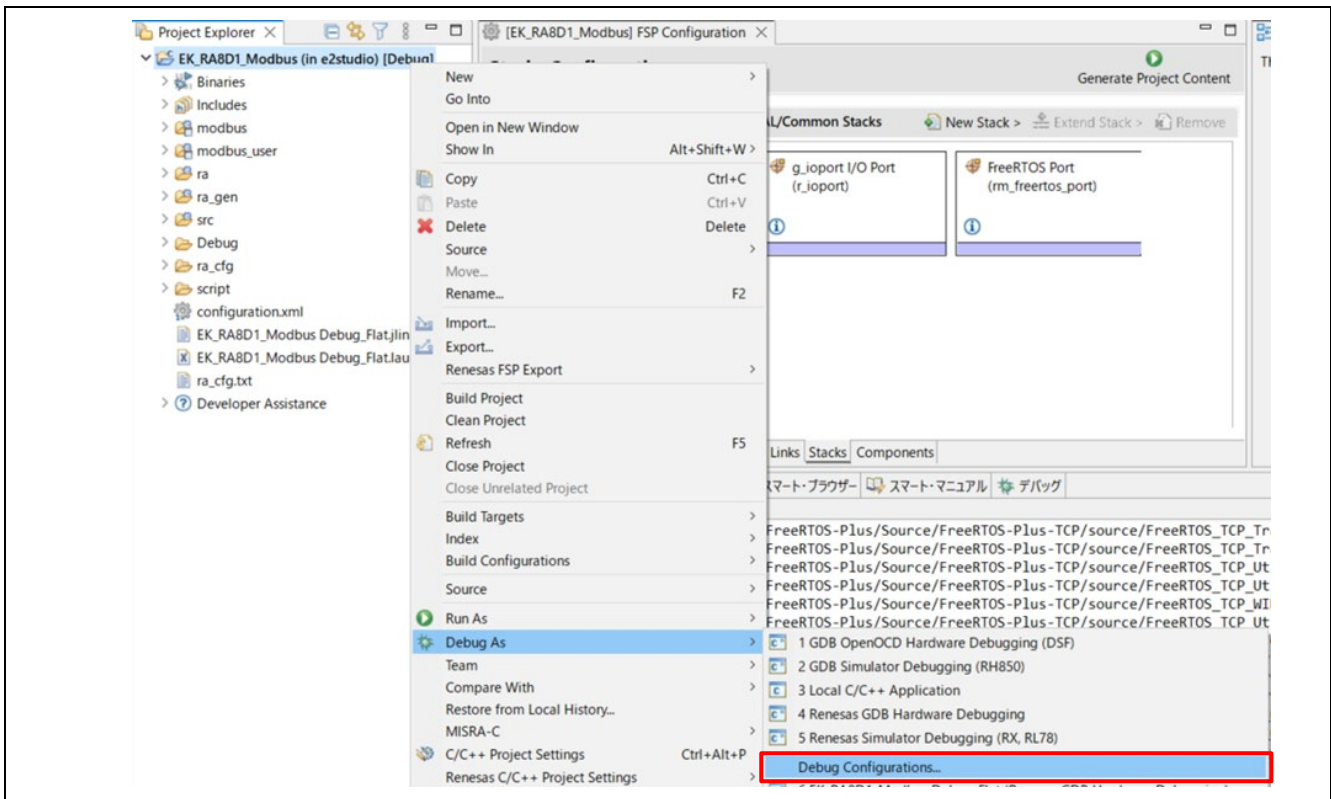
このとき、ビルドエラーがないことを確認してください。



5. アプリケーションのダウンロードとデバッガの実行

ボードと J-Link OB を接続後、以下の手順でデバッグを開始します。

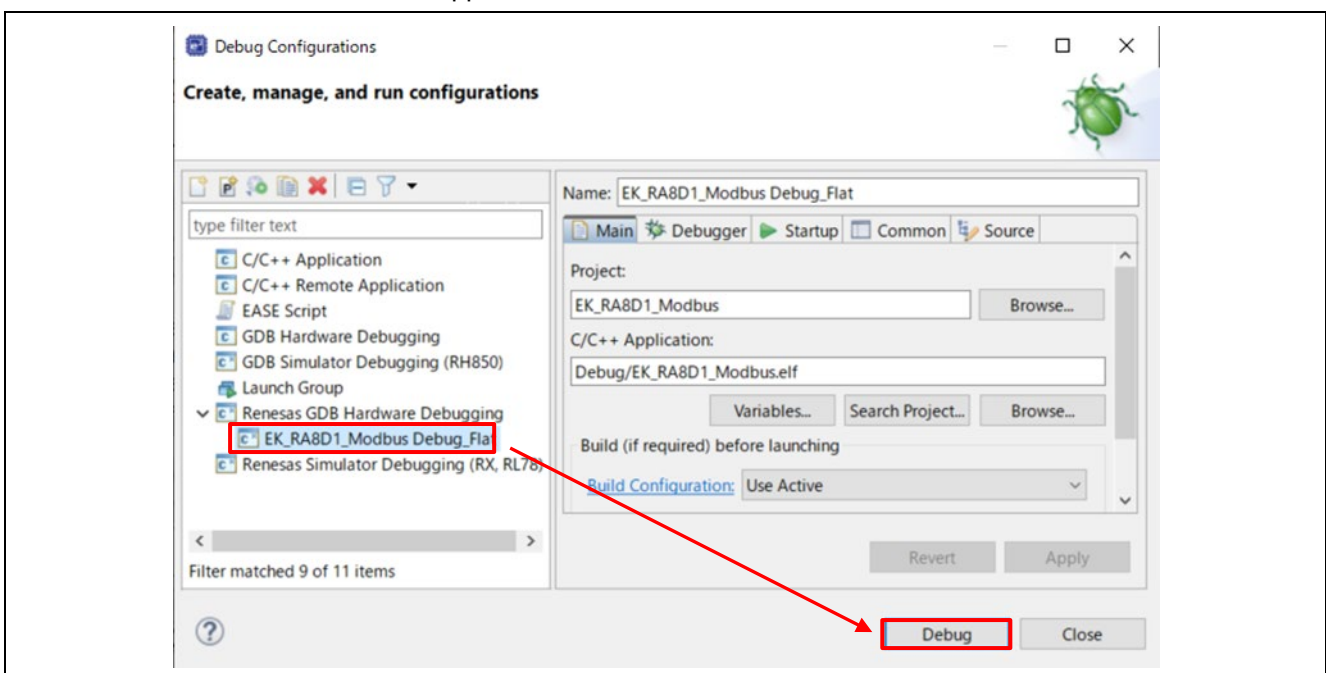
[Project Explorer]ビューで、デバッグするプロジェクトのノードを右クリックし、[Debug As]→[Debug Configurations]を選択します。



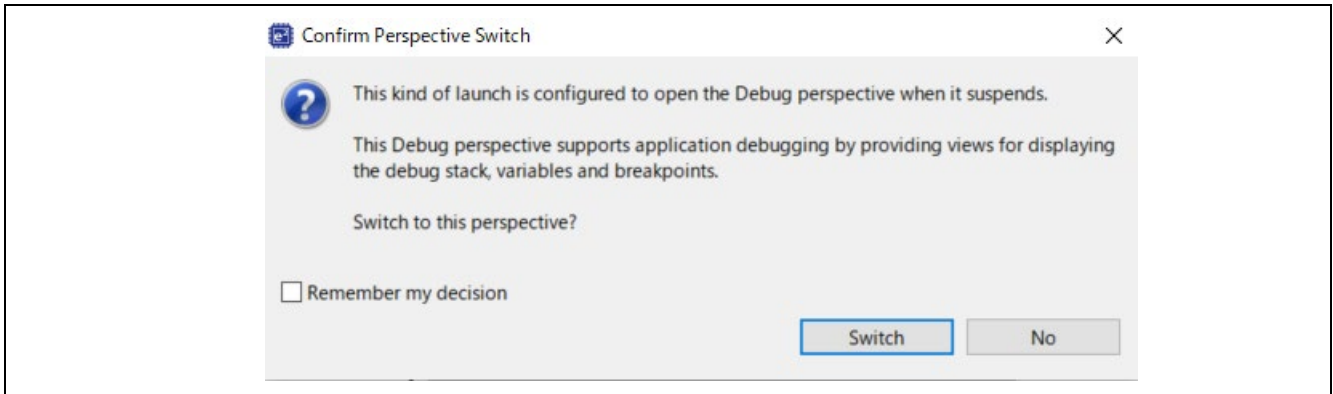
6. プログラムのダウンロード

[Renesas GDB Hardware Debugging] → [EK_RA8D1_Modbus Debug_Flat]を選択肢し、[Debug]を押します。

* デバッガの構成については、Appendix C で説明されています。



以下のダイアログが表示されるので、デバッグ画面に切り替えます。

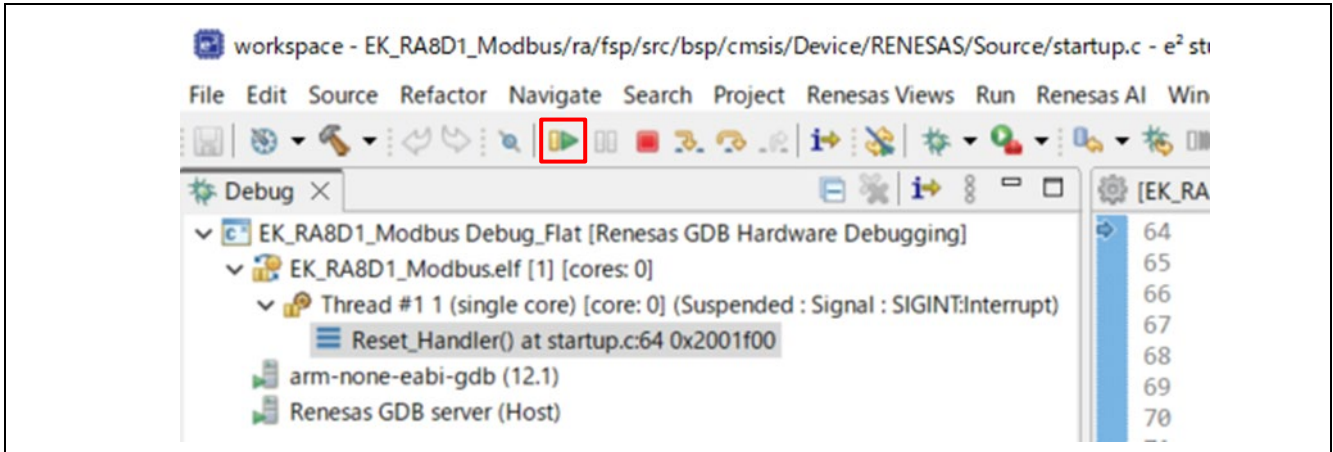


7. プログラムの開始

"Resume" ボタンを押します。

デバッグが開始されると、プログラムは「main.c;」で中断されます。

もう一度「Resume」ボタンを押してください。プログラムが実行されます。



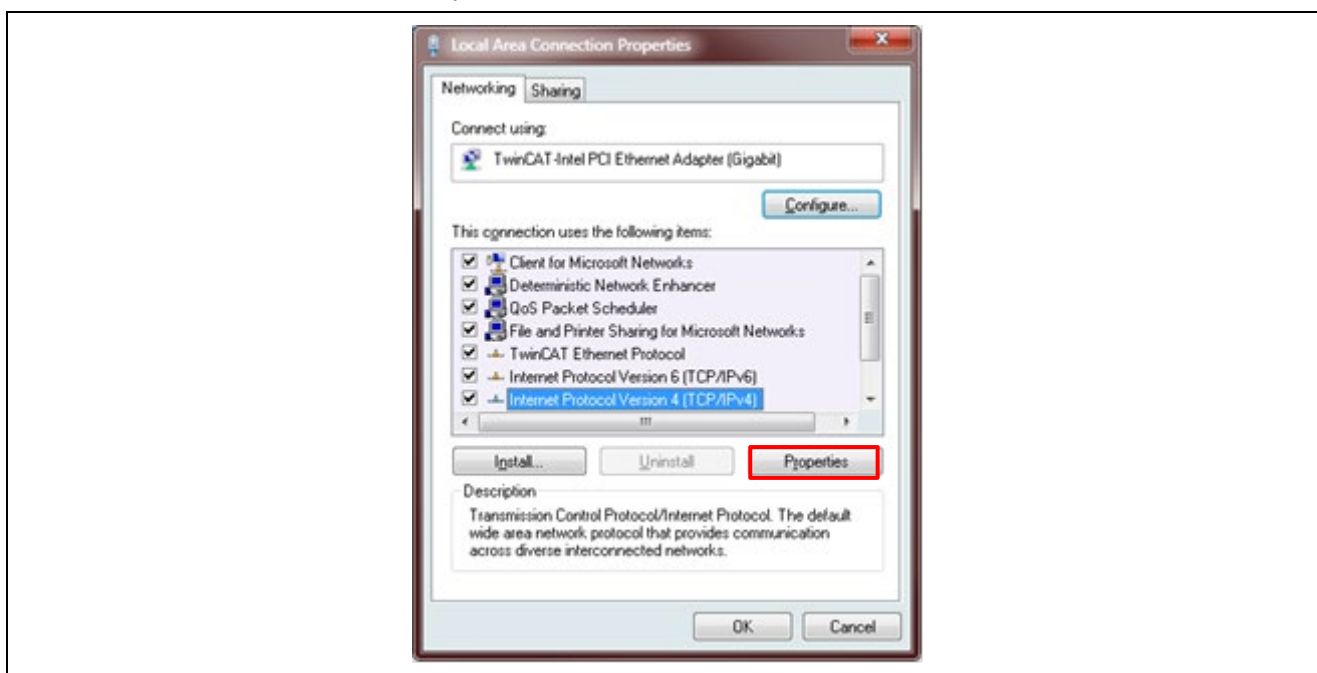
7. 評価ツールを用いた Modbus 通信デモ

この章では、評価ツール(ModbusDemoApplication.exe)を使用して、サンプルプログラムのデモ動作を確認する手順を説明します。

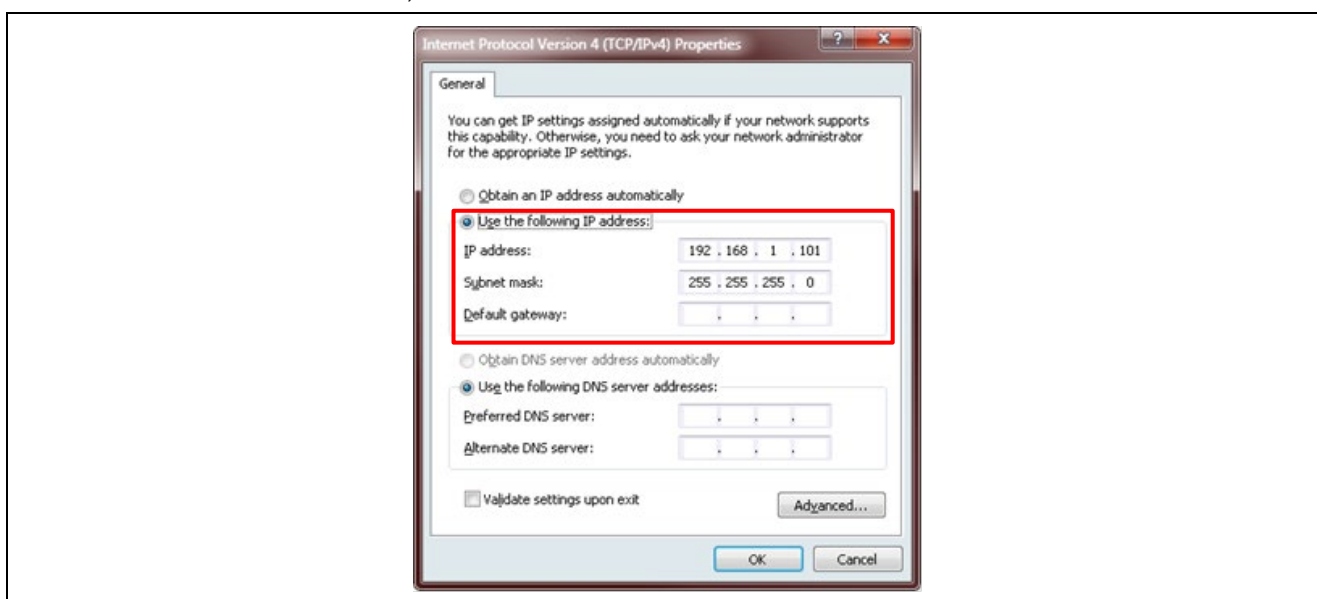
7.1 IP アドレスの設定

Modbusサンプルプログラムを実行するには、PCのIPアドレスを評価ボードと同じドメインに設定する必要があります。

1. ネットワーク接続リストを開きます。
[Control panel] → [Network and Sharing Center] → [Change adapter settings]
ローカルエリア接続をダブルクリック（または右クリック）し、[Properties] を選択します。
「TCP/IPv4」を選択し、「Properties」ボタンを押します。



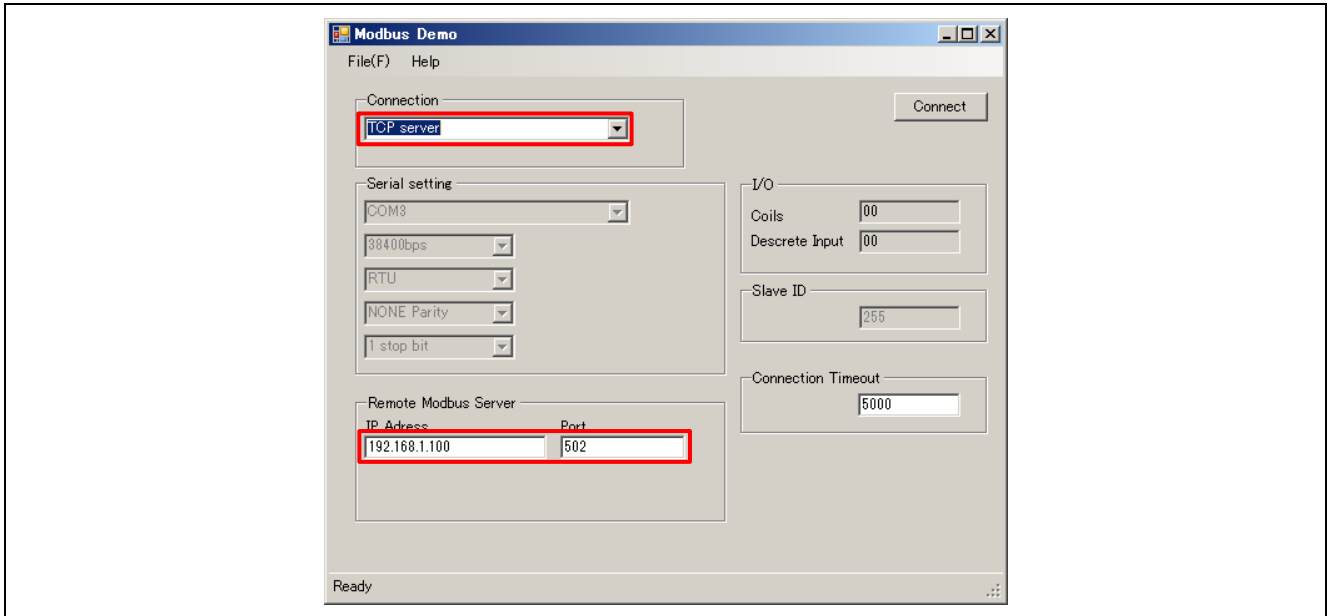
2. IP アドレスとサブネットマスクを設定します。
IP-address :192.168.1.101, Sub net mask 255.255.255.0



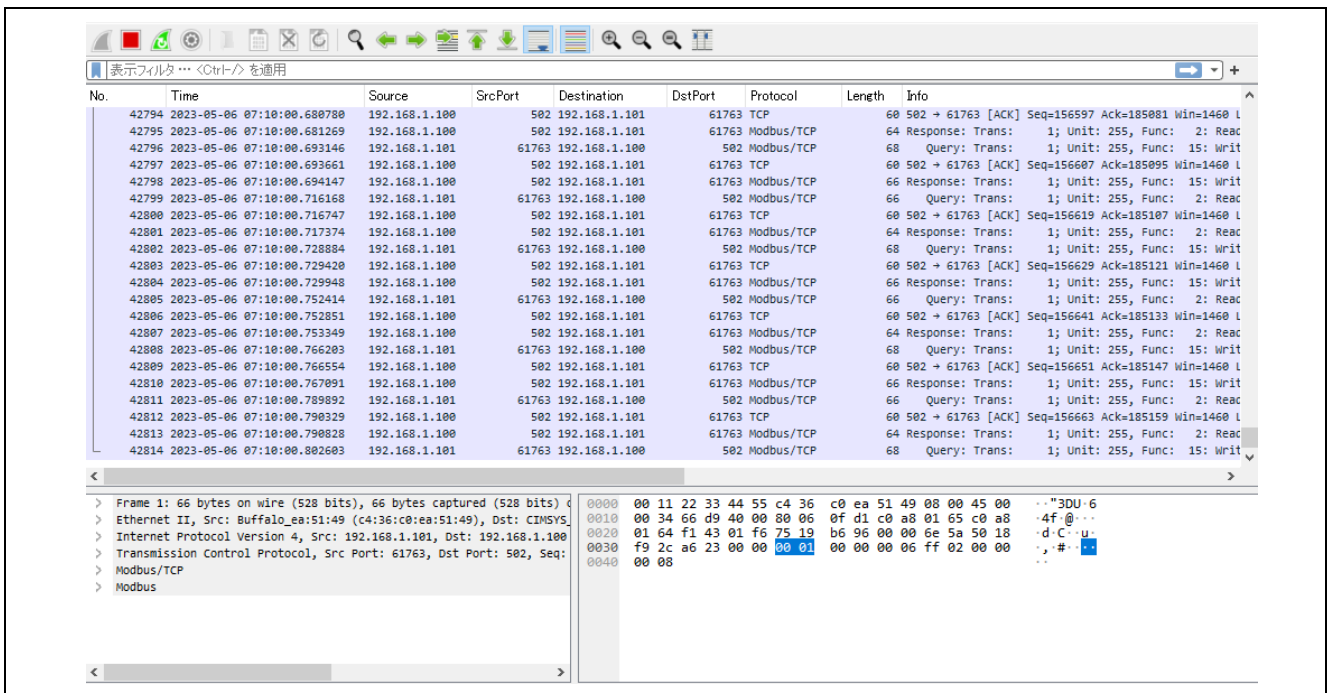
7.2 Modbus デモアプリケーションの設定

1. このサンプルプロジェクトパッケージに含まれている「ModbusDemoApplication.exe」を起動します。

「Connection」ではTCPサーバを選択し、サーバーのIPアドレス（例：「192.168.1.100」）とポート番号（例：「502」）を設定します。



2. 「Connect」ボタンを押すと Modbus 通信により LED の点滅が始まります。
3. Wireshark などのパケット解析ツールを使用すると、以下のように Modbus の通信状況を確認できます。



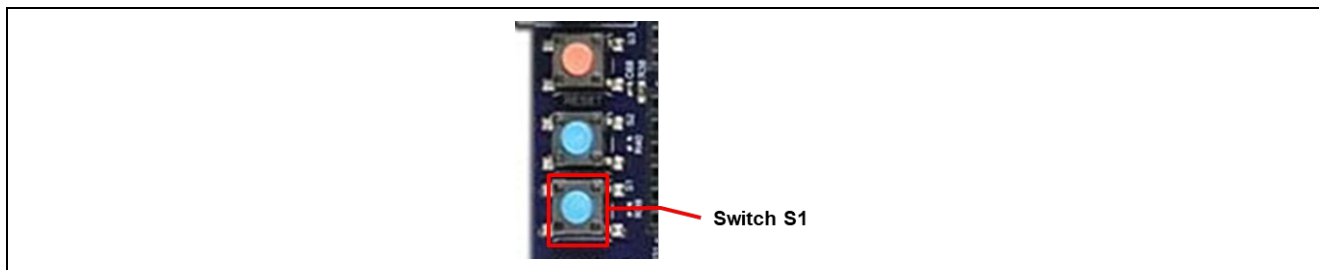
7.2.1 デモアプリケーションの仕様

Modbus TCPプロトコルを介してPCと通信することで、LEDの点滅を動的に制御します。

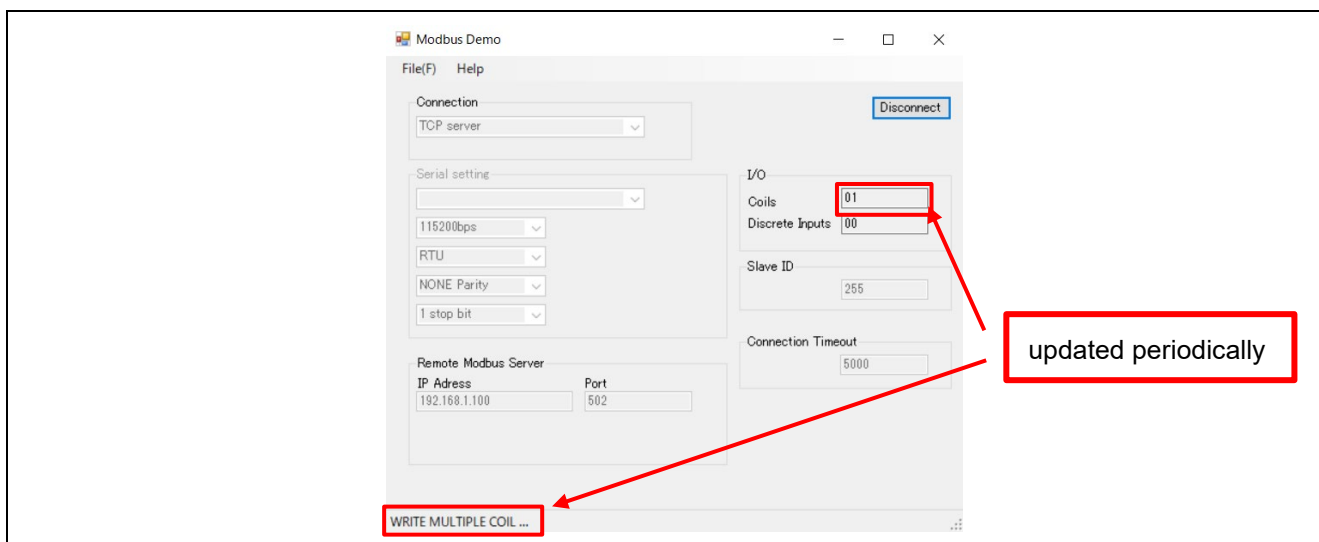
この制御には、ReadコイルコマンドとWrite Coilコマンドが使用されます。

PC アプリケーションは、Modbus の「Read_Discrete_Inputs」機能コードを使用して、スイッチ (SW1) の状態をチェックします。

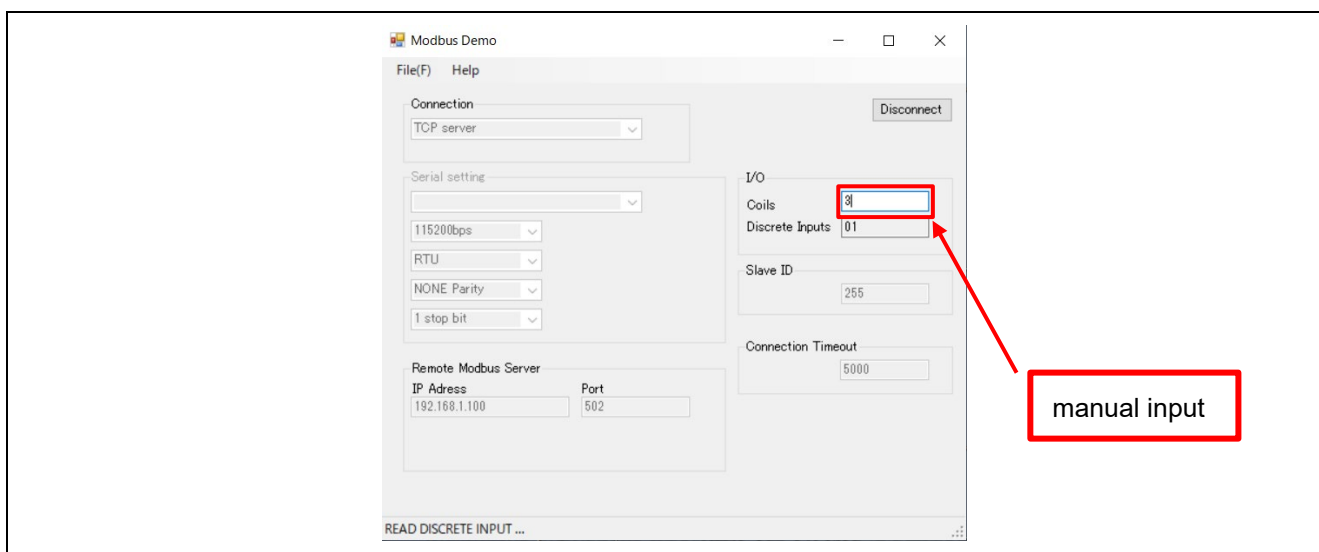
スイッチの状態に応じて、LED に接続されている出力ポートの状態が定期的に更新されます。



1. SW1 が OFF の場合、LED1～3 が周期的に点滅します。



2. SW1 が ON の場合デモアプリケーションの I/O のコイル値の入力値に応じて LED1～3 が点灯します。

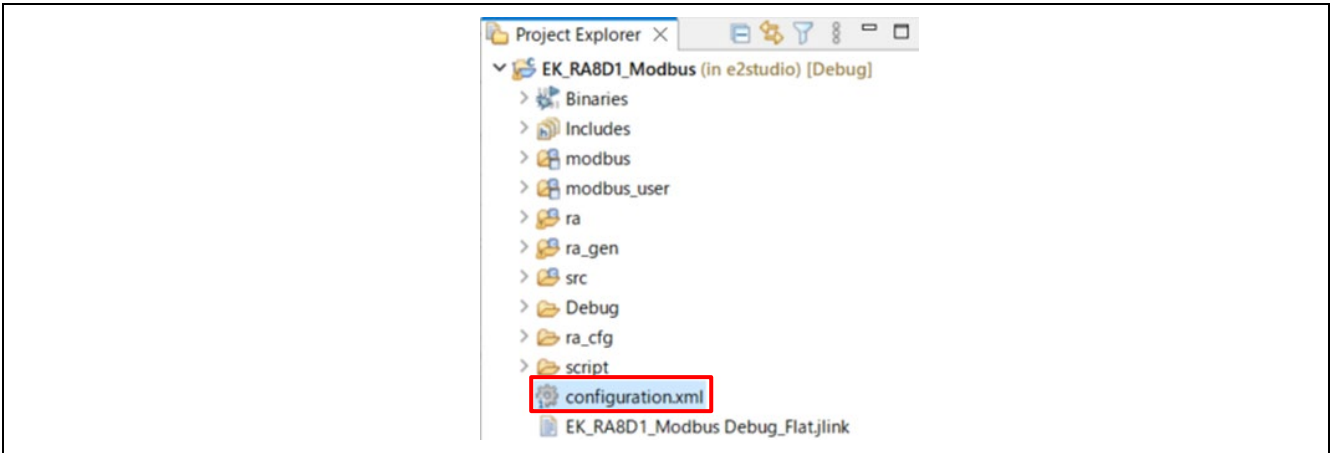


8. Appendix

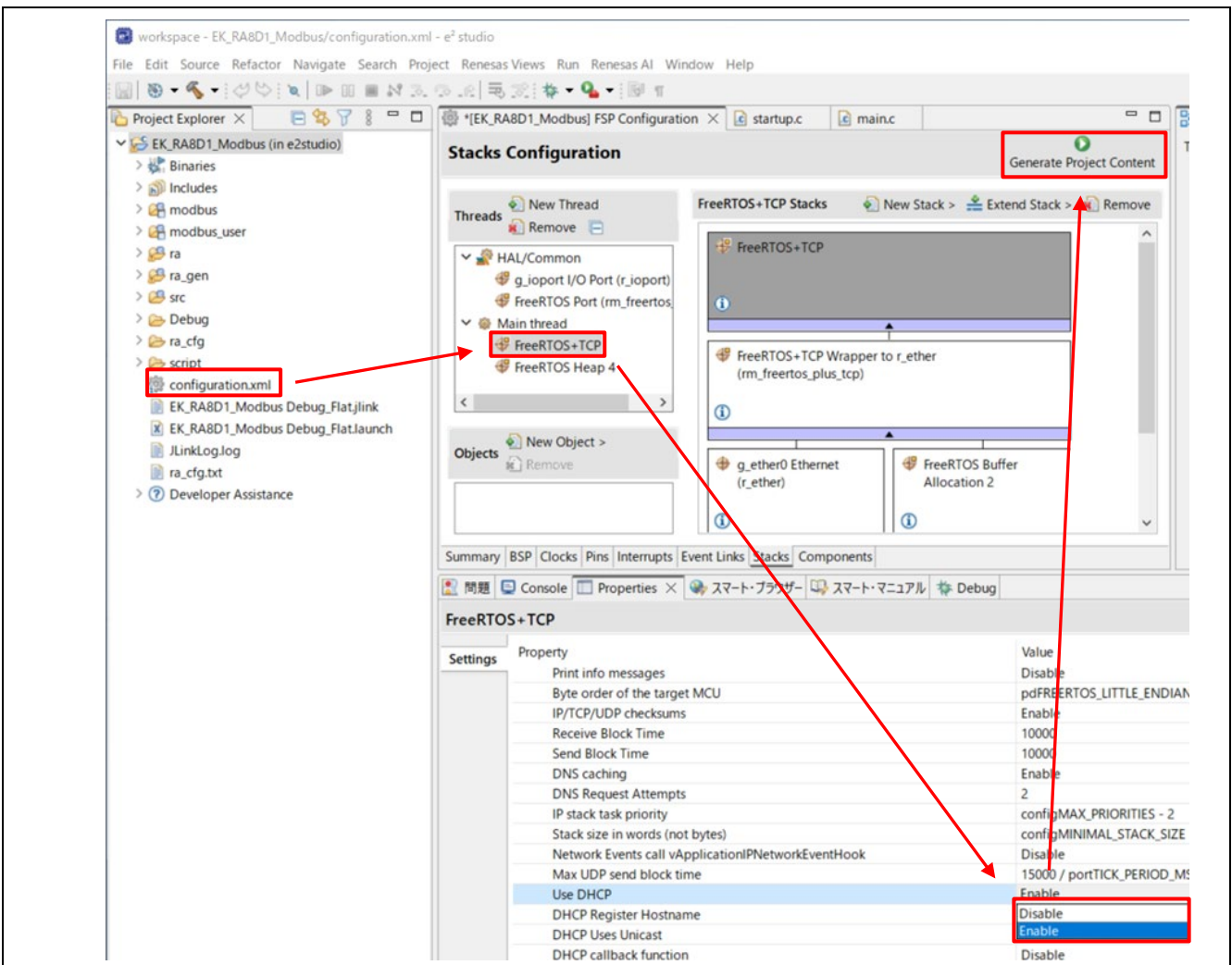
8.1 Appendix A: DHCP Mode

FSP を使用して DHCP mode をセットします。

1. Modbus プロジェクトから configuration.xml を開きます。



2. 「Stacks」タブをクリックしてスタック構成画面を開き、左側のスレッドウィンドウで「FreeRTOS + TCP」を選択します。プロパティを開き、「Use DHCP」を「Enable」に変更して「Generate Project Content」ボタンをクリックします。



8.2 Appendix B: User-defined function

このセクションでは、Modbus 機能コードについて説明します。ユーザーは、Modbus 関数コードの独自の実装を Modbus プロトコルスタックに登録できます。

8.2.1 Register callback function.

ユーザー作成の関数を定義します。このサンプルプログラムでは例として LED を使用しています。

定義ファイル: /Modbus_user/modbus_init.c

- ファンクションコード 1 に対応したコールバック関数(リードコイル)
 - スイッチに対応したものを読み出し。
- ファンクションコード 5 に対応したコールバック関数(シングルコイル書き込み)
 - LED に対応したコイルへの書き込み。

8.2.2 Register function code.

定義ファイル: /Modbus_user/modbus_func.c

コールバック関数に登録する関数を定義します。

8.2.3 User-Defined Functions

ユーザー定義関数は Modbus_user/modbus_user.c で定義されます。

各関数の処理にはユーザー定義の Read / Write 関数を使用します。

コイル/ディスクリット入力/ホールディングレジスタ/入力レジスタの各アドレスの対応する Read/Write 関数とそのテーブルを用意しています。

【Read Coils】

address	access
0001	LED1, g_coils_area
0002	LED2, g_coils_area
0003	LED3, g_coils_area
0004	g_coils_area
0005	g_coils_area
0006	g_coils_area
0007	g_coils_area
0008	g_coils_area

【Write_Single_Coils】

address	access
0001	LED1, g_coils_area
0002	LED2, g_coils_area
0003	LED3, g_coils_area
0004	g_coils_area
0005	g_coils_area
0006	g_coils_area
0007	g_coils_area
0008	g_coils_area

【Read_Discrete_Inputs】

address	access
1001	S1,g_discrete_input_area
1002	g_discrete_input_area
1003	g_discrete_input_area
1004	g_discrete_input_area
1005	g_discrete_input_area
1006	g_discrete_input_area
1007	g_discrete_input_area
1008	g_discrete_input_area
1009	g_discrete_input_area
10010	ILLEGAL DATA ADDRESS
10011	g_discrete_input_area
10012	g_discrete_input_area

【Read_Discrete_Inputs】

address	access
3001	g_input_reg_area
3002	g_input_reg_area
3003	g_input_reg_area
3004	ILLEGAL DATA ADDRESS
3005	ILLEGAL DATA ADDRESS
3006	ILLEGAL DATA ADDRESS
3007	ILLEGAL DATA ADDRESS
3008	g_input_reg_area

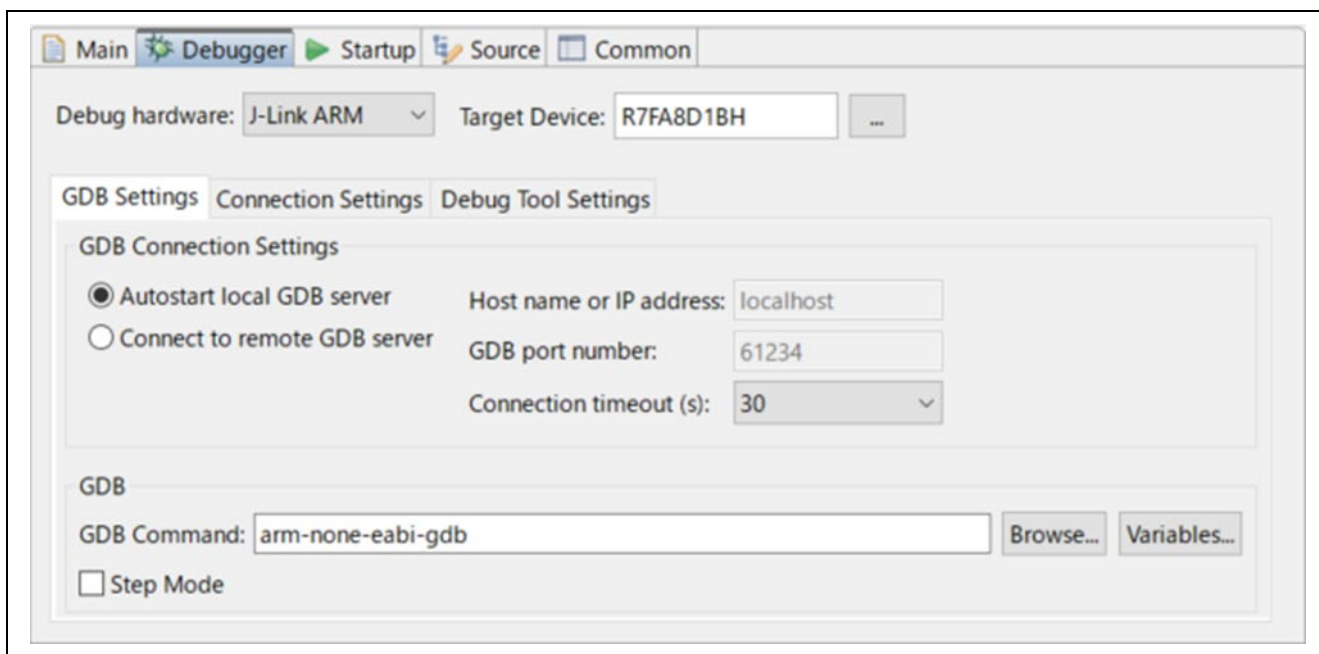
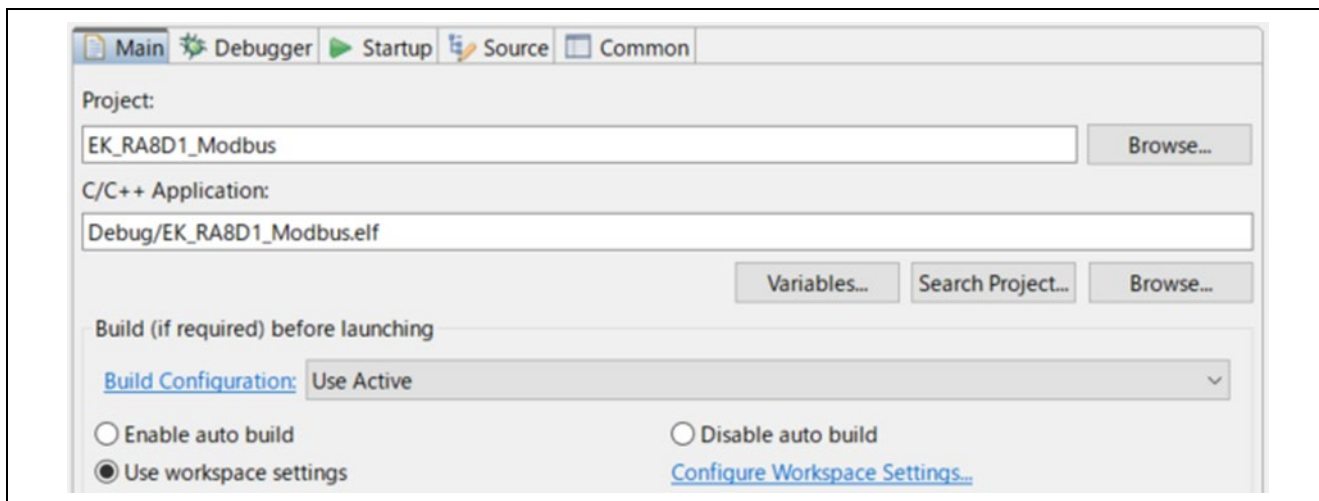
【READ_HOLDING_REGISTERS】

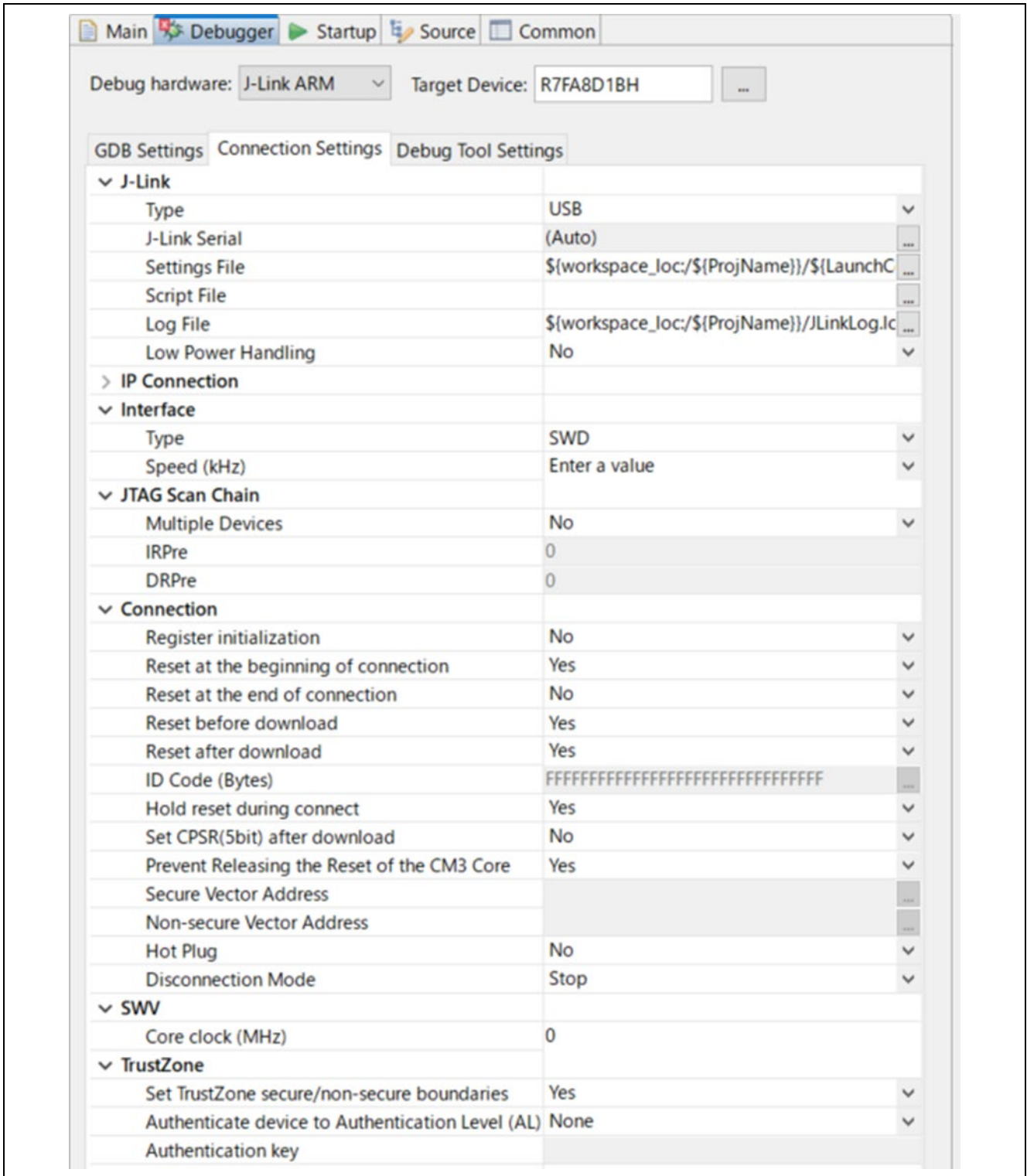
address	access
4001	g_holding_reg_area
4002	g_holding_reg_area
4003	g_holding_reg_area
4004	ILLEGAL DATA ADDRESS
4005	ILLEGAL DATA ADDRESS
4006	ILLEGAL DATA ADDRESS
4007	g_holding_reg_area

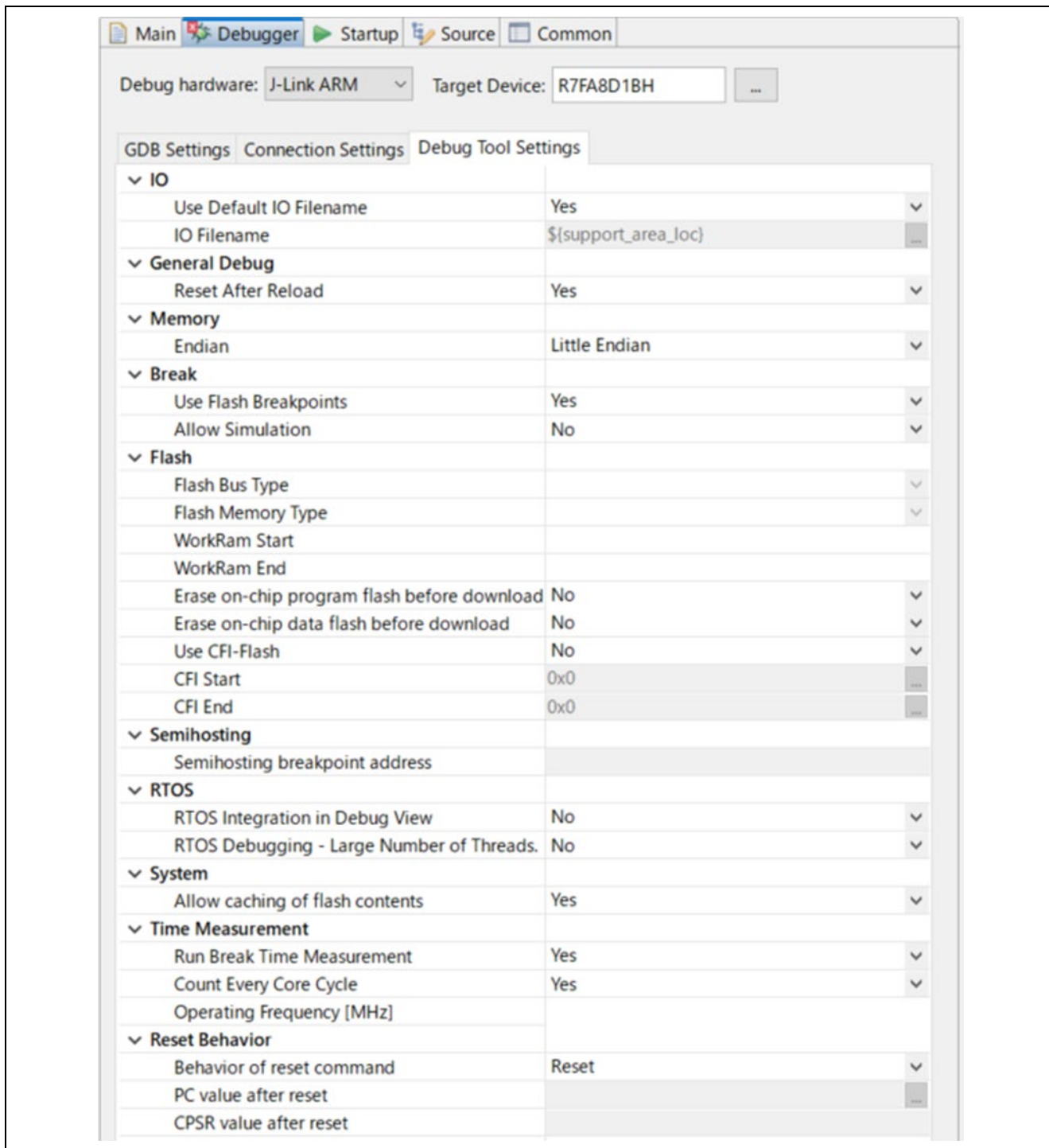
【WRITE_SINGLE_REGISTER】

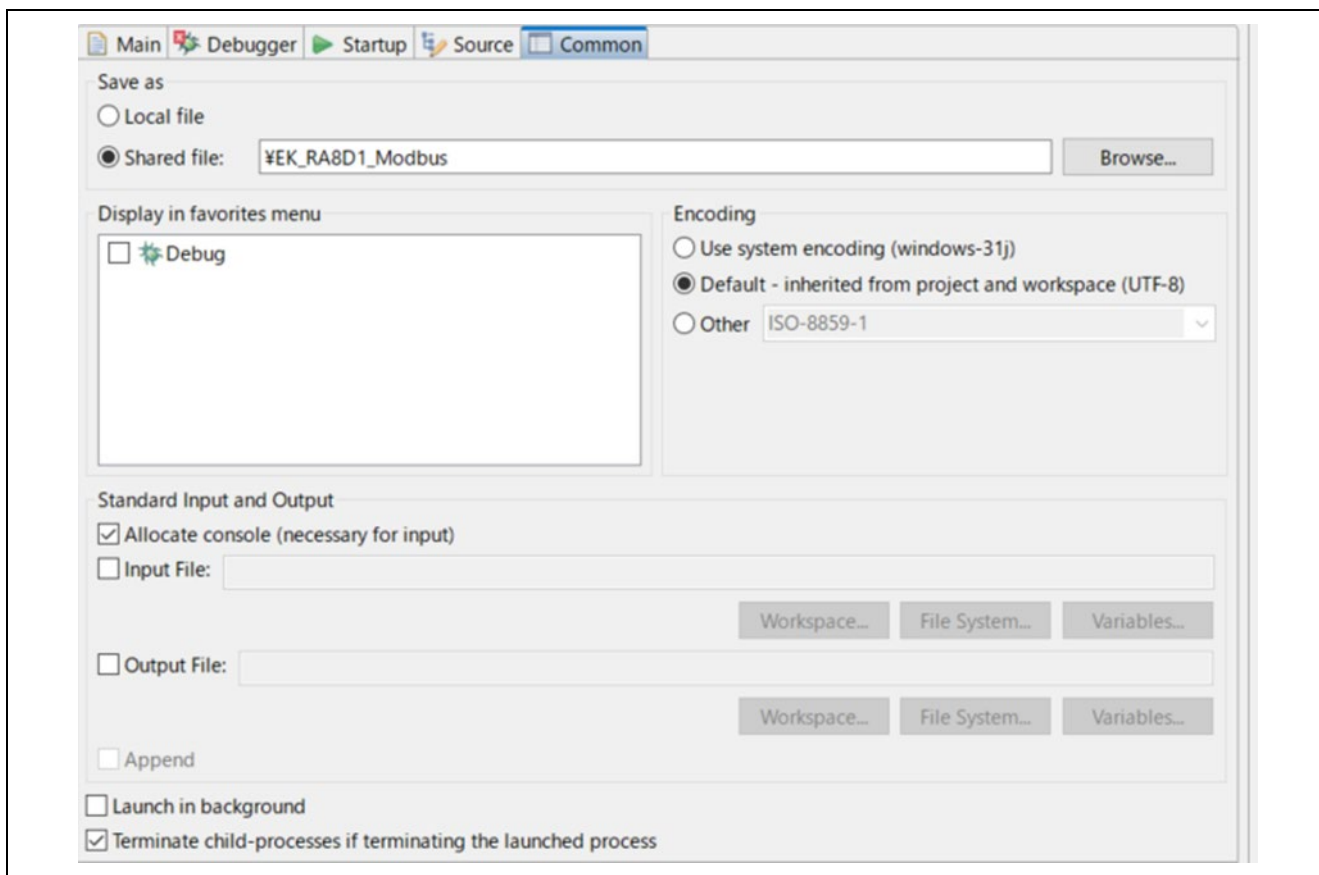
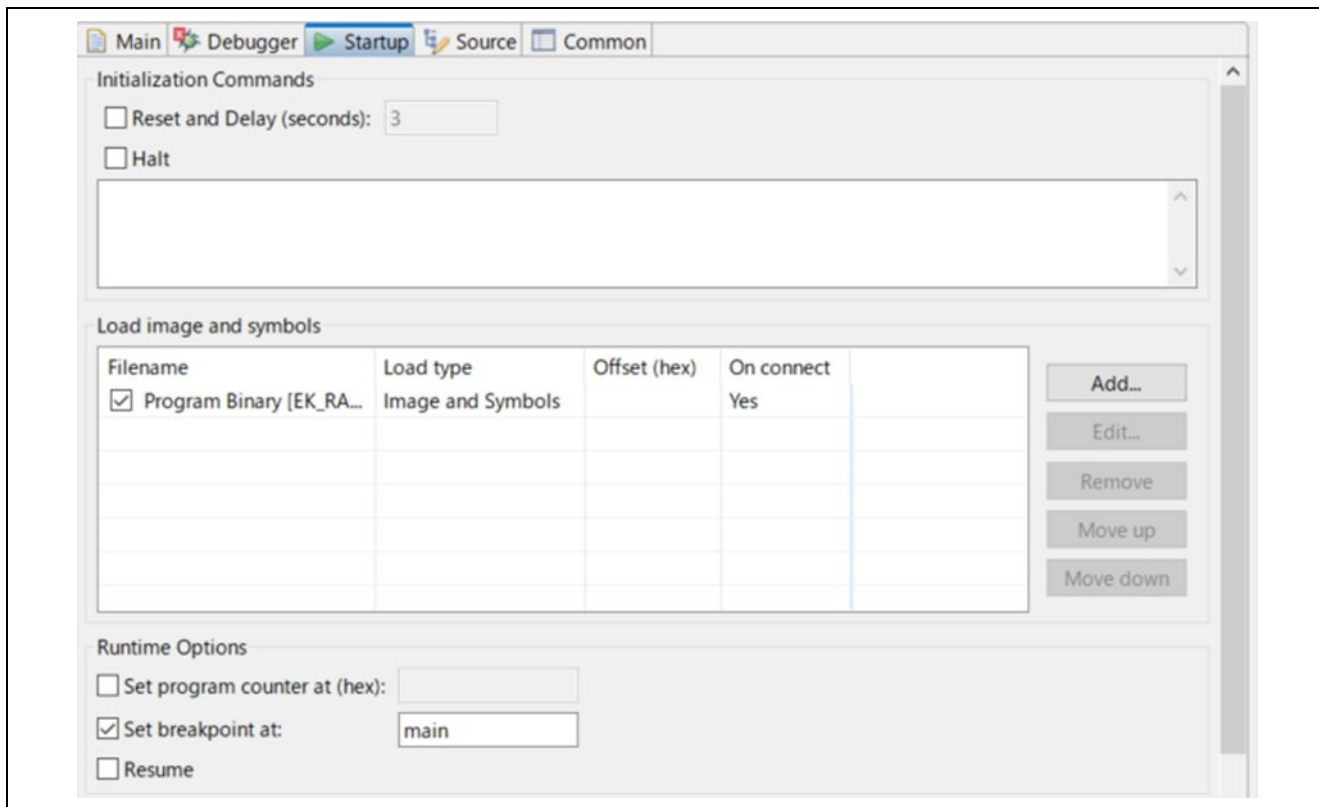
address	access
4001	g_holding_reg_area
4002	g_holding_reg_area
4003	g_holding_reg_area
4004	ILLEGAL DATA ADDRESS
4005	ILLEGAL DATA ADDRESS
4006	ILLEGAL DATA ADDRESS
4007	g_holding_reg_area

8.3 Appendix C: デバッカ構成









改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2023.12.11	-	初版発行

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しないでください。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えば、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、ノイズ輻射量などが異なる場合があります。型名が異なる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。

○Arm[®] およびCortex[®] は、Arm Limited（またはその子会社）のEUまたはその他の国における登録商標です。 All rights reserved.

○Ethernetおよびイーサネットは、富士ゼロックス株式会社の登録商標です。

○Modbus[®]は、Schneider Electric SAの登録商標です。

○その他、本資料中の製品名やサービス名は全てそれぞれの所有者に属する商標または登録商標です。

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。