

RA2E1 グループ

IO-Link ソリューション例

要旨

アプリケーションノートでは、Evaluation Kit for RA2E1 Microcontroller Group (EK-RA2E1) と IA Sensor Network Connector Board、ZSSC3240 Evaluation Board、EK-RA2E1 用変換ボードを使用し、RA2E1 で IO-Link 通信を実現するサンプルアプリケーションについて説明します。IO-Link は IEC61131-9 に準拠したセンサーおよびアクチュエータ用の通信技術です。IO-Link 通信には、TMG 社製の IO-Link スタックを使用します。

動作確認デバイス

RA2E1、ZIOL2401 (IO-Link PHY)、ZSSC3240

本アプリケーションノートを他のマイコンへ適用する場合、そのマイコンの仕様にあわせて変更し、十分に評価してください。

目次

| | |
|--|----|
| 1. 概要 | 4 |
| 2. 関連ドキュメント | 5 |
| 3. 動作確認環境 | 6 |
| 4. ハードウェア構成 | 7 |
| 4.1 IO-Link マスター | 8 |
| 4.2 IA Sensor Network Connector Board(IO-Link 通信ボード) | 8 |
| 4.3 EK-RA2E1 Board | 9 |
| 4.4 ZSSC3240 Evaluation Board | 10 |
| 4.5 EK-RA2E1 用変換ボード | 10 |
| 4.6 電源供給 | 11 |
| 4.6.1 5V サービス電源の接続 | 12 |
| 5. サンプルアプリケーション概要 | 13 |
| 5.1 全体処理フロー概要 | 14 |
| 5.2 App_SensorValue 関数フローチャート | 16 |
| 5.3 動作モードとスイッチング状態 | 21 |
| 5.4 IO-Link 通信仕様 | 22 |
| 5.4.1 ビットレート | 22 |
| 5.4.2 SIO モード | 22 |
| 5.4.3 プロセスデータ(PDIn) | 22 |
| 5.4.4 パラメータ | 23 |
| 5.5 使用する周辺機能と端子 | 24 |
| 5.6 プログラム構成 | 26 |
| 5.6.1 ファイル構成 | 26 |
| 5.6.2 関数一覧 | 28 |
| 6. サンプルプロジェクト実行方法 | 32 |
| 6.1 サンプルプロジェクトのインポート | 32 |
| 6.2 プロジェクトのビルド | 34 |
| 6.3 EK-RA2E1 ボードとホスト PC 間のデバッグ接続のセットアップ | 36 |
| 6.4 MCU へのサンプルプロジェクトの書き込み | 36 |
| 6.5 プログラムの開始 | 38 |
| 7. IO-Link Device Tool の使用準備及び機能解説 | 39 |
| 7.1 IO-Link Device Tool V5.1 – PE を起動 | 39 |
| 7.2 IO-Link Device Catalog の更新 | 40 |
| 7.3 IO-Link Master Catalog の更新 | 41 |
| 7.4 Catalog の更新の確認 | 43 |
| 7.5 IO-Link 通信のセットアップ | 44 |
| 7.6 IO-Link デバイスツールの EK-RA2E1 ボードセンサーデモ | 48 |
| 7.6.1 Common タブ | 49 |
| 7.6.2 Process Data タブ | 50 |

| | | |
|-------|------------------------------------|----|
| 7.6.3 | Identification タブ | 51 |
| 7.6.4 | Observation タブ | 52 |
| 7.6.5 | Parameter タブ | 53 |
| 7.6.6 | Scope タブ | 54 |
| 7.6.7 | デバイスのパラメータ変更 (Teach-In/Read) | 55 |
| 7.6.8 | IODD タブ | 60 |
| 8. | 使用例 | 61 |
| 8.1 | IO-Link パラメータをデフォルト値にリセット | 61 |
| 8.2 | 操作方法 | 62 |
| 9. | ZSSC の設定変更方法 | 65 |
| 9.1 | USB-SERIAL 変換ケーブルの接続 | 66 |
| 9.2 | 仮想 COM ポート番号確認 | 67 |
| 9.3 | ZSSC GUI の起動 | 68 |
| 9.4 | 通信経路確立 | 69 |
| 9.4.1 | Open Port | 69 |
| 9.4.2 | Interface | 70 |
| 9.4.3 | Scan I2C-Bus | 71 |
| 9.5 | 測定(Corr.Measurement) | 72 |
| 9.6 | NVM 操作 | 77 |
| 9.6.1 | Read NVM | 77 |
| 9.6.2 | Write NVM | 78 |
| 9.6.3 | Write CRC | 81 |
| 9.6.4 | ADC 分解能設定値の戻し | 82 |
| | 改訂記録 | 83 |

1. 概要

本アプリケーションノートは、TMG 社製の IO-Link スタックを使用して IO-Link 通信を実現する方法を説明します。

本例では EK-RA2E1 ボードと IA Sensor Network Connector Board を IO-Link デバイスとし、Pepperl + Fuchs 製 IO-Link-Master02-USB を IO-Link マスターとします。IO-Link マスターとの通信は、TMG 社が提供する「IO-Link Device Tool V5.1 - PE」を使用します。「IO-Link Device Tool V5.1 - PE」は、Windows PC 上で動作するアプリケーションソフトです。

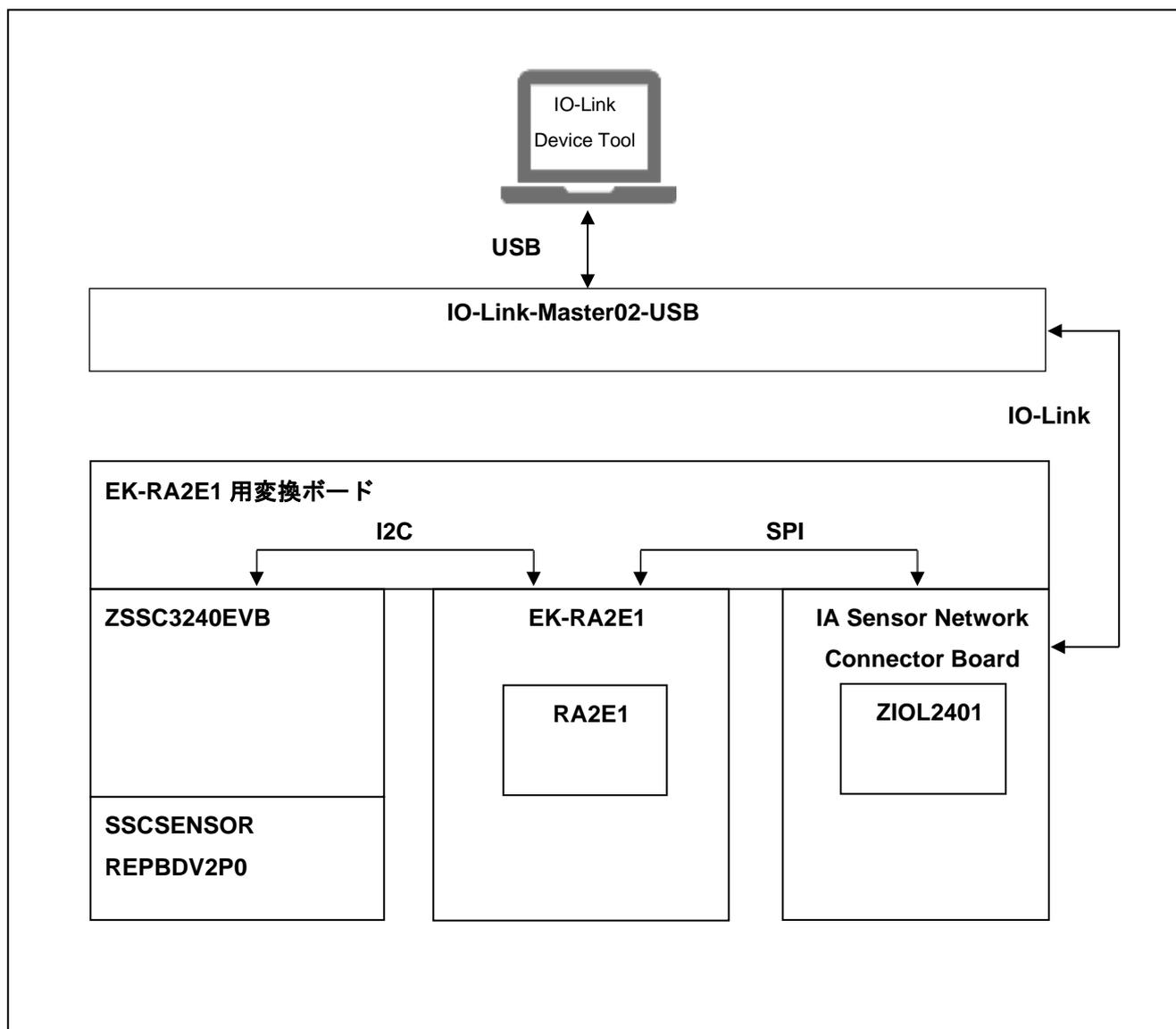


図 1-1 本アプリケーションノートにおけるシステム構成

2. 関連ドキュメント

- R11QS0035JGxxxx RA2E1 グループ RA2E1 MCU グループ用評価キット EK-RA2E1 クイックスタートガイド
- R20UT4825JGxxxx RA2E1 グループ RA2E1 MCU グループ用評価キット EK-RA2E1 v1 ユーザーズマニュアル
- R01UH0852JJxxxx Renesas RA2E1 グループ ユーザーズマニュアル ハードウェア編
- R11UM0155EU0270 Renesas Flexible Software Package (FSP) v3.7.0 User's Manual
- Renesas ZSSC3240 Evaluation Kit User Manual
- Renesas Datasheet ZSSC3240
- Renesas SSC Communication Board SSC-CB Datasheet

3. 動作確認環境

動作確認環境を表 3-1 に記載します。また、機器の構成を「4 ハードウェア構成」に記載します。

表 3-1 開発環境

| 項目 | 説明 |
|-------------------------|---|
| CPU ボード | EK-RA2E1 ボード |
| MCU | R7FA2E1A92DFM |
| IDE | Renesas e ² studio 2022-04 (22.4.0) |
| Tool Chain | GCC ARM Embedded 10.3.1.20210824 |
| FSP | 3.7.0 |
| ライブラリ | TMG 社製 IO-Link スタック |
| エミュレータ | SEGGER J-Link OB |
| IO-Link 通信ボード | IA Sensor Network Connector Board (RTK0EF0085B00001BJ) EK-RA2E1 ボードと接続し、IO-Link デバイスとして動作します。 |
| IO-Link Line Driver | ZIOL2401 |
| IO-Link マスター | Pepperl+Fuchs 製 IO-Link-Master02-USB デバイスタイプマネージャ ^注 |
| IO-Link ツール | IO-Link Device Tool V5.1 – PE |
| IO-Link ツール用ホスト PC | Windows10 Professional |
| ZSSC3240EVK | ZSSC3240 評価キット |
| ZSSC3240EVB | ZSSC3240 搭載 SSC 評価ボード |
| SSCCOMMBOARDV4P1C | CB(SSC 通信ボード) CB H/W Version(SSC CB V4.1) CB F/W Version(V4.19 © ZMD AG 2020 - CB SVN: 11903) ※ このボードは ZSSC3240EVK に含まれますが、本アプリケーション では使用しません。EK-RA2E1 ボードが PC と ZSSC3240EVB との通 信する為のパイプになります。 |
| SSCSENSORREPBVDV2P0 | SRB(Sensor Replacement Board) |
| ZSSC GUI | ZSSC3240 Evaluation Software (manual COMport selection, recommended for older PC) v3.02 |
| EK 変換ボード EX-EK-RA2E1 | EK-RA2E1、ZSSC 3240EVB、IA Sensor Network Connector Board を 接続します。 |
| USB-SERIAL 変換ケーブル | FT232RL 内蔵 USB-SERIAL 変換ケーブル |

【注】 IO-Link Device Tool のマスター検索 (7.5 章) において、IO-Link-Master02-USB が検出されない場合、ソフトウェア「デバイスタイプマネージャ」を再インストールください。

4. ハードウェア構成

本アプリケーションノートは IO-Link マスターと IO-Link デバイスで構成されます。IO-Link マスターは Pepperl+Fuchs 製 IO-Link-Master02-USB、IO-Link デバイスは ZSSC3240 Evaluation Board、EK-RA2E1 Board、IA Sensor Network Connector Board とそれらを繋げる EK-RA2E1 用変換ボードの 4 つで構成されます。各ボードの接続について以下に示します。

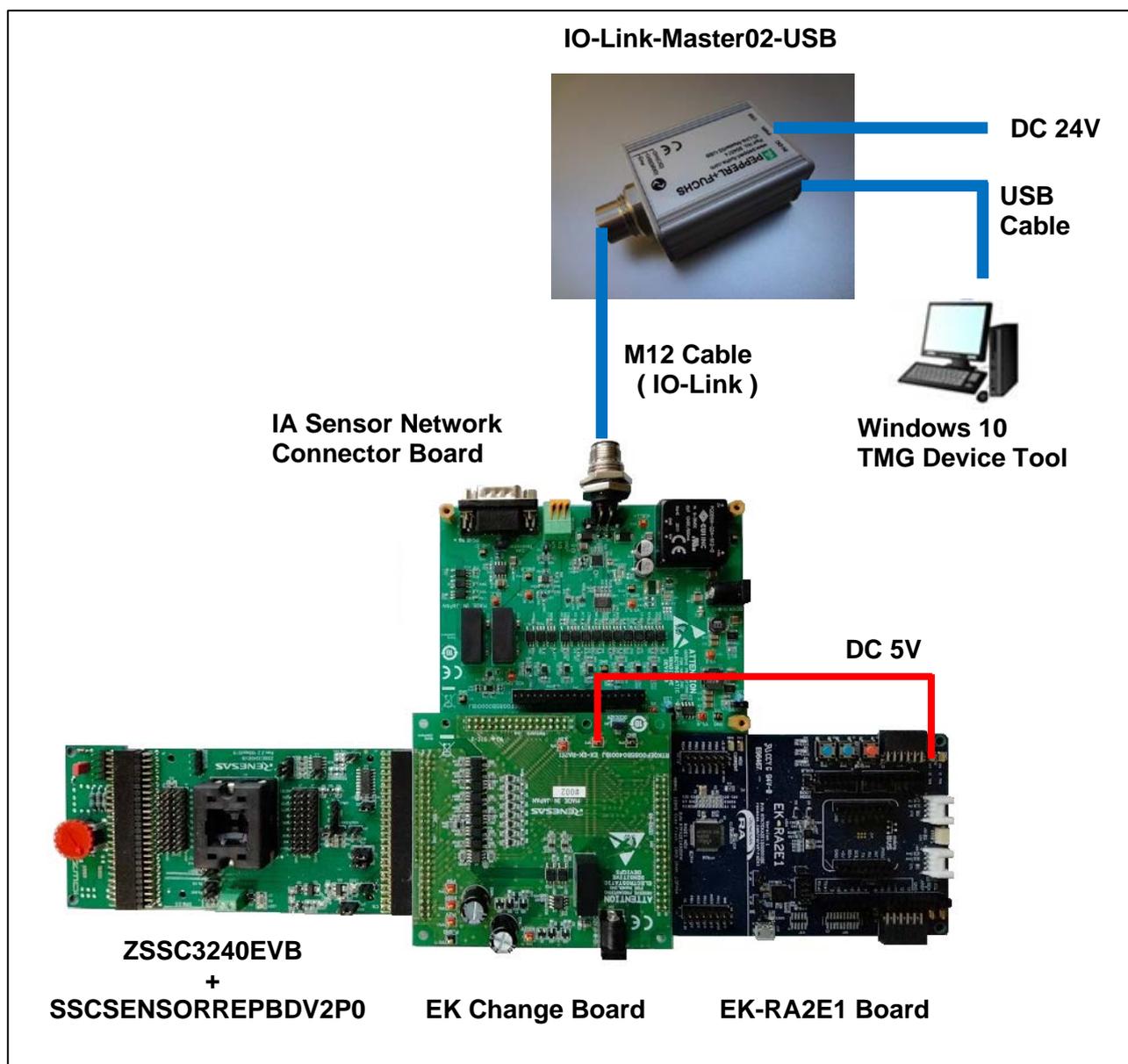


図 4-1 ハードウェア構成

4.1 IO-Link マスター

本アプリケーションノートでは Pepperl+Fuchs IO-Link-Master02-USB を使用しています。



図 4-2 IO-Link マスターコネクタ説明

電源は DC 24V を DC ジャックに、USB ケーブルは USB 端子に接続して下さい。

IO-Link ケーブル M12 をコネクタに接続して下さい。

4.2 IA Sensor Network Connector Board(IO-Link 通信ボード)

このボードは「IO-Link」「RS485」「CAN」に対応していますが、本アプリケーションノートでは「IO-Link」のみを使用します。IO-Link マスターボードから IO-Link ケーブルを介して電源が供給されます。

Jumper は、JP1:1-2、JP2:Open の設定とします。

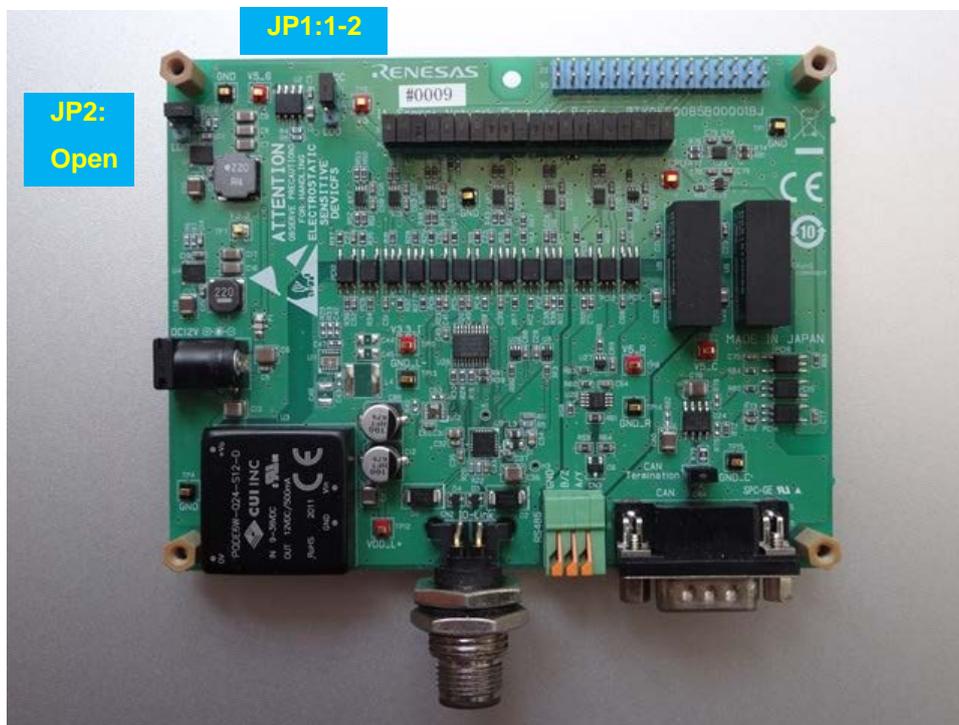


図 4-3 IO-Link 通信ボード

【注】 IO-Link マスターに電源が供給されている状態で IO-Link 通信ボードへの M12 ケーブル接続を行うと、IO-Link 通信ボードが故障する可能性があります。IO-Link 通信ボードへの M12 ケーブル接続は、IO-Link マスターに電源が供給されていない状態で実施してください。

4.3 EK-RA2E1 Board

このボードにサンプルプロジェクトのファームウェアを書き込み、ZSSC での計測と IO-Link マスターとの IO-Link 通信を制御します。

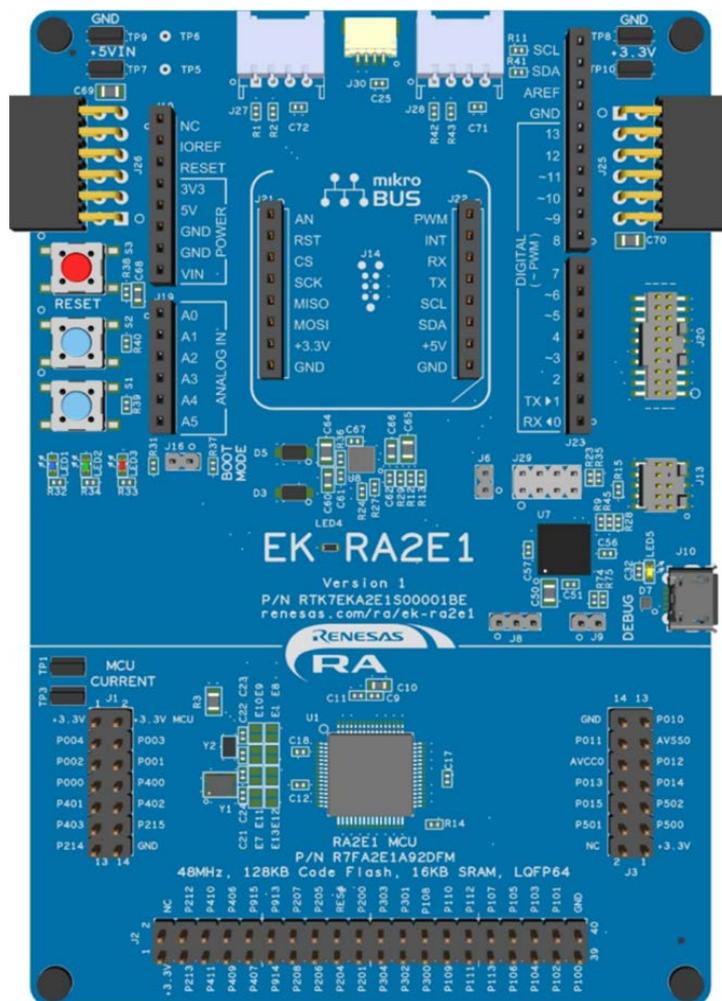


図 4-4 EK-RA2E1 ボード

4.4 ZSSC3240 Evaluation Board

ZSSC3240 Evaluation Board (以降 ZSSC EVB) と Sensor Replacement board (以降 SRB)を使用します。通信方式は「I2C」「SPI」「OWI」と3種類が用意されておりますが本アプリケーションノートでは「I2C」のみを対象とします。

ZSSC EVB を「I2C」で使用する為には、ボードの J21/J22 それぞれの 1-2 を Jumper 接続します。



図 4-5 ZSSC3240 Evaluation Board (I2C 設定)

4.5 EK-RA2E1 用変換ボード

この変換ボードを使用して、[ZSSC3240 Evaluation Board][EK-RA2E1 Board][IA Sensor Network Connector Board]を接続します。

電源は I/F コネクタを介して IO-Link 通信ボードから供給されます。ZSSC EVB にもこのボードから電源供給されます。また、このボードの 5V サービス出力と EK-RA2E1 ボードの+5VIN 端子をケーブル接続して EK-RA2E1 ボードに電源供給することができます。

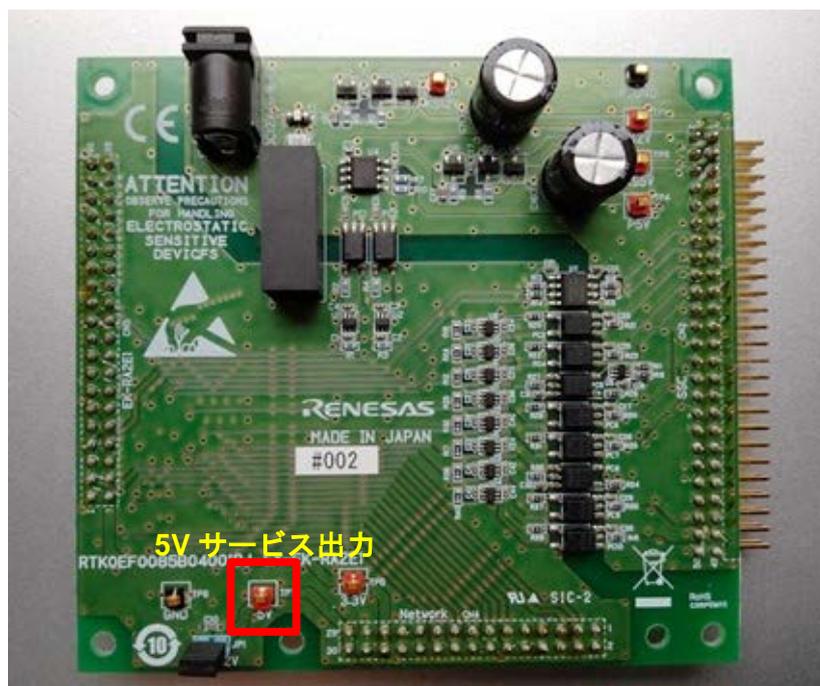


図 4-6 EK-RA2E1 用変換ボード

4.6 電源供給

各ボードへの電源は IO-Link マスターの IO-Link-Master02-USB から、IO-Link の M12 Cable を介して供給されます。

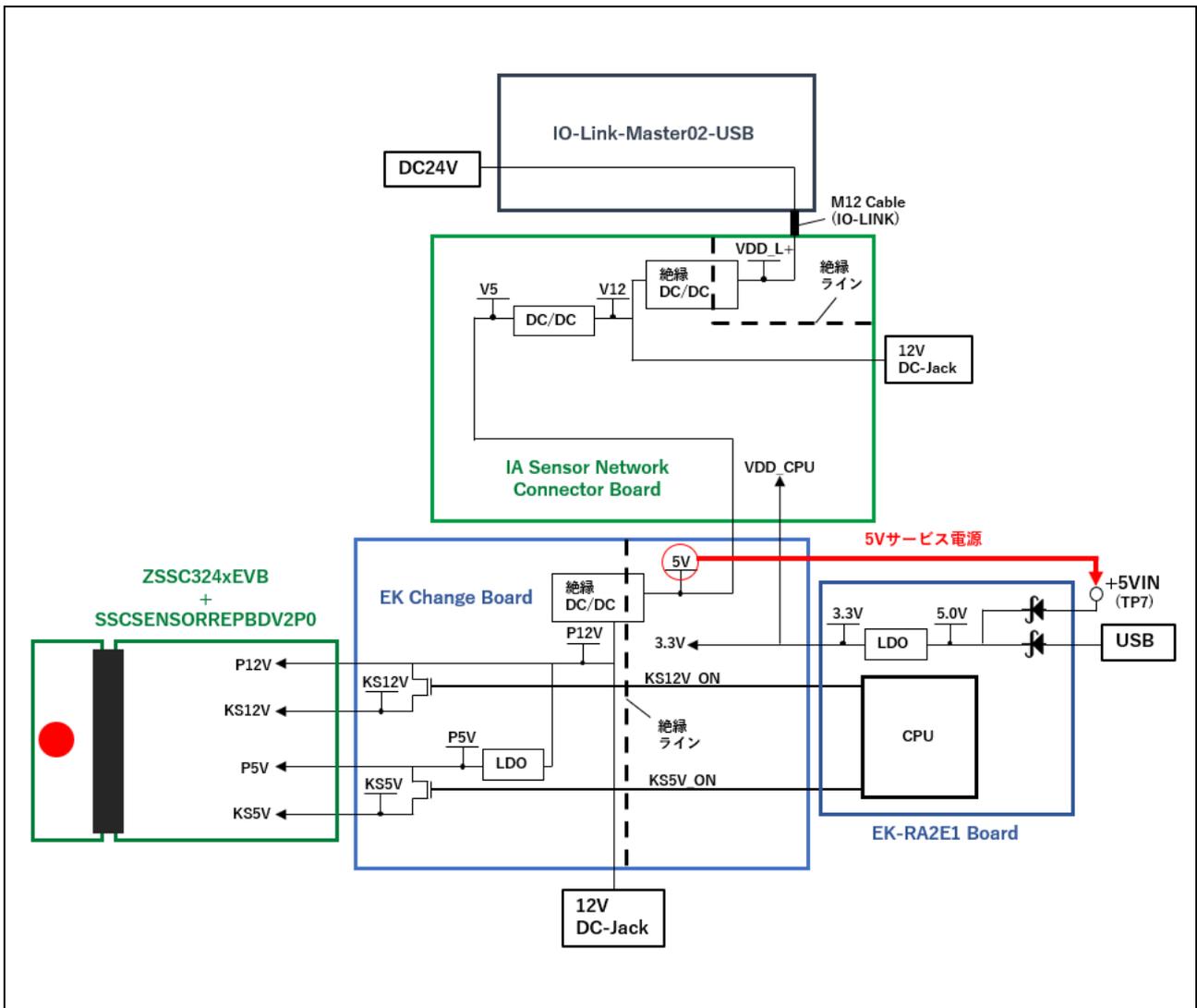


図 4-7 電源供給

4.6.1 5V サービス電源の接続

変換ボードの 5V(TP7) を EK-RA2E1 Board の +5VIN(TP7) に接続して、EK-RA2E1 Board に電源を供給します。

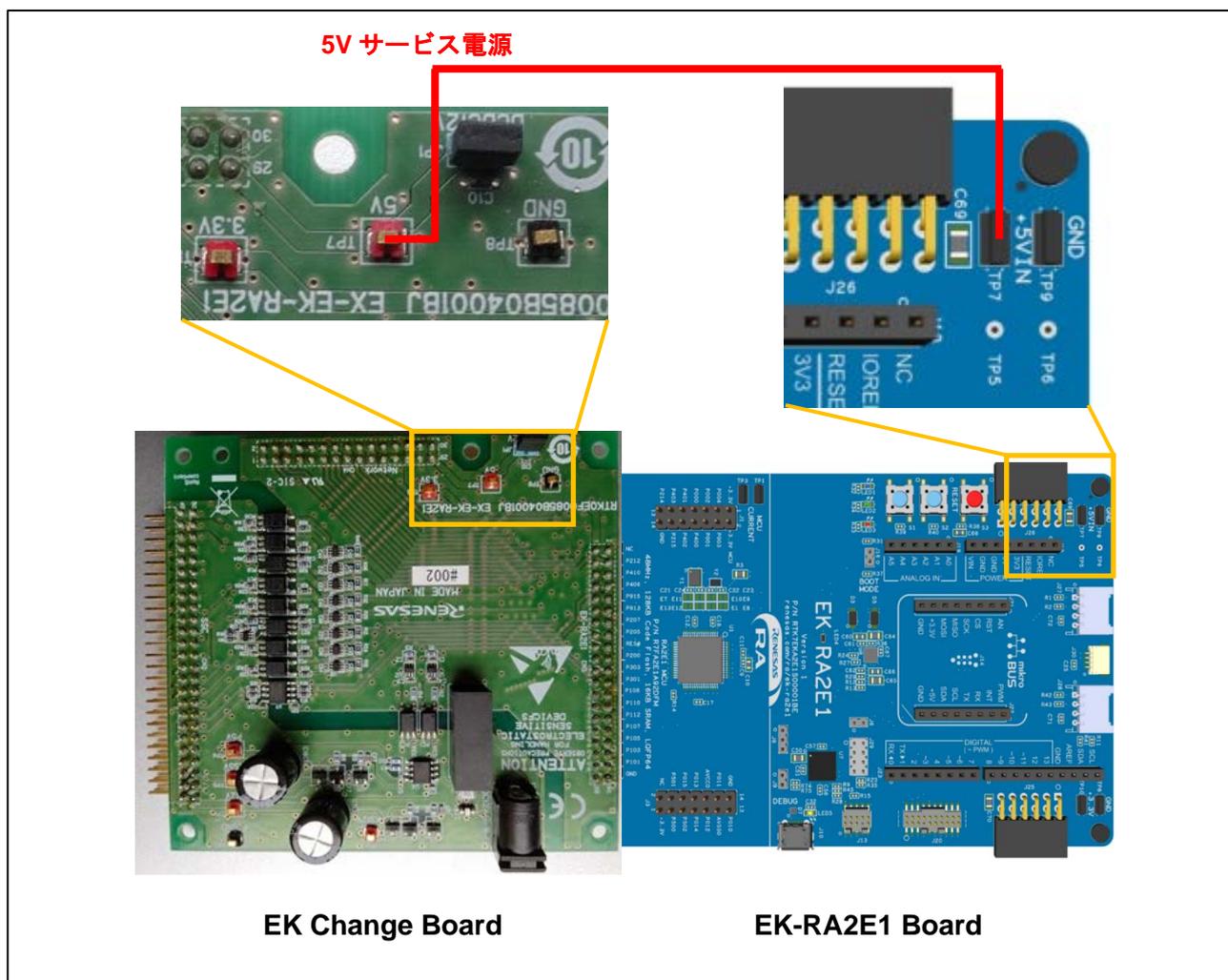


図 4-8 5V サービス電源の接続

5. サンプルアプリケーション概要

本サンプルアプリケーションで使用する IO-Link デバイスはスマートセンサープロファイルを搭載しており、ティーチングプロセスによってスイッチングモードと閾値に関する情報を受け取ります。IO-Link デバイスは測定値と閾値判定を定期的に（100[ms]に 1 回）実行し、IO-Link 通信で IO-Link マスターに情報を送信します。送信する情報（プロセスデータ）は、測定値とスイッチング状態（閾値判定結果）で構成されています。

スマートセンサープロファイルの詳細については、<https://io-link.com/en/> からダウンロードできる IO-Link スマートセンサープロファイルに関連するドキュメントを参照してください。

サンプルアプリケーションを構成するソフトウェアコンポーネントの概要を記載します。

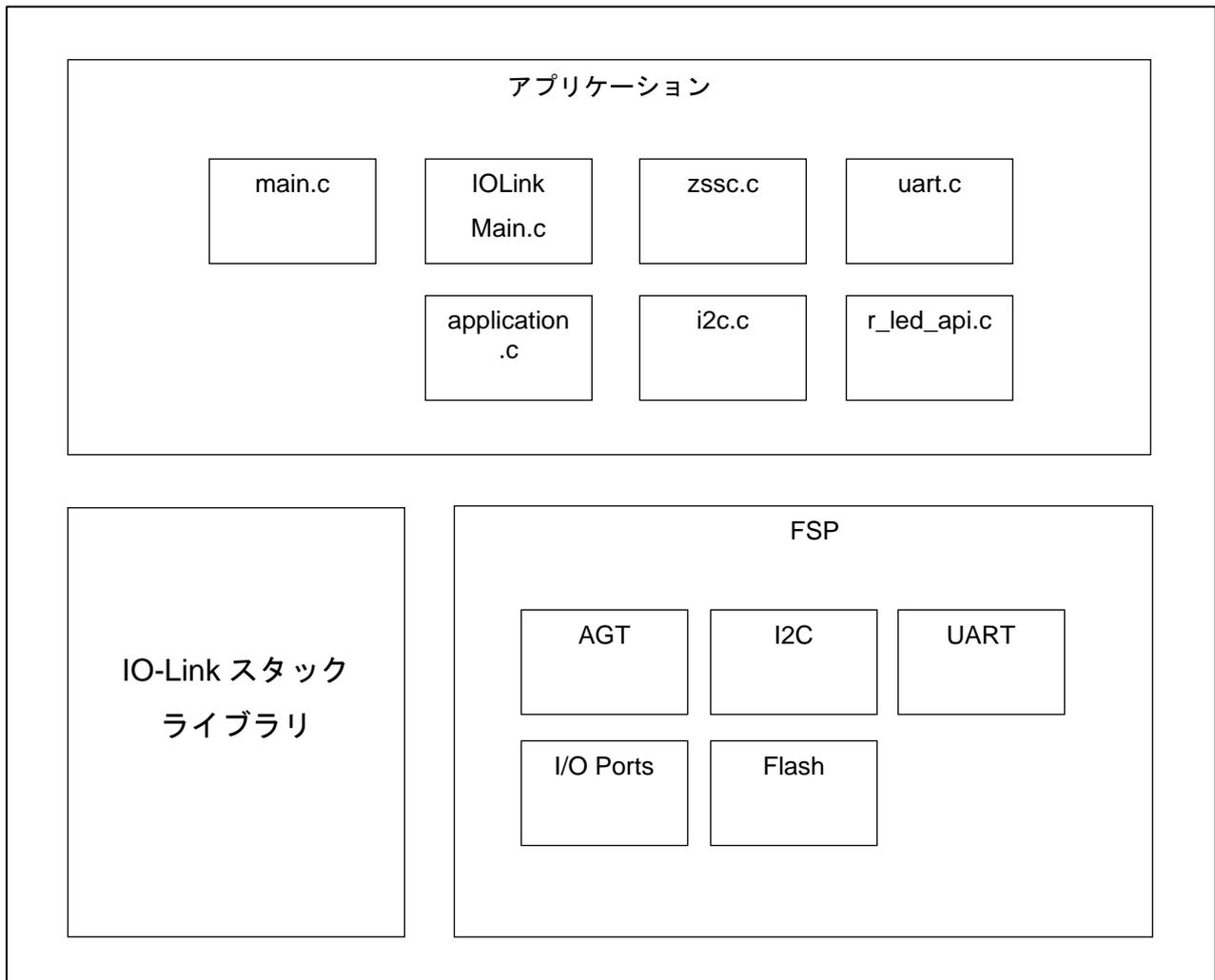


図 5-1 ソフトウェアコンポーネント

5.1 全体処理フロー概要

サンプルアプリケーションの処理フローを記載します。

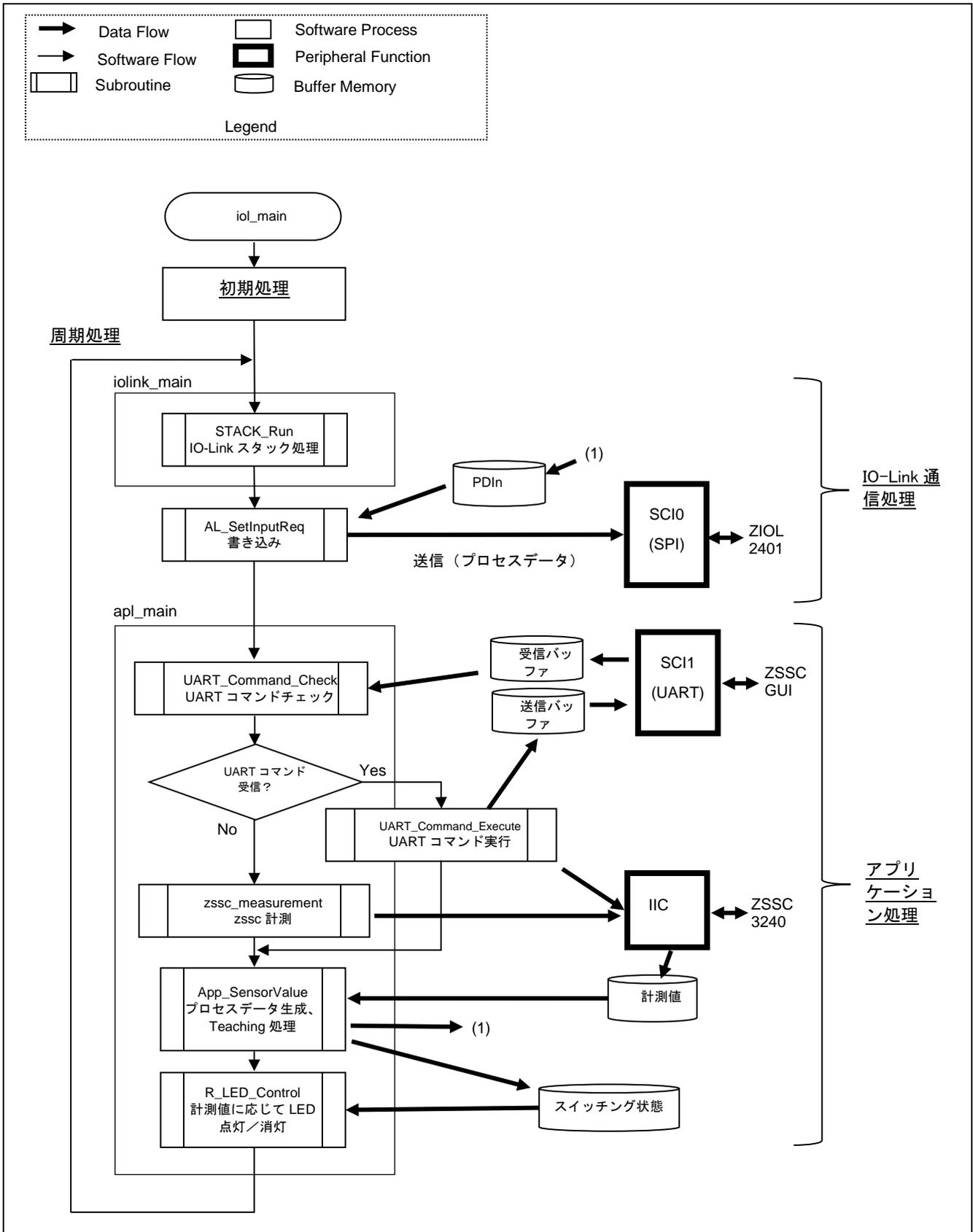


図 5-2 サンプルアプリケーション処理フロー

各処理の概略を記載します。

1. 初期処理

IO-Link、UART、I2C の初期化及び、ZSSC のリセットを行います。また ZSSC NVM データを読み出して変数に格納します。

2. IO-Link 通信処理

A) STACK_Run の実行(iolink_main)

IO-Link スタックが提供している API、STACK_Run を実行します。

B) プロセスデータ書き込み処理

IO-Link スタックが提供している API、AL_SetInputReq を実行します。引数としてプロセスデータのポインタを渡します。

3. アプリケーション処理

A) UART コマンド受信チェック関数の実行 (UART_Command_Check)

UART コマンド受信をチェックします。

B) UART コマンド受信有無による状態遷移

コマンド受信あれば UART_Command_Execute 関数を実行し UART コマンド処理を行います。UART コマンドが NVM 設定コマンドの場合は NVM データを読み出して変数に格納します。

C) 計測処理

UART コマンドが受信されていない場合、zssc_measurement をコールし ZSSC 計測を行います。ZSSC GUI から計測コマンドを受信していた場合は UART_Command_Execute 内で計測コマンドを発行して計測を行います。

どちらの場合でも計測結果をグローバル変数(計測値)に格納します。

D) プロセスデータ生成、Teaching 処理 (App_SensorValue)

計測値からプロセスデータを生成します。

Teaching コマンド実行中であれば、パラメータチェックを行い、有効範囲外であれば Teaching コマンド失敗とします。有効範囲内であれば Teaching コマンド成功として、コマンドに応じて SP1、または SP2 の設定値を更新します。

動作モードと計測値から、スイッチング状態の判定を行います。Teaching コマンド実行中

(Teach_Result が IDLE か SUCCESS 以外) の場合にはスイッチング状態の判定を行わず、スイッチング状態は OFF とします。

ConfigLogic (スイッチング状態の論理設定) が Inverted になっている場合には、スイッチング状態のビットを反転します。

この処理の詳細については「5.2 App_SensorValue 関数フローチャート」を参照してください。

E) LED 点灯/消灯 (R_LED_Control)

スイッチング状態を R_LED_Control の引数に渡します。スイッチング状態が ON であれば LED が点灯、OFF であれば消灯します。

4. 周期処理

2に戻り周期処理を繰り返します。

5.2 App_SensorValue 関数フローチャート

App_SensorValue 関数のフローチャートを記載します。

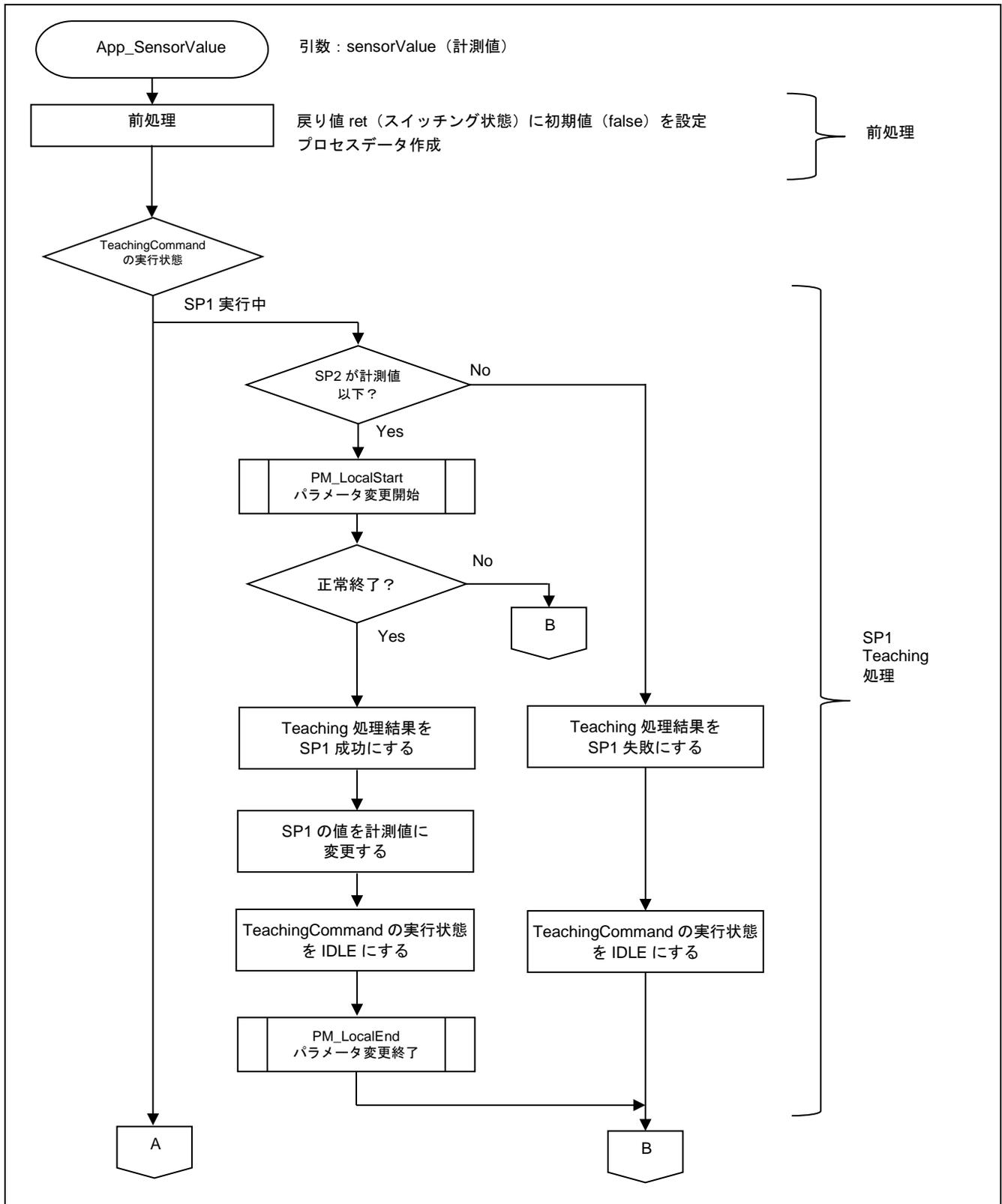


図 5-3 App_SensorValue フローチャート(1/4)

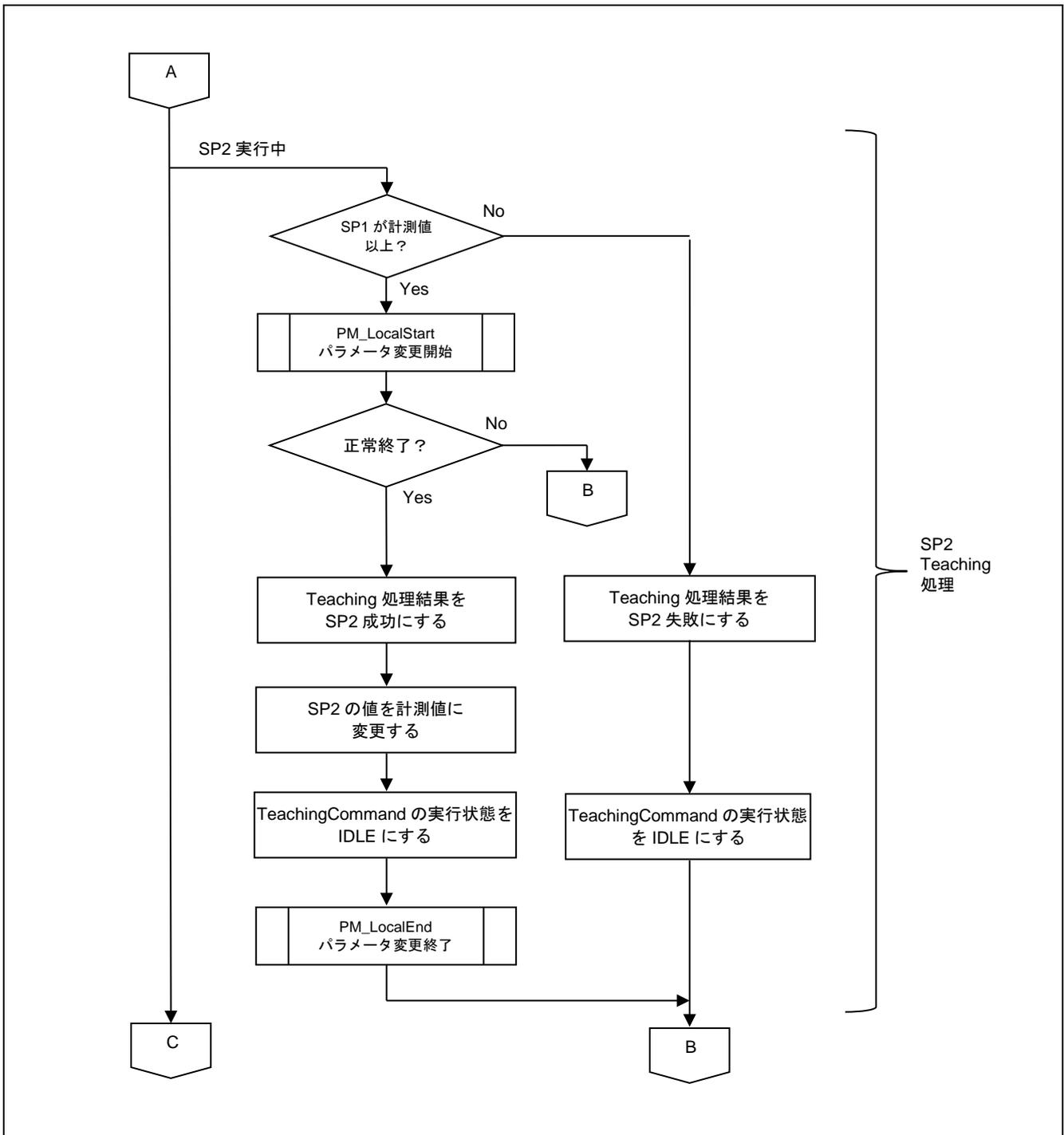


図 5-4 App_SensorValue フローチャート(2/4)

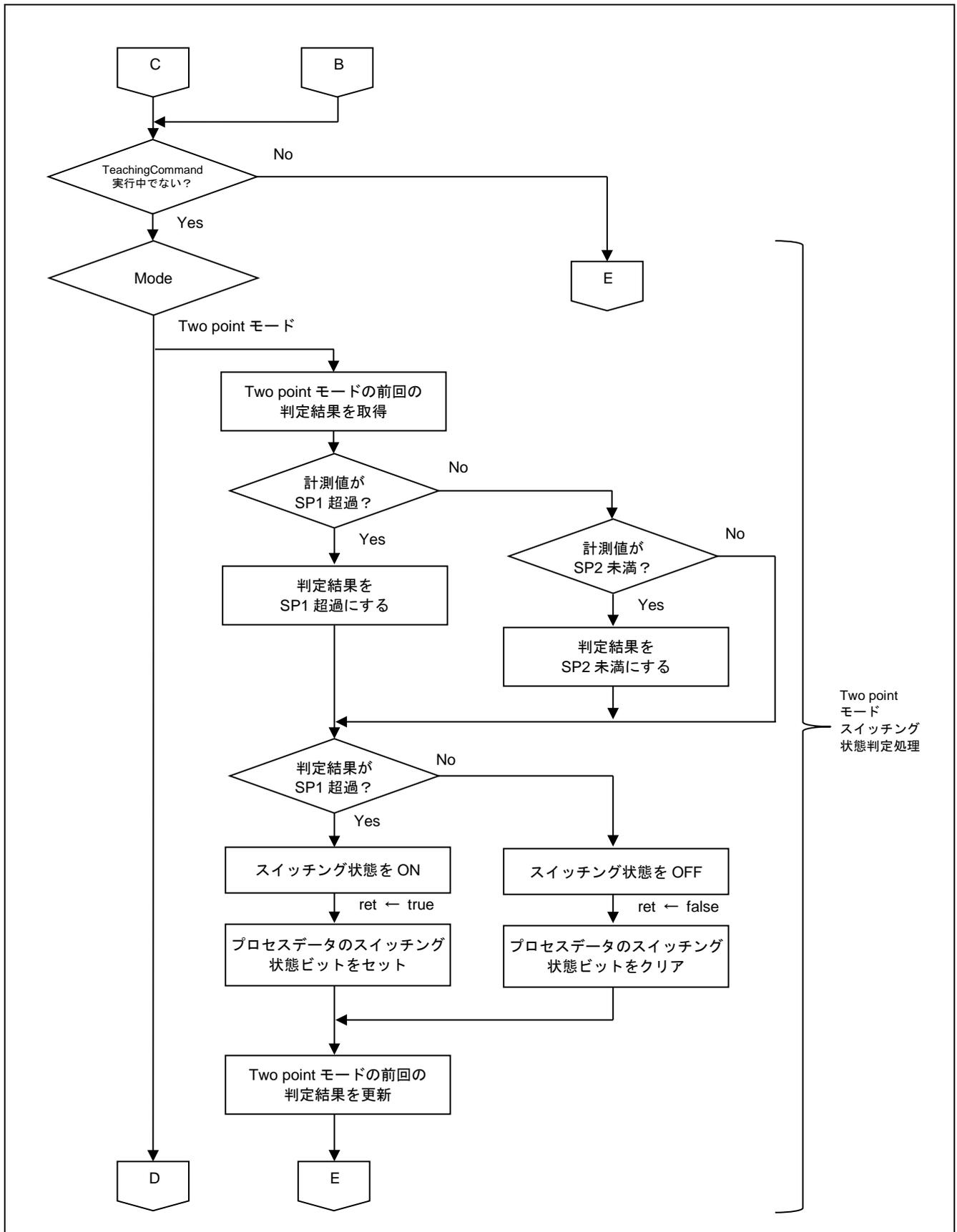


図 5-5 App_SensorValue フローチャート(3/4)

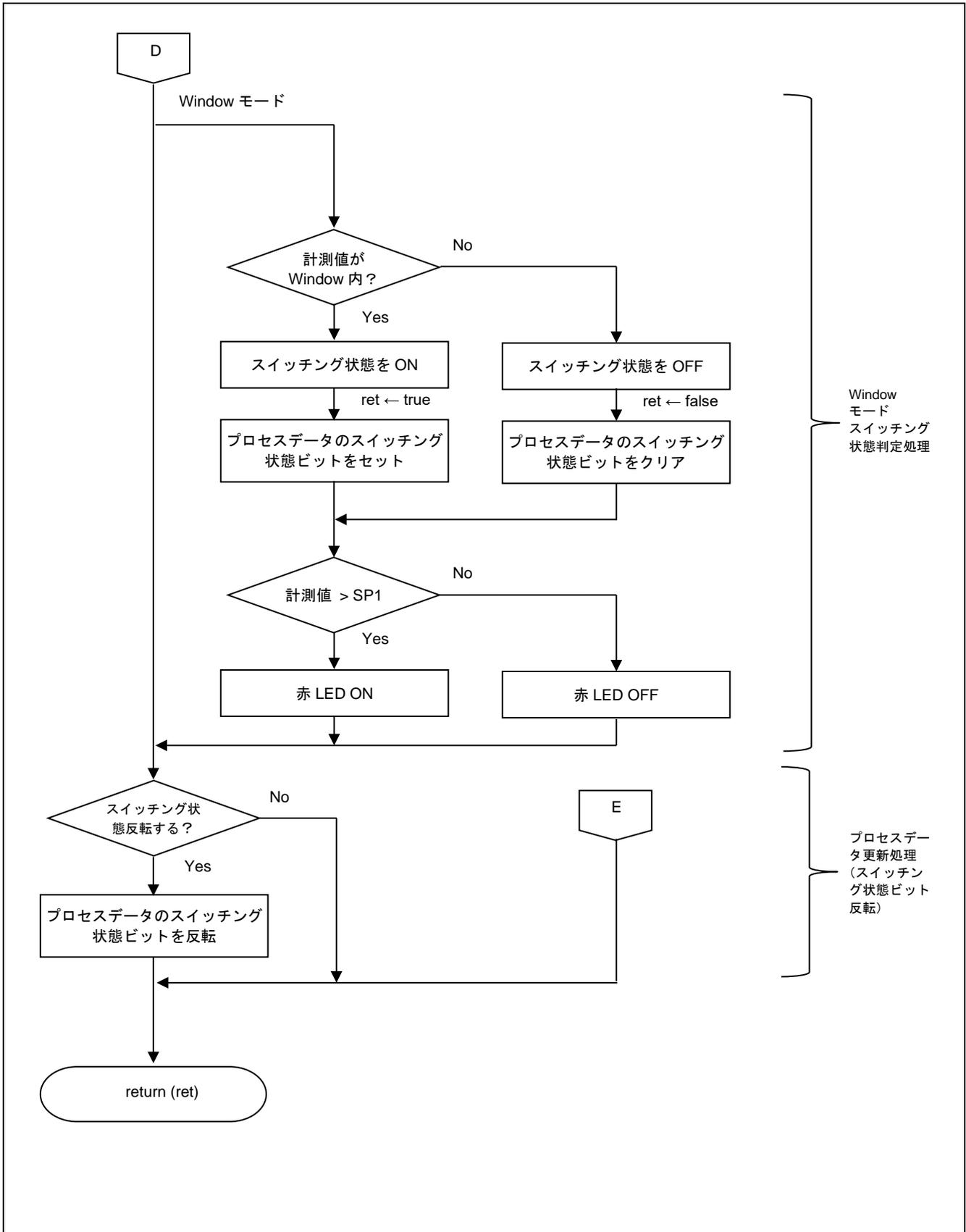


図 5-6 App_SensorValue フローチャート(4/4)

各処理の概略を記載します。

1. 前処理

戻り値 ret (スイッチング状態) に初期値 (false : OFF) を設定します。

プロセスデータを作成します。

引数で渡されたセンサー値 (32bit) の下位 24bit を ProcessData.PDIn[1] ~ [3],[5] に格納します。

ProcessData.PDIn[1] にセンサー値 [16 : 23] bit を格納します。

ProcessData.PDIn[2] にセンサー値 [8 : 15] bit を格納します。

ProcessData.PDIn[3] にセンサー値 [0 : 7] bit を格納します。

ProcessData.PDIn[5] の最下位 bit に Switching Point1 検出状態を格納します。

2. SP1 Teaching 処理^{※注意}

TeachingCommand の実行状態が「SP1 の TeachingCommand」の場合に、以下の処理を行います。

SP1 ≥ SP2 を満たす必要があるため、計測値 < SP2 の場合には、Teaching 処理の結果

(ParSetStatic.V_TeachResult) を SP1 失敗とします。

これ以外の場合、Teaching 処理の結果を SP1 成功として、SP1 (ParSet.V_SetPointValues.SP1) に計測値をコピーし、TeachingCommand の実行状態をアイドル状態にします。

3. SP2 Teaching 処理^{※注意}

TeachingCommand の実行状態が「SP2 の TeachingCommand」の場合に、以下の処理を行います。

SP1 ≥ SP2 を満たす必要があるため、SP1 < 計測値の場合には、Teaching 処理の結果

(ParSetStatic.V_TeachResult) を SP2 失敗とします。

これ以外の場合、Teaching 処理の結果を SP2 成功として、SP2 (ParSet.V_SetPointValues.SP2) に計測値をコピーし、TeachingCommand の実行状態をアイドル状態にします。

4. Two point モード スwitching状態判定処理

TeachingCommand が実行中でない場合(アイドル状態、または成功)で、且つ、動作モード (ParSet.V_SetPointConfig.Mode) が Two point モードの場合に、以下の処理を行います。

Two point モードの前の判定結果を取得します(初期値は SP2 未満とします)。

SP1 < 計測値の場合には、判定結果を SP1 超過とします。

SP2 > 計測値の場合には、判定結果を SP2 未満とします。

それ以外の場合には、判定結果は前の判定結果とします。

判定結果が SP1 超過の場合には、スイッチング状態を ON として、ProcessData.PDIn[5] を 1 にします。

判定結果が SP2 未満の場合には、スイッチング状態を OFF として、ProcessData.PDIn[5] を 0 にします。

Two point モードの前の判定結果を、今回の判定結果に更新します。

5. Window モード スwitching状態判定処理

TeachingCommand が実行中でない場合(アイドル状態、または成功)で、且つ、

動作モード (ParSet.V_SetPointConfig.Mode) が Window モードの場合に、以下の処理を行います。

SP1 ≥ 計測値 ≥ SP2 の場合には、スイッチング状態を ON として、ProcessData.PDIn[5] を 1 にします。

これ以外の場合には、スイッチング状態を OFF として、ProcessData.PDIn[5] を 0 にします。

赤色 LED 処理 :

SP1 < 現在のセンサー値の場合には赤色 LED を点灯します。

SP1 ≥ 現在のセンサー値の場合には赤色 LED を消灯します。

6. プロセスデータ更新処理(スイッチング状態ビット反転)

スイッチング状態論理設定 (ParSet.V_SetPointConfig.Logic) が 1 (Inverted) の場合に ProcessData.PDIn[5] の最下位ビットを反転します。

【注】 ParSet.V_SetPointValues を変更する際は、IO-Link スタックが提供する PM_LocalStart 関数を実行し、戻り値が True であることを確認してから変更を行います。変更後には PM_LocalEnd 関数を実行する必要があります。PM_LocalStart 関数の戻り値が false であった場合には、ParSetStatic.V_TeachResult、Parset.V_SetPointValues、TeachingCommand の実行状態を変更せずに App_SensorValue 関数を抜け、次回の計測後の処理で Teaching 処理をリトライします。

5.3 動作モードとスイッチング状態

本サンプルアプリケーションの動作モードとスイッチング状態について記載します。本サンプルアプリケーションでは、Window モードと Two point モードの2つのモードで動作します。スイッチング状態についてはプロセスデータの他に LED でも確認できるよう、スイッチング状態に応じて LED の点灯/消灯を行います。

- Window モード
Switching Point1(SP1) \geq 計測値 \geq Switching Point2 (SP2)を満たしたら LED(緑)が ON。
計測値 $>$ Switching Point1(SP1) を満たしたら LED(赤)が ON。
- Two point モード
このモードでは、ヒステリシスを持たせてスイッチング状態を判定します。
 - 測定値が上昇中の場合、閾値は SwitchingPoint1(SP1) にあります。測定値 $>$ SP1 を満たしたら LED(緑)が ON に変化します。その後、測定値が SP2 を下回るまでは状態を維持します。
 - 測定値が下降中の場合、閾値は Switching Point2(SP2)にあります。SP2 $>$ 測定値を満たしたら LED(緑)が OFF に変化します。その後、測定値が SP1 を上回るまでは状態を維持します。

動作イメージを記載します。

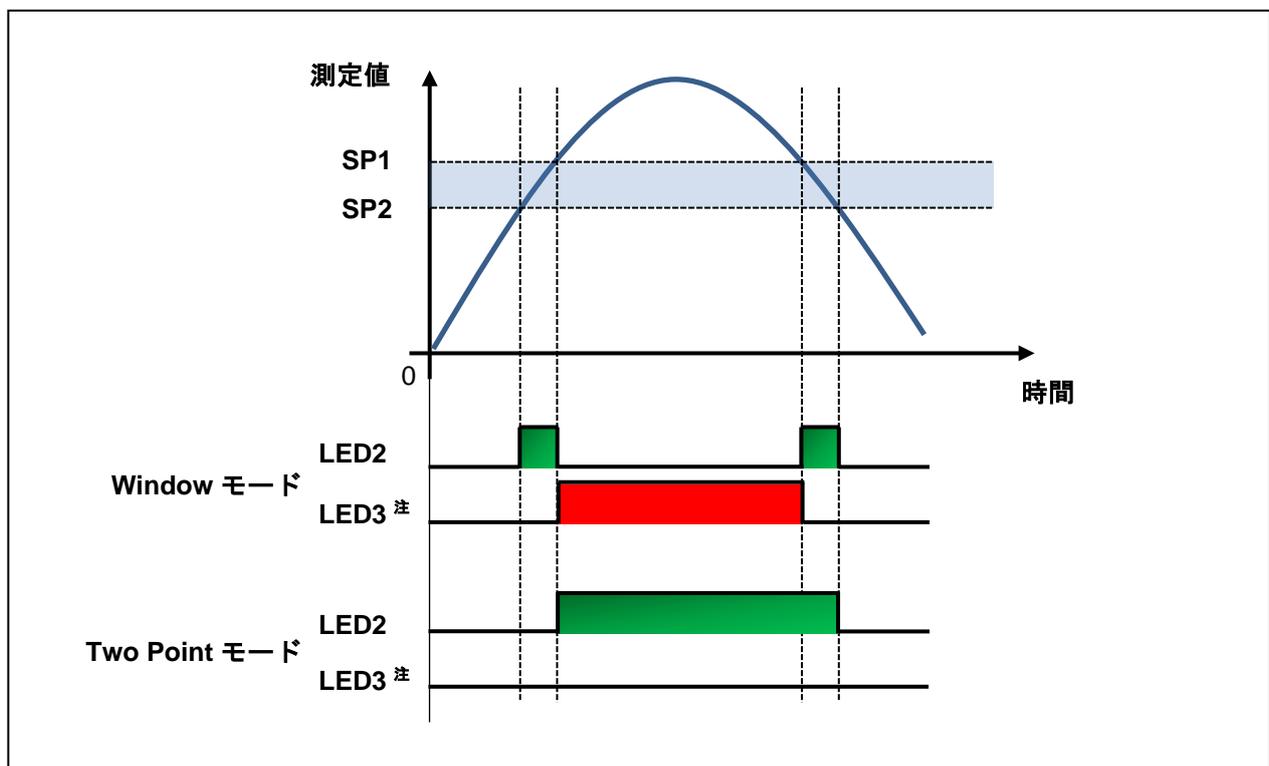


図 5-7 LED 動作イメージ

【注】 赤色 LED は Window モード時のみ動作します。また赤色 LED の状態は ProcessData には反映はされない為、TMG Device Tool での状態確認はできません。

5.4 IO-Link 通信仕様

IO-Link の通信仕様について記載します。

5.4.1 ビットレート

ビットレートは COM2(38.4[kbps])です。

5.4.2 SIO モード

SIO モードは本サンプルアプリケーションでは非対応です。

5.4.3 プロセスデータ (PDIn)

計測情報はプロセスデータとして IO-Link マスターに送信されます。プロセスデータの内容を表 5-1 に記載します。

表 5-1 プロセスデータ (PDIn)

| プロセスデータ (PDIn) データ長 : 6[bytes] | | | |
|--------------------------------|-----|------------------------|------------------------|
| PDIn[n] | Bit | 格納データ | Details |
| PDIn[0] | 0-7 | Sensor Value(bit24-32) | センサー値の Bit24-32 を格納 |
| PDIn[1] | 0-7 | Sensor Value(bit16-23) | センサー値の Bit16-23 を格納 |
| PDIn[2] | 0-7 | Sensor Value(bit8-15) | センサー値の Bit8-15 を格納 |
| PDIn[3] | 0-7 | Sensor Value(bit0-7) | センサー値の Bit0-7 を格納 |
| PDIn[4] | 0-7 | - | (不使用) |
| PDIn[5] | 1-6 | - | (不使用) |
| | 0 | Switching Signal | Switching Signal 状態を格納 |

センサー値が 0x00010B4F、Switching Signal 状態が 1 の場合のプロセスデータの例を表 5-2 に示します。

表 5-2 プロセスデータの例

| PDIn[0] | | | | | | | | |
|---------|---|---|---|---|---|---|---|---|
| 値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| PDIn[1] | | | | | | | | |
| 値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| PDIn[2] | | | | | | | | |
| 値 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| PDIn[3] | | | | | | | | |
| 値 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| PDIn[4] | | | | | | | | |
| 値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| PDIn[5] | | | | | | | | |
| 値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

5.4.4 パラメータ

IO-Link でマスターと送受信するパラメーター一覧を表 5-3 に記載します。

表 5-3 設定パラメーター一覧

| 名称 (型) | ビット 数 | 値域 (初期値) | R/W | 単位 | 概要 |
|---|----------|--|-----|----|---|
| Switching Signal Channel1 | | | | | |
| Param SP1 (unsigned int) | 32 | 0x00000000 to 0x00FFFFFF (55000) 注2 | RW | - | Switching Point1 (SP1) 設定 ^{注1} SP1 ≥ SP2 を満たす必要があります。 |
| Param SP2 (unsigned int) | 32 | 0x00000000 to 0x00FFFFFF (45000) 注2 | RW | - | Switching Point2 (SP2) 設定 ^{注1} SP1 ≥ SP2 を満たす必要があります。 |
| Config Logic (unsigned char) | 8 | 0, 1 (0) | RW | - | スイッチング状態論理設定 0 : High active ON であれば 1 を送信 1 : Low active ON であれば 0 を送信 |
| Config Mode 注3 (unsigned char) | 8 | 0, 1, 2, 3 (0) | RW | - | スイッチング状態判定モード設定 0 : Deactivated 無効、スイッチング状態は常に OFF 1 : Single point 2 : Window 3 : Two point |
| Hysteresis 注4 (unsigned sort) | 16 | 0x0000 to 0xFFFF (0) | R/W | - | ヒステリシス |
| Teach-In Result : state (unsigned char) | 8 | 0, 1, 2, 3, 4, 5, 7 (0) | RO | - | 前回の Teach-In コマンドの結果 0 : Idle, 1 : SP1 Success, 2 : SP2 Success, 3 : SP12 Success, 4 : Wait, 5 : Busy, 7 : Error |

- 【注】
- 「5.3 動作モードとスイッチング状態」を参照してください。
 - Device Tool では 0~16777215(0xFFFFF)の範囲で設定します。表示イメージについては「7.6.5 Parameter タブ」を参照してください。
 - デフォルトは Deactivated(無効)です。スイッチング判定を使用する場合は Window または Two point に変更してください。Single point は本サンプルアプリケーションでは非サポートの為、選択しないでください。
 - 本サンプルアプリケーションでは非サポートです。使用しないでください。

5.5 使用する周辺機能と端子

本サンプルアプリケーションで使用する周辺機能一覧を表 5-4 に、使用端子一覧を表 5-5 に記載します。

表 5-4 使用する周辺機能一覧

| 周辺機能 | 用途 |
|--------------|---|
| SCI0(簡易 SPI) | ZIOL2401 通信用 SPI Master(IO-Link スタックにて使用) |
| SCI1(UART) | PC(ZSSC GUI)との UART 通信 |
| IIC | ZSSC 通信用 IIC Master |
| AGT0 | IO-Link 通信用の周期タイマー(IO-Link スタックにて使用) |

表 5-5 使用端子一覧

| 端子名 | 入出力 | 用途 |
|------------|-----|---|
| P411/IRQ4 | 入力 | IO_INT(LOW で割込み発生) |
| P410 | 入力 | IO-Link の WakeUp 検出信号入力端子 |
| P409/IRQ6 | 入力 | ZIOL2401 の DC/DC Ready 信号入力端子 |
| P408/SCL | 出力 | ZSSC 通信用 IIC クロック端子 |
| P407/SDA | 入出力 | ZSSC 通信用 IIC データ端子 |
| P913 | 出力 | EK ボード LED3 の制御 |
| P914 | 出力 | EK ボード LED2 の制御 |
| P401/TXD | 出力 | PC(ZSSC GUI)通信用 UART TXD |
| P402/RXD | 入力 | PC(ZSSC GUI)通信用 UART RXD |
| P304 | 出力 | ZSSC KS5V ON/OFF (L 出力で ON) |
| P303 | 出力 | ZSSC KS12V ON/OFF (L 出力で ON) |
| P302 | 出力 | IO-Link IO_AUX_TX |
| P301 | 出力 | IO-Link IO_AUX_EN |
| P106 | 出力 | ZIOL2401 のリセット制御端子 (L 出力でリセット解除) |
| P105 | 出力 | ZSSC のリセット制御端子 (L 出力でリセット解除) |
| P104/IRQ1 | 入力 | ZSSC の EOC(End-Of-Conversion signaling) |
| P103/SSL0 | 出力 | ZIOL2401 の SPI 通信有効/無効制御端子 (L 出力で有効) |
| P102/SCK0 | 出力 | ZIOL2401 の SPI 通信 SCK0 |
| P101/MOSI0 | 出力 | ZIOL2401 の SPI 通信 MOSI0 |
| P100/MISO0 | 入力 | ZIOL2401 の SPI 通信 MISO0 |

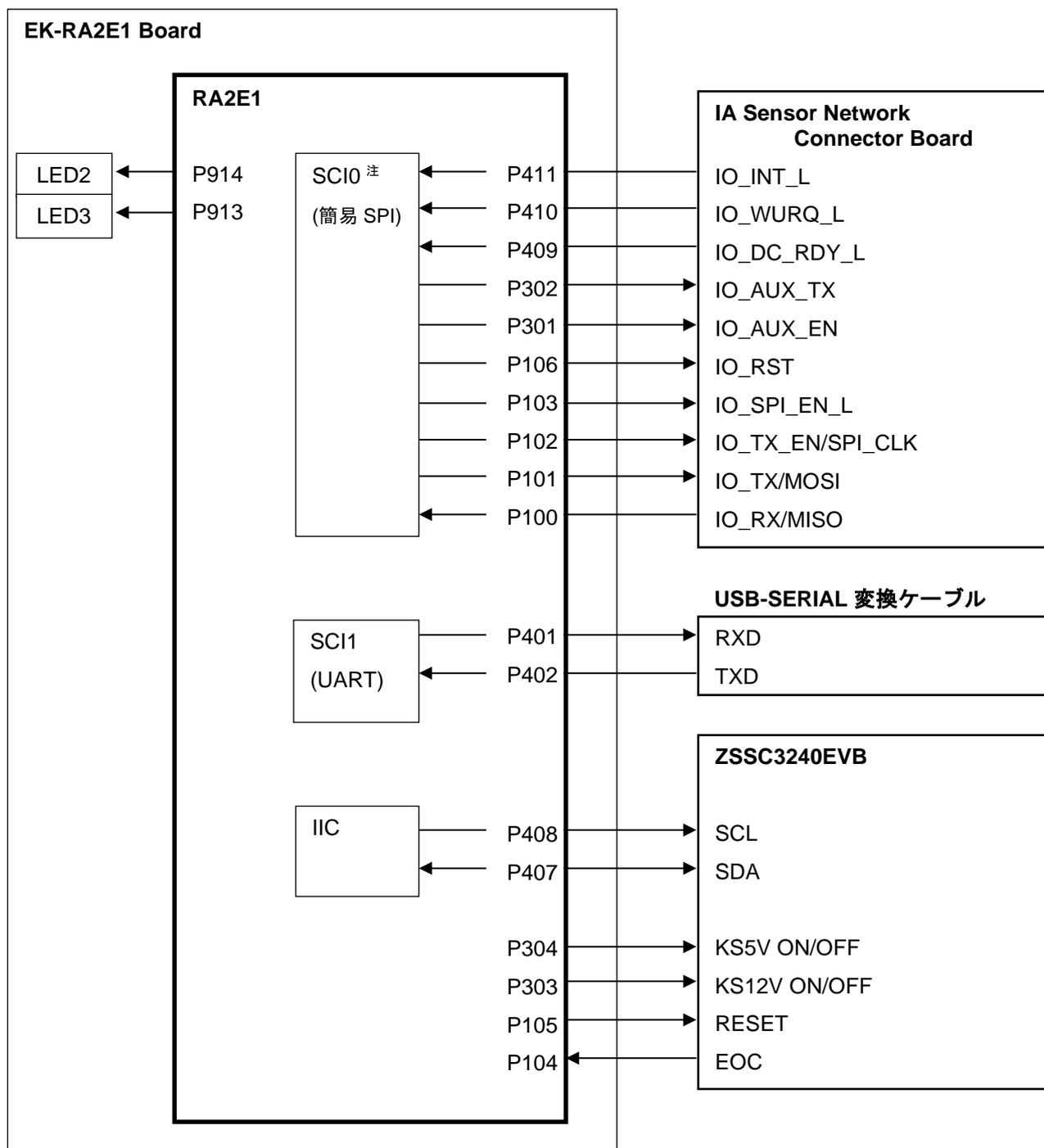


図 5-8 使用する周辺機能と端子

【注】 SCI0 簡易 SPI は、IO-Link スタックライブラリが提供する独自ドライバで制御します。FSP が提供する SCI.SPI ドライバは使用しません。

5.6 プログラム構成

5.6.1 ファイル構成

TMG 社製 IO-Link スタックと、これに関係するファイルは Library フォルダ内に配置されています。IO-Link スタックのマニュアルは Manuals フォルダ内に配置されています。IO-Link フォルダ内のファイル構成を表 5-6 に記載します。

表 5-6 ファイル構成

| フォルダ名、ファイル名 | 説明 |
|---|---|
| IO-Link | |
| ├─Application | アプリケーションフォルダ |
| │ application.c | アプリケーションプログラム |
| │ application.h | アプリケーションヘッダ |
| │ define.h | Define ヘッダ |
| │ global.h | グローバルヘッダ |
| │ i2c.c | I2C プログラム |
| │ i2c.h | I2C ヘッダ |
| │ main.c | アプリケーション main |
| │ r_led_api.c | LED 制御プログラム |
| │ r_led_api.h | LED 制御ヘッダ |
| │ spi.c | SPI 処理プログラム |
| │ spi.h | SPI ヘッダ |
| │ Starterkit_Config.h | Starterkit_Config.h |
| │ tool.c | tool 関数 |
| │ tool.h | tool 関数ヘッダ |
| │ uart.c | UART コマンド処理プログラム |
| │ uart.h | UART コマンド処理ヘッダ |
| │ utils.h | ユーティリティヘッダ |
| │ zssc.c | ZSSC 処理プログラム、ZSSC 計測処理 |
| │ zssc.h | ZSSC ヘッダ |
| | |
| ├─BSP | IO-Link に関連しない、アプリケーション部分の BSP フォルダ |
| │ BSPStack.h | User-implemented Hardware Abstraction definitions |
| │ IOLD_Config.h | system specific Defines |
| │ SystemInit.h | Hardware settings definitions |
| │ ZIOL2401.h | Functions for ZMDI L2401 |
| | |
| ├─IODD | IO-Link デバイスの IODD ファイル格納フォルダ |
| │ TMG-logo.png | TMG ロゴ画像ファイル |
| │ TMG-RA2E1-Starterkit-20210715-IODD1.0.1.xml | IODD1.0.1 ファイル |
| │ TMG-RA2E1-Starterkit-20210715-IODD1.1.xml | IODD1.1 ファイル |
| │ TMG-RA2E1-Starterkit-con-pic.png | M12 コネクタ 4Pin 画像ファイル |
| │ TMG-RA2E1-Starterkit-icon.png | ボードアイコン画像ファイル |
| │ TMG-RA2E1-Starterkit-pic.png | ボード画像ファイル |
| | |
| ├─Library | IO-Link スタックライブラリ、パラメータセット格納フォルダ |

| | | |
|--|--|--|
| | BSPInterface.h | Defines for the BSP |
| | DeviceAccess.h | Device Access protection |
| | DeviceStack.h | IO-Link Device Stack |
| | DStorage.h | DataStorage for the DeviceStack |
| | DTypes.h | Datatype Definitions |
| | EventDispatcher.h | functions for EventDispatching |
| | libIO_Link_Starterkit_RA2E1_lib.a | IO-Link Library for the starterkit solution |
| | ParameterManager.h | Interface to the application framework |
| | Profile_common.h | common Declarations for Profiles |
| | | |
| | └─ Manuals | マニュアルフォルダ |
| | TMG IO-Link Device Integration Library Manual.pdf | IO-Link Device Evaluation Library |
| | | |
| | └─ StackExtensionsApp | アプリケーションの IO-Link 関連部分格納フォルダ |
| | BSPExtensions.h | definitions for Stack Extensions Board Support Package |
| | IOLinkMain.c | IO-Link Application |
| | IOLinkMain.h | definitions for main program |
| | MemoryManager.h | definitions for Stack Extensions Memory Manager |
| | ParameterSet.h | Definitions for Stack Extensions |
| | ProductionSettings.h | definitions for Stack Extensions Production Settings |

5.6.2 関数一覧

以下では、「5.1 全体処理フロー概要」で紹介の関数のみ、詳細を記します。その他の関数については、同梱のサンプルアプリケーションプロジェクトのソースコードをご参照ください。

5.6.2.1 main.c

[関数名] iol_main

| | |
|-------|--|
| 概要 | iol_main 関数 |
| ヘッダ | なし |
| 宣言 | int iol_main(void) |
| 説明 | TMG 社から提供された iol_main() にアプリケーション固有の処理を追加して構成しています。IO-Link stack ライブラリやアプリケーションで使用する周辺機能などの初期化処理を実施後に周期処理に移行します。 |
| 引数 | なし |
| リターン値 | なし |
| 備考 | なし |

5.6.2.2 IOLinkMain.c

[関数名] iolink_main

| | |
|-------|--|
| 概要 | IO-Link メイン処理 |
| ヘッダ | IOLinkMain.h |
| 宣言 | TUnsigned8 iolink_main(void) |
| 説明 | STACK_Run 関数をコールします。IO-Link スタックを周期的に呼び出し、ステータスを取得します |
| 引数 | なし |
| リターン値 | IO-Link スタックの状態 STACK_STATUS_SIO : IO-Link 接続は SIO モード STACK_STATUS_STARTUP : マスターが検出され、デバイスはスタートアップ状態 STACK_STATUS_PREOPERATE : デバイスは動作前状態 STACK_STATUS_OPERATE : デバイスは動作状態 STACK_STATUS_DISCONNECTED : 接断状態、デバイスは IO-Link モードで次のウェイクアップを待機 |
| 備考 | なし |

5.6.2.3 application.c

[関数名] apl_main

| | |
|-------|---|
| 概要 | アプリケーションメイン |
| ヘッダ | application.h |
| 宣言 | void apl_main(void) |
| 説明 | UART_Command_Check()をコールし UART コマンドの受信を確認します。 UART コマンド受信時は UART_Command_Execute()をコールして UART コマンド処理を行います。 UART コマンド受信が 1 秒間無かった場合は zssc_measurement_reset()をコールし、その後 ZSSC 計測処理に移行します。UART コマンド受信が無い限り zssc_measurement()を周期的にコールして ZSSC の計測を行います。 UART コマンドを受信した場合は直ちに ZSSC 計測処理を終了します。 |
| 引数 | なし |
| リターン値 | なし |
| 備考 | なし |

[関数名] App_SensorValue

| | |
|-------|---|
| 概要 | センサー値を使用した IO-Link アプリケーション処理 |
| ヘッダ | なし |
| 宣言 | static bool App_SensorValue (TUnsigned32 sensorValue) |
| 説明 | 以下の 3 つの処理を実行します。 <ol style="list-style-type: none">1. プロセスデータの作成2. Teaching 処理 (閾値の設定処理)3. スイッチポイント処理 |
| 引数 | 計測値 |
| リターン値 | スイッチング判定結果 false : 有効範囲外 (OFF) true : 有効範囲内 (ON) |
| 備考 | 本関数は、apl_main 関数から呼び出される内部関数です。 Teaching 処理実行中は、戻り値は false を返します。 |

5.6.2.4 uart.c

[関数名] UART_Command_Check

| | |
|-------|---|
| 概要 | UART コマンド受信チェック |
| ヘッダ | uart.h |
| 宣言 | uint8_t UART_Command_Check(uint8_t *buffer) |
| 説明 | ZSSC GUI からコマンドを受信しているかどうかを確認します。 |
| 引数 | buffer バッファ先頭ポインタ |
| リターン値 | 受信状態 0 : コマンド未受信 1 : コマンド受信 |
| 備考 | なし |

[関数名] UART_Command_Execute

| | |
|-------|---|
| 概要 | UART コマンド実行 |
| ヘッダ | uart.h |
| 宣言 | void UART_Command_Execute(uint8_t *buffer) |
| 説明 | ZSSC GUI から受信したコマンドを実行します。計測コマンドを実施した場合は計測結果をグローバル変数 sensor_value に格納します。 |
| 引数 | buffer バッファ先頭ポインタ |
| リターン値 | なし |
| 備考 | なし |

5.6.2.5 zssc.c

[関数名] zssc_measurement

| | |
|-------|---|
| 概要 | ZSSC 計測 |
| ヘッダ | zssc.h |
| 宣言 | bool zssc_measurement(void) |
| 説明 | AA _{HEX} コマンドで ZSSC 計測を行い、計測結果をグローバル変数 sensor_value に格納します。 |
| 引数 | なし |
| リターン値 | true |
| 備考 | AA _{HEX} コマンドについては以下の資料を参照してください。 ・ ZSSC3240 Evaluation Kit User Manual ・ ZSSC3240 DataSheet |

5.6.2.6 r_led_api.c

[関数名] R_LED_Control

| | |
|-------|--|
| 概要 | LED の制御 |
| ヘッダ | r_led_api.h |
| 宣言 | void R_LED_Control(uint16_t led_pin, bool state) |
| 説明 | LED を点灯/消灯します。 |
| 引数 | led_pin LED Pin 番号 |
| 引数 | state LED 状態 |
| | false : OFF |
| | true : ON |
| リターン値 | なし |
| 備考 | なし |

6. サンプルプロジェクト実行方法

本章では、サンプルプロジェクトのインポートからプログラム実行までの手順を解説します。

6.1 サンプルプロジェクトのインポート

1. e² studio を起動してください。
2. Workspace ダイアログボックスに新規ワークスペース名を入力してください。その後 Launch をクリックしてください。

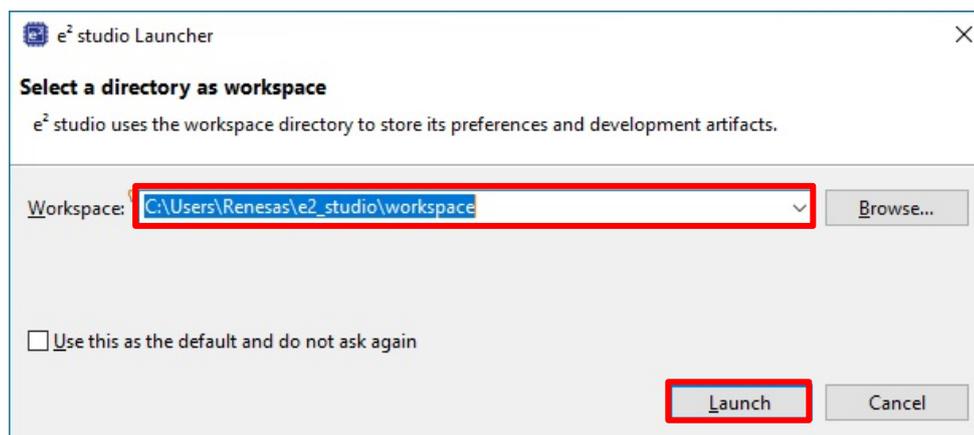


図 6-1 新規ワークスペースの使用開始

3. Import existing projects をクリックしてください。

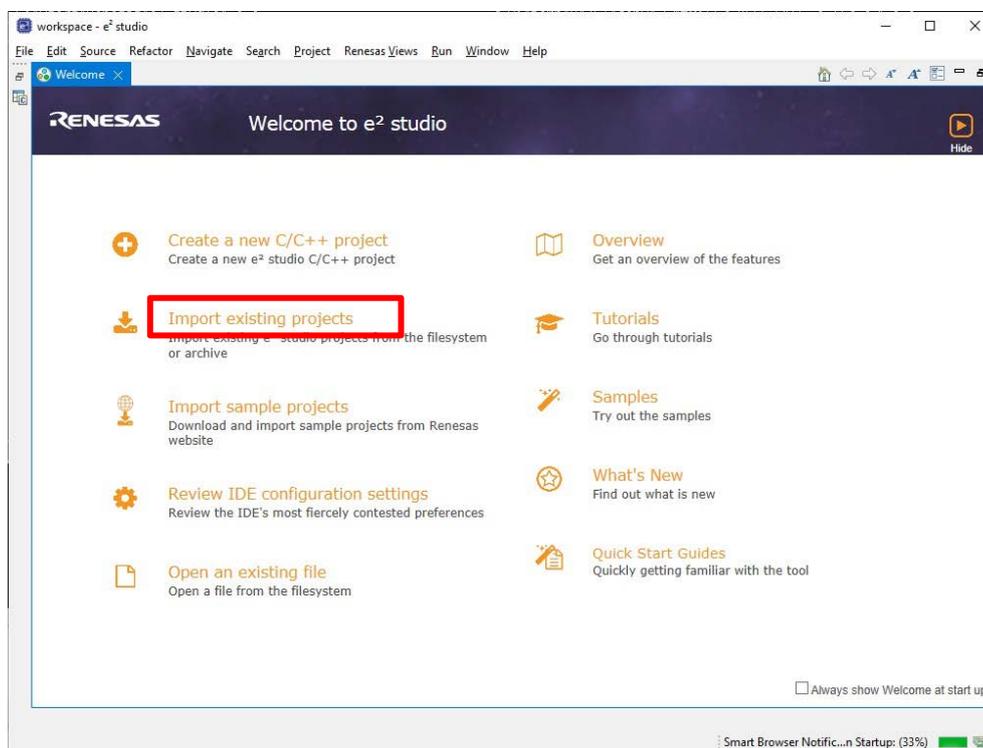


図 6-2 Welcome to e²studio

4. Select archive file をクリックして、次いで Browse をクリックし、サンプルプロジェクトの zip ファイルのあるロケーションを開きます。
5. サンプルプロジェクトの zip ファイルを選択し、Finish をクリックしてください。

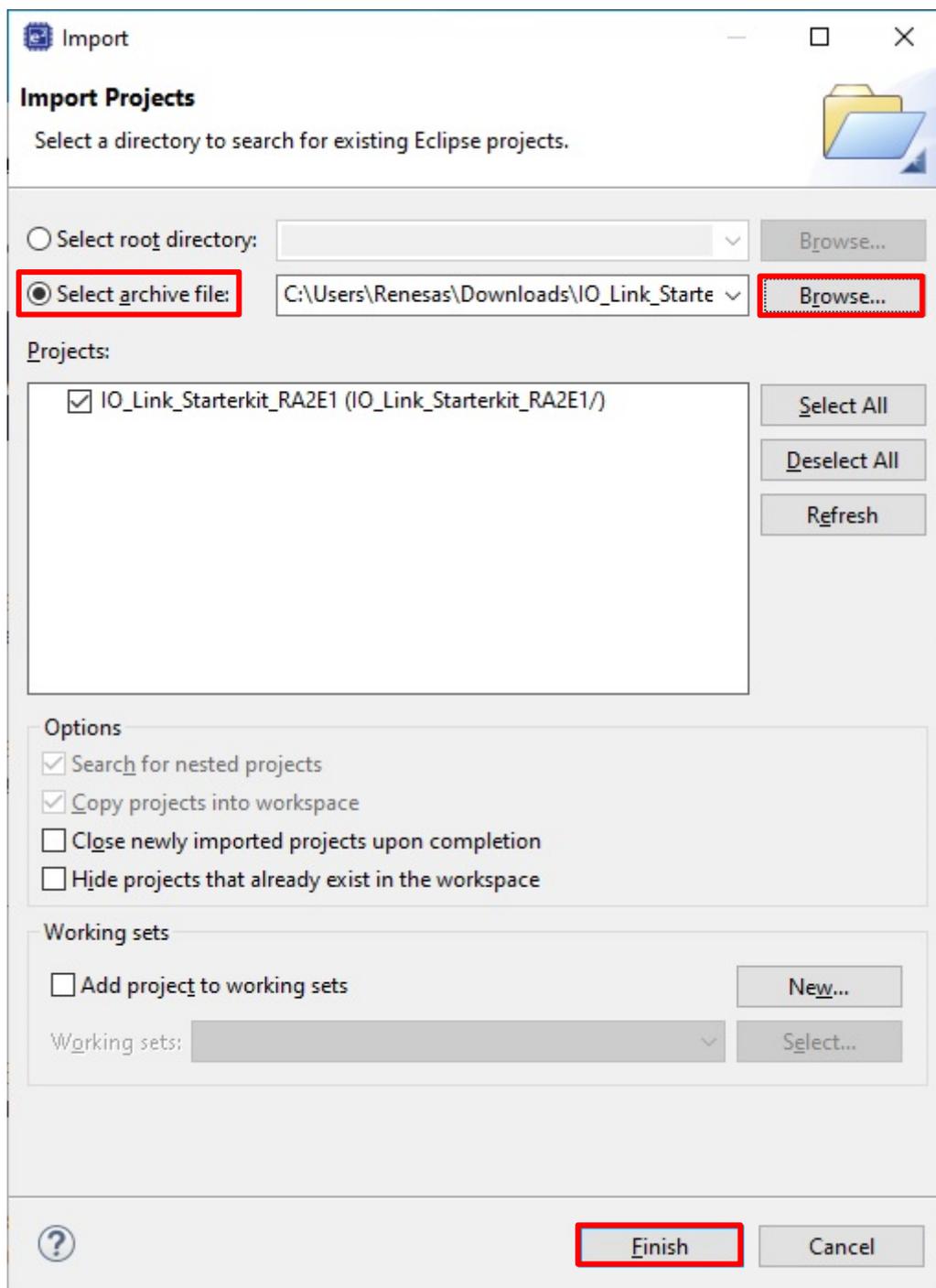


図 6-3 サンプルプロジェクトのインポート

6.2 プロジェクトのビルド

1. サンプルプロジェクトがインポートされたら、configuration.xml をダブルクリックしてコンフィグレータを開きます。

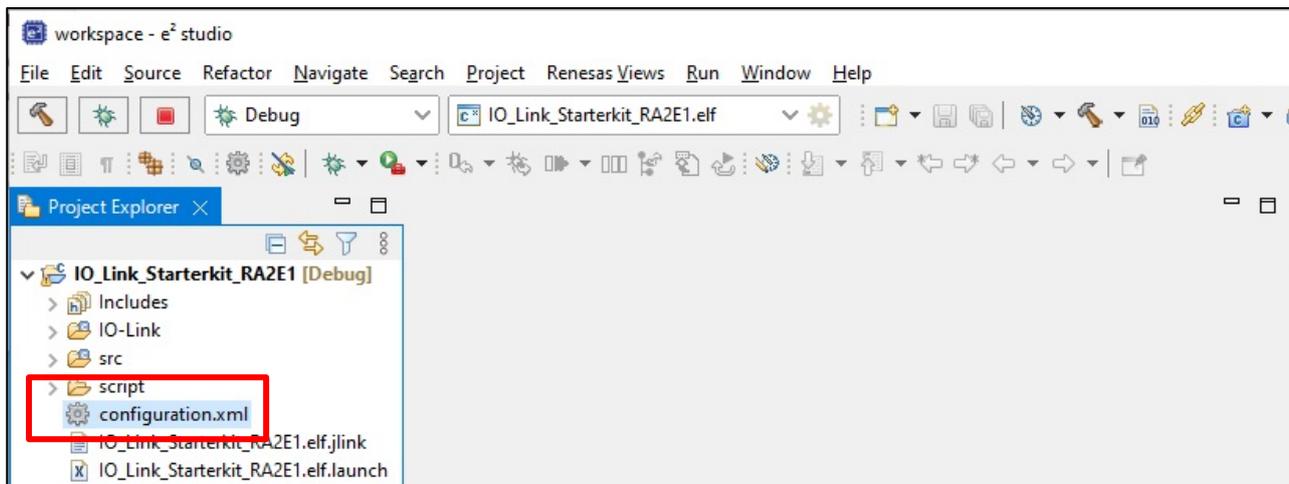


図 6-4 コンフィグレータの起動

2. Stacks タブの Generate Project Content をクリックしてください。コンフィギュレータは必要なファイルを生成し、プロジェクトに追加を行います。

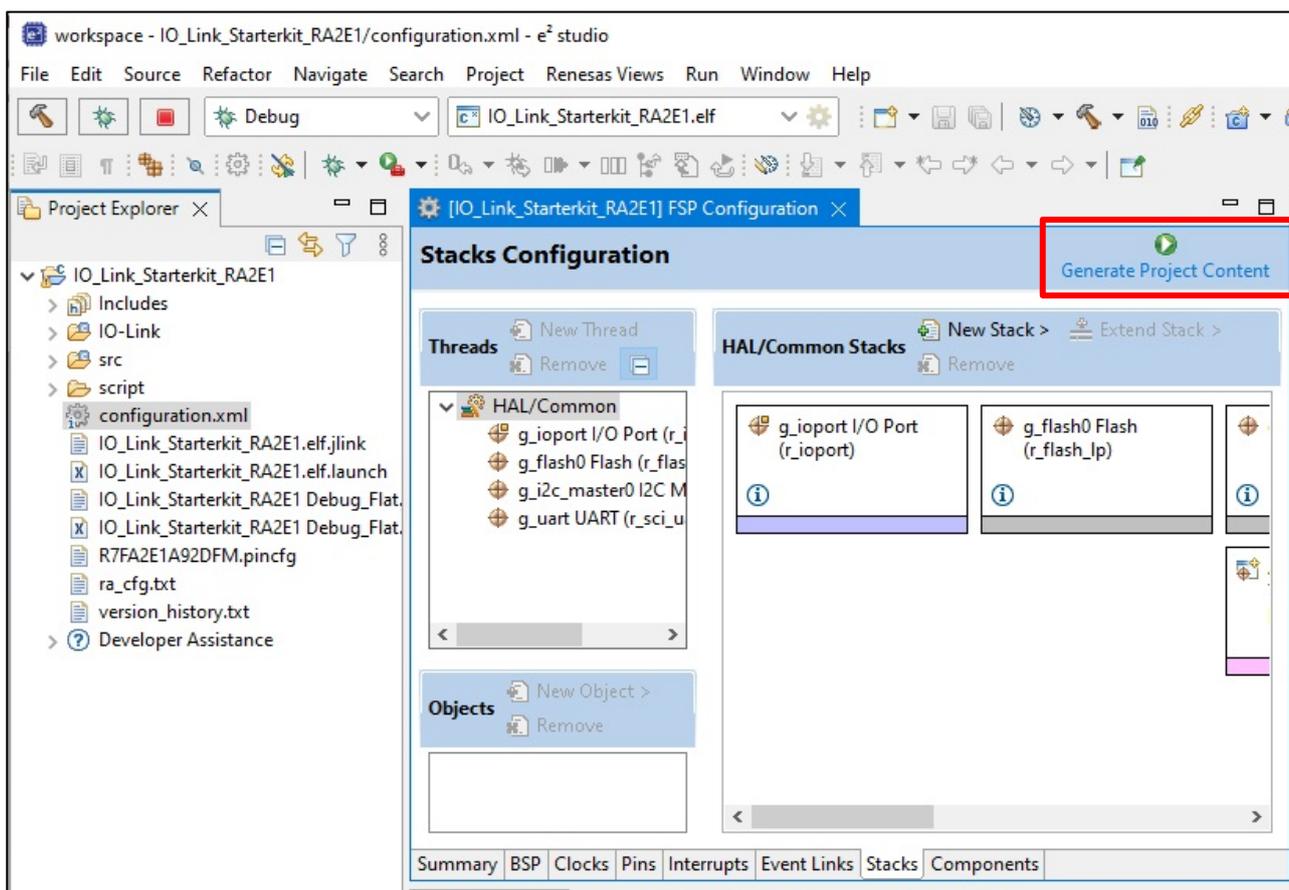


図 6-5 プロジェクトコンテンツの生成

3. Build アイコンをクリックして、プロジェクトをビルドしてください。

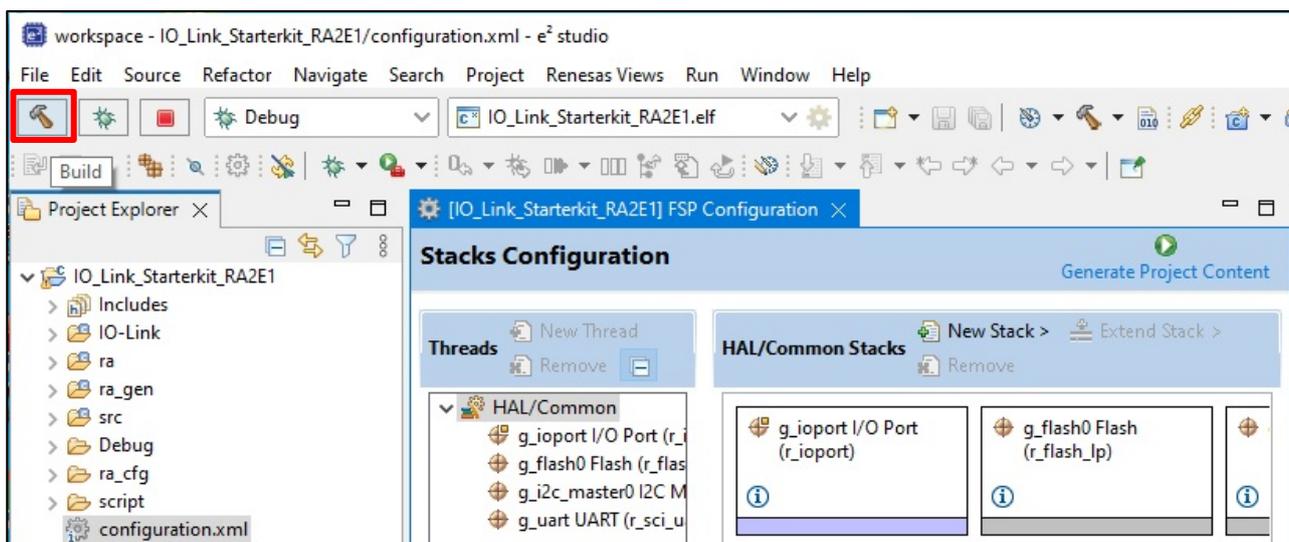


図 6-6 プロジェクトのビルド

4. ビルドが正常に終了すると、以下のような出力が生成されます。

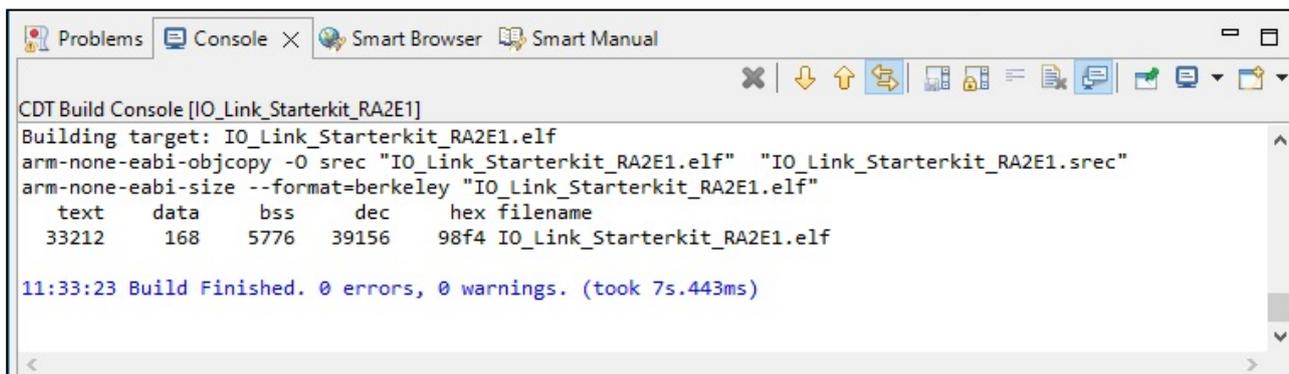


図 6-7 ビルド成功時のアウトプット

6.3 EK-RA2E1 ボードとホスト PC 間のデバッグ接続のセットアップ

以下に EK-RA2E1 ボードとホスト PC を USB ケーブルで接続し、デバッグインターフェース経由でプログラムをダウンロードする方法を示します。

1. PC からの USB ケーブルを EK-RA2E1 ボードの USB デバッグポート(J10)に接続します。
2. デバッグ LED(LED5)がオレンジ色に点灯することを確認します。

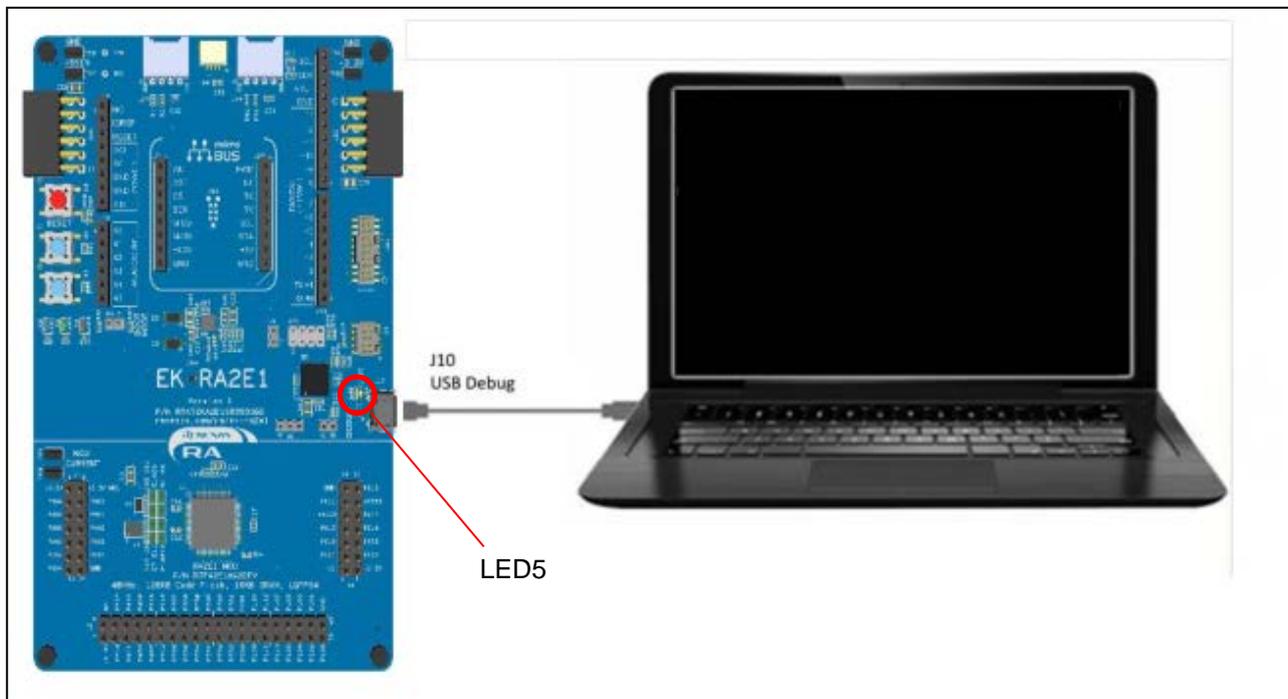


図 6-8 ボードの USB デバッグポート(J10)をホスト PC に接続

6.4 MCU へのサンプルプロジェクトの書き込み

1. Debug アイコンをクリックします。

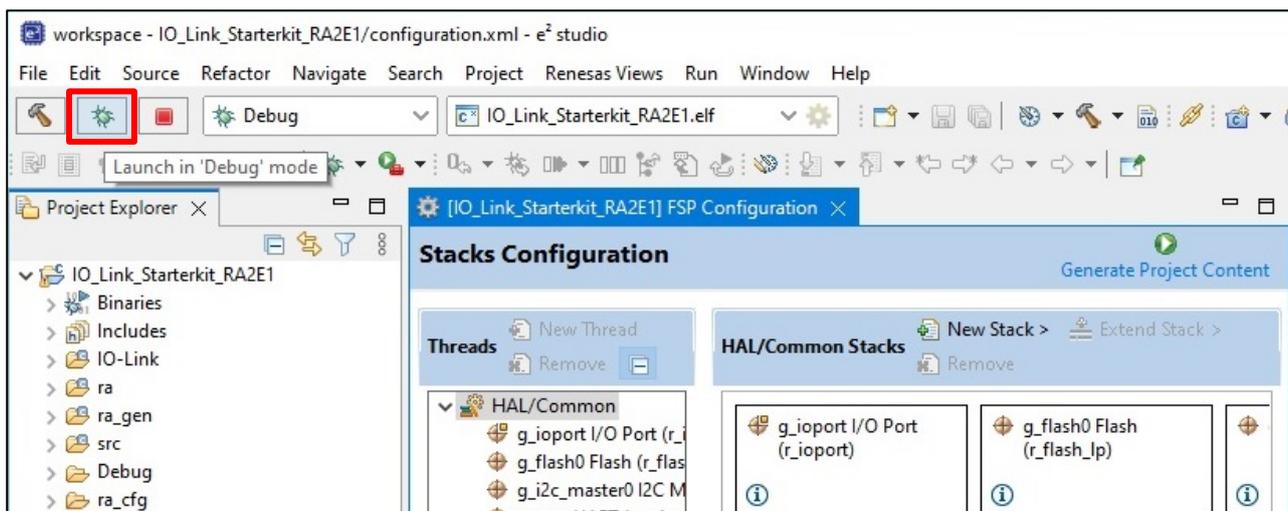


図 6-9 Launch in 'Debug' mode

- 'e2- server-gdb.exe'に対してファイアウォールの警告が表示される場合があります。'Private networks, such as my home or work network' チェックボックスをオンにして、Allow access をクリックします。

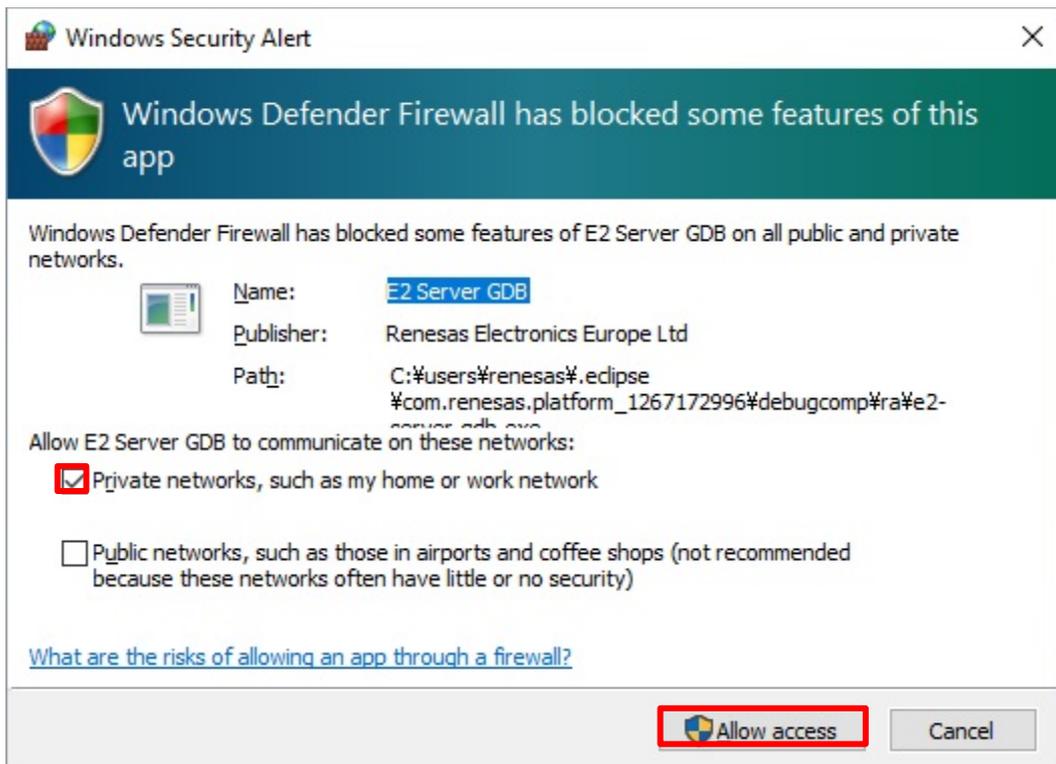


図 6-10 ファイアウォール警告

- デバッグパースペクティブへの切り替えを促すダイアログが表示される場合があります。Switch をクリックしてビューを切り替えます。

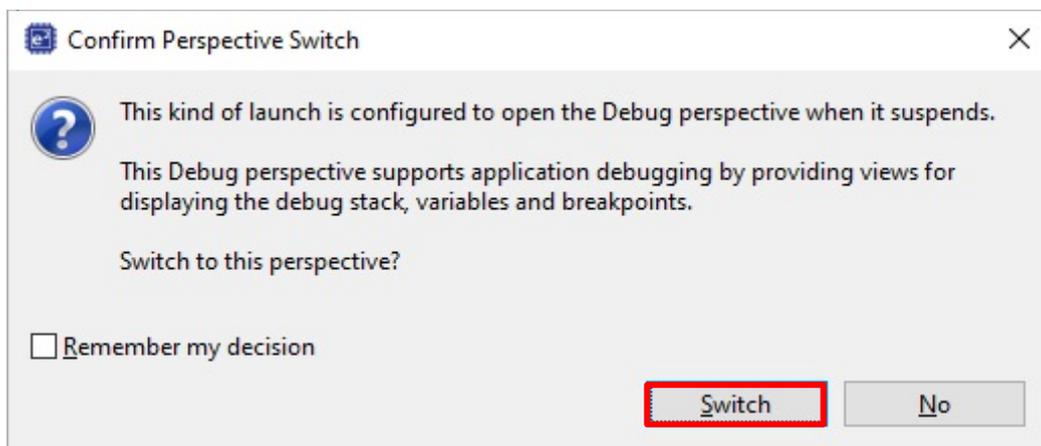


図 6-11 Confirm Perspective Switch

4. MCU へのサンプルプロジェクトの書き込みが実施され、画面が切り替わります。

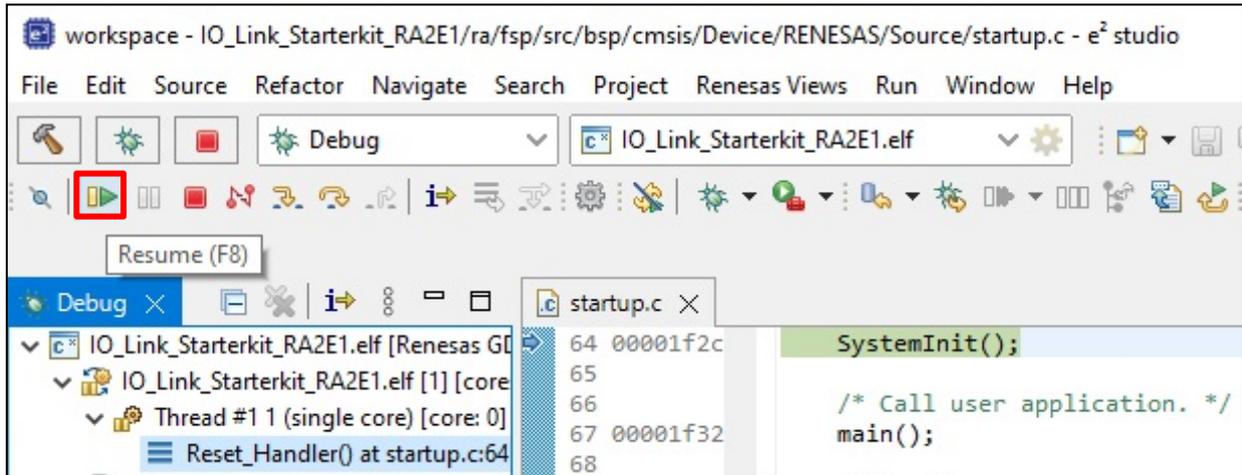


図 6-12 MCU への書き込み完了

6.5 プログラムの開始

1. 図 6-12 の状態で、F8 または Resume アイコンをクリックして、プログラムの実行を開始します。
2. 一度、main()関数の先頭で停止します。再度 F8 または Resume アイコンをクリックします。
3. プログラムが稼働状態となり、e2studio のステータスバーに'Running'と表示されます。

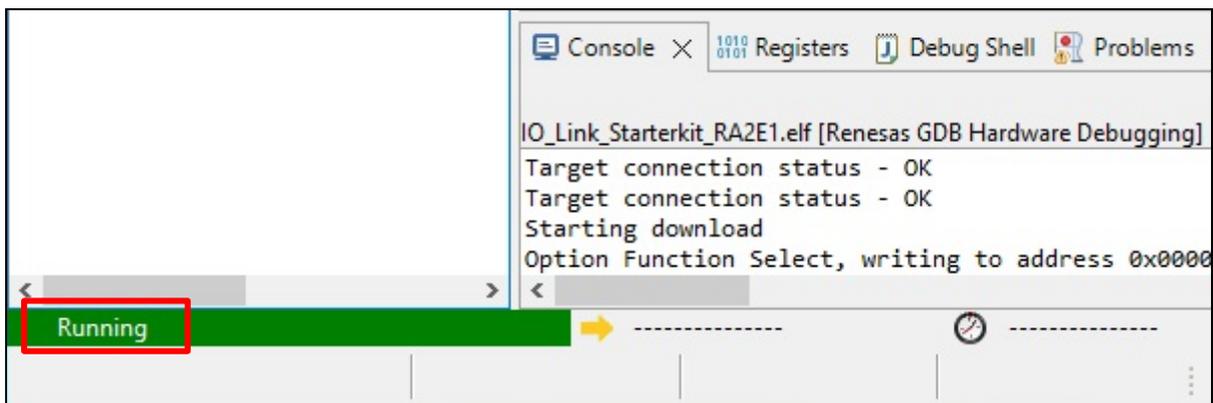


図 6-13 プログラムの開始

7. IO-Link Device Tool の使用準備及び機能解説

本章では、Windows PC に TMG 社製 IO-Link Device Tool V5.1 – PE がインストールされていることを前提として、使用準備と機能解説を記載します。ハードウェアは「4 ハードウェア構成」を参考にセットアップを行ってください。

7.1 IO-Link Device Tool V5.1 – PE を起動

以下図 7-1 に示す Topology ペインには PC から IO-Link デバイスまでのトポロジーが表示されます。

Device Catalog にはインポートが完了し使用可能になった全てのデバイスが表示されます。

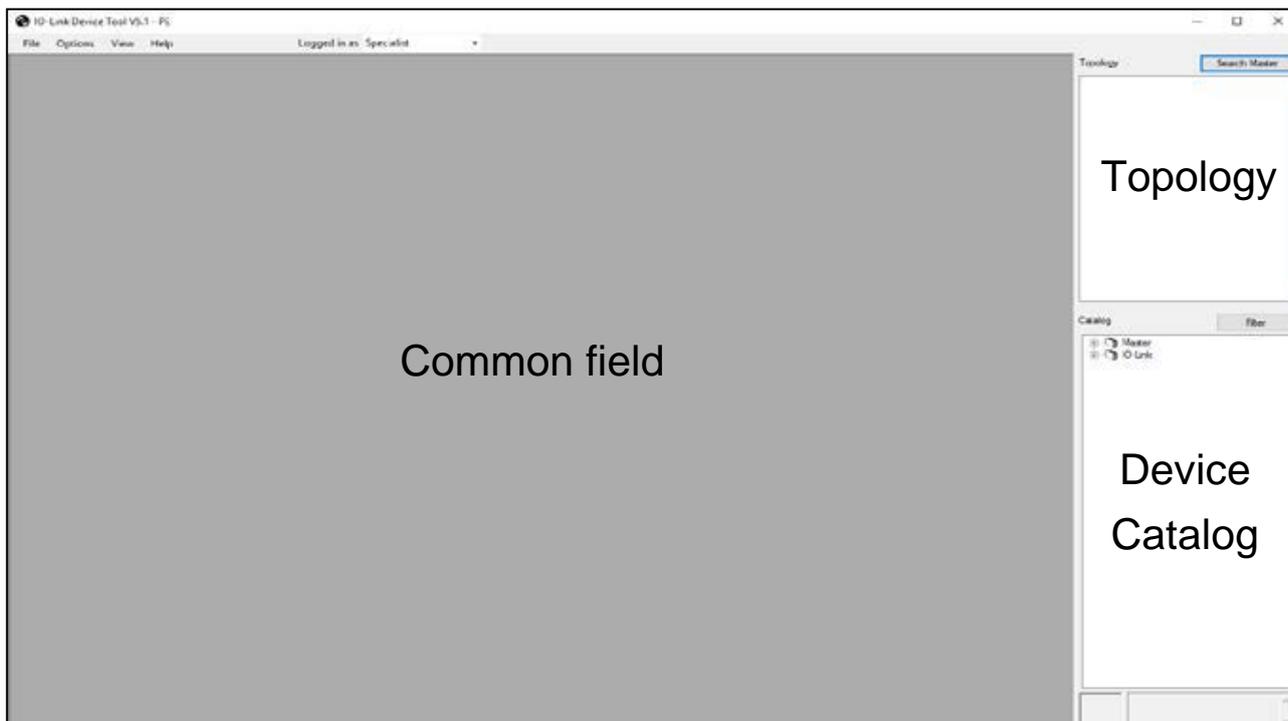


図 7-1 IO-Link Device Tool V5.1 – PE

7.2 IO-Link Device Catalog の更新

1. メニューバーで「Options」を選択し、「Import IODD(IO Device Description)」を選択します。
2. IODD ファイルがあるフォルダのパスを直接入力するか、またはブラウズして選択します。
本サンプルプロジェクトで使用する IODD ファイルは、サンプルプロジェクトの IODD フォルダに保存されています。このフォルダへのパスを図 7-3 に示すように設定することで、使用可能な IODD ファイルを自動的に検出できます。
3. 「TMG-RA2E1-Starterkit-20210715-IODD1.1.xml」をチェックします。

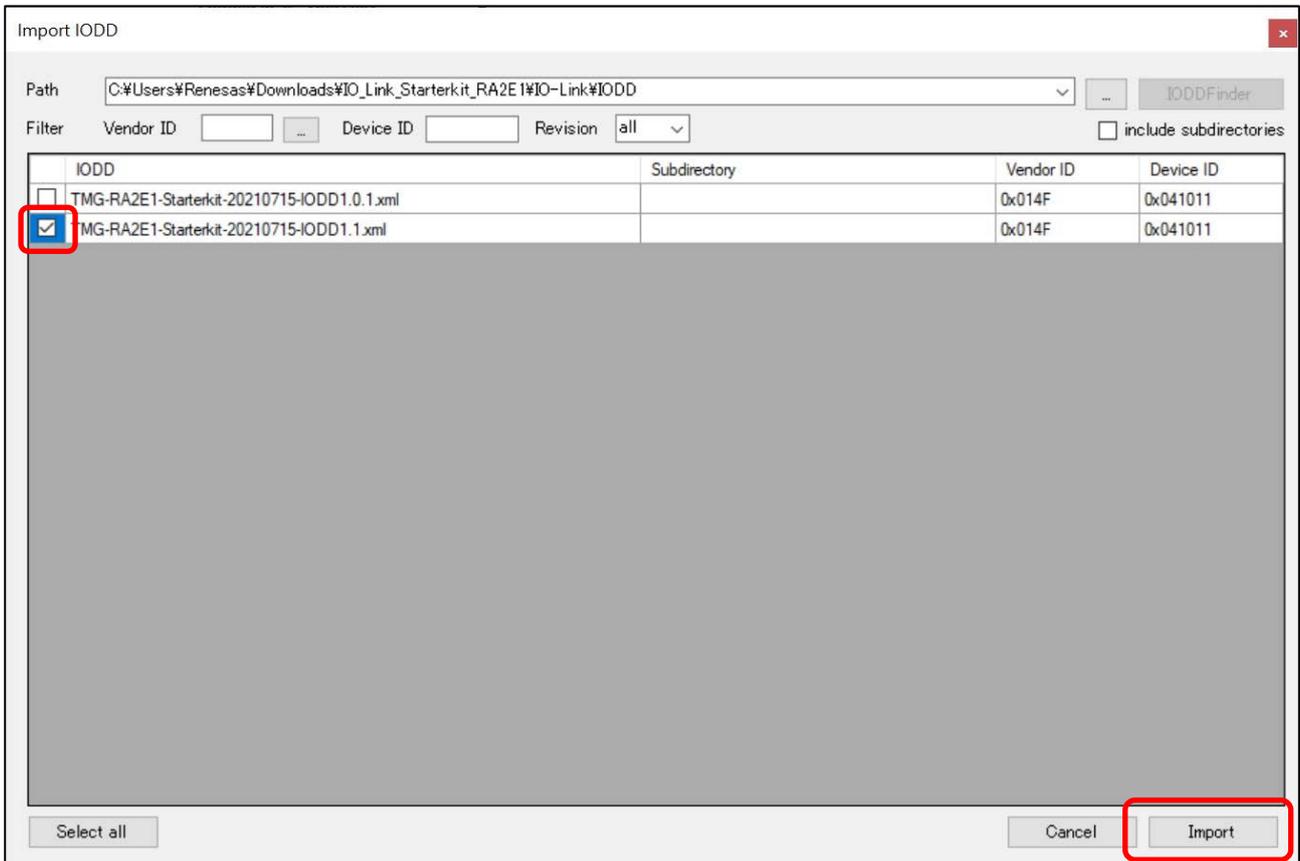


図 7-2 IO-Link Device Tool V5.1 – PE (IODD ファイルのロード)

4. 「Import」を選択します。

7.3 IO-Link Master Catalog の更新

1. メニューバーで「Options」を選択し、「Import IOLM(IO-Link Master Description)」を選択します。
2. 図 7-3 が表示されたら「TMG WEB」を選択します。

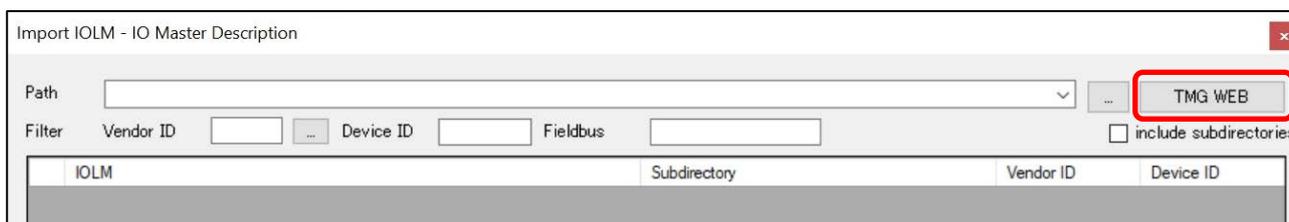


図 7-3 IO-Link Device Tool V5.1 – PE (Import IOLM)

3. 画面が切り替わり IOLM の一覧が表示されたら、Vendor ID の「...」ボタンを選択します。

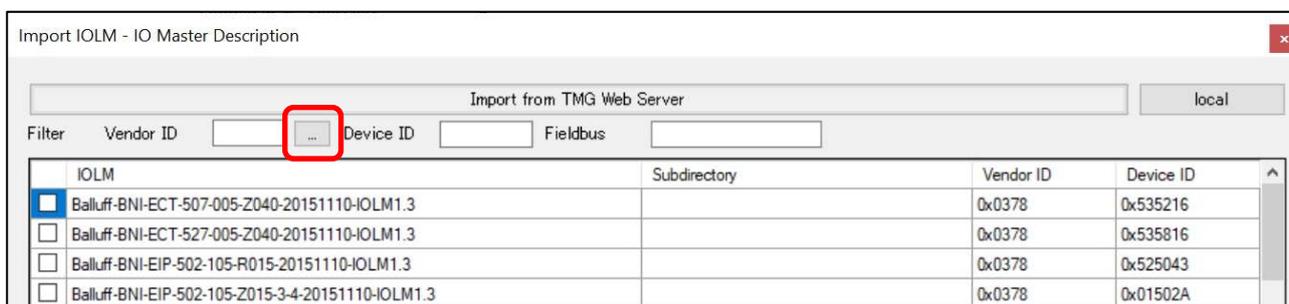


図 7-4 IO-Link Device Tool V5.1 – PE (Vendor ID)

4. Vendor ID Table が表示されたら、「Pepperl+Fuchs GmbH (0x0001)」を選択します。

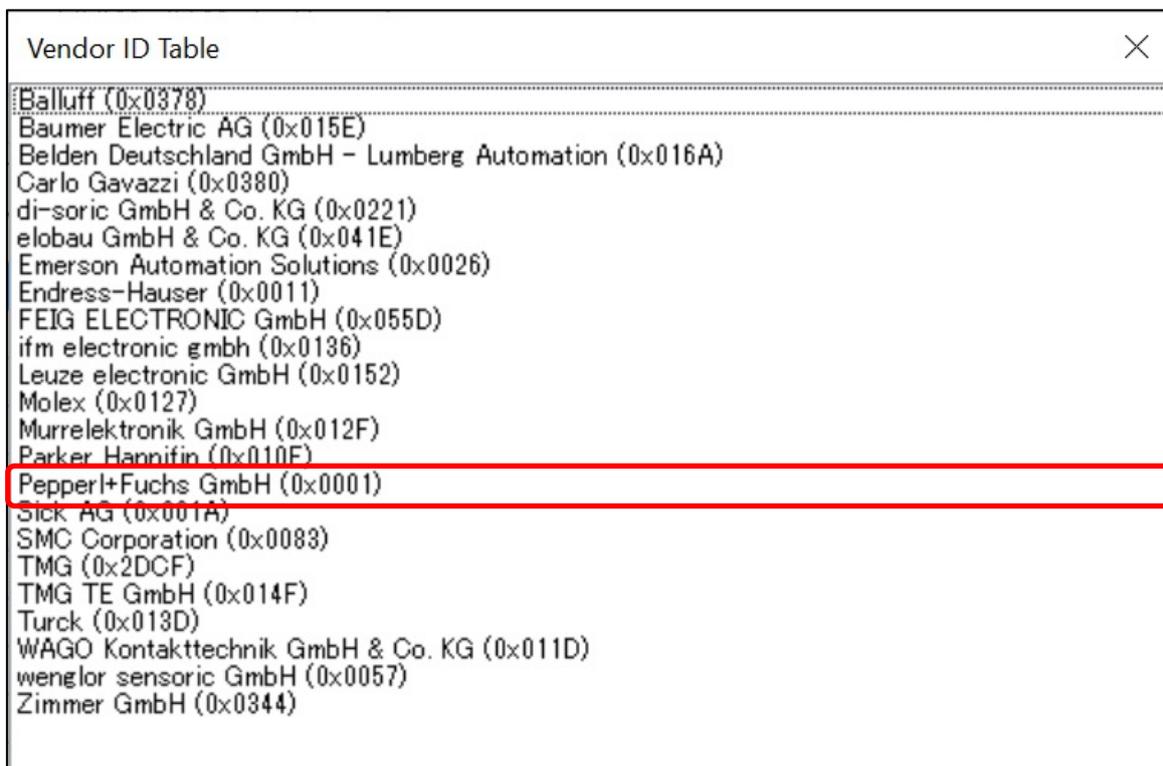


図 7-5 IO-Link Device Tool V5.1 – PE (Vendor ID Tab)

5. 使用可能な IOLM ファイルが検出されます。
6. 「Pepperl+Fuchs-IO-Link-Master02-USB-20200517-IOLM1.4」 をチェックします。

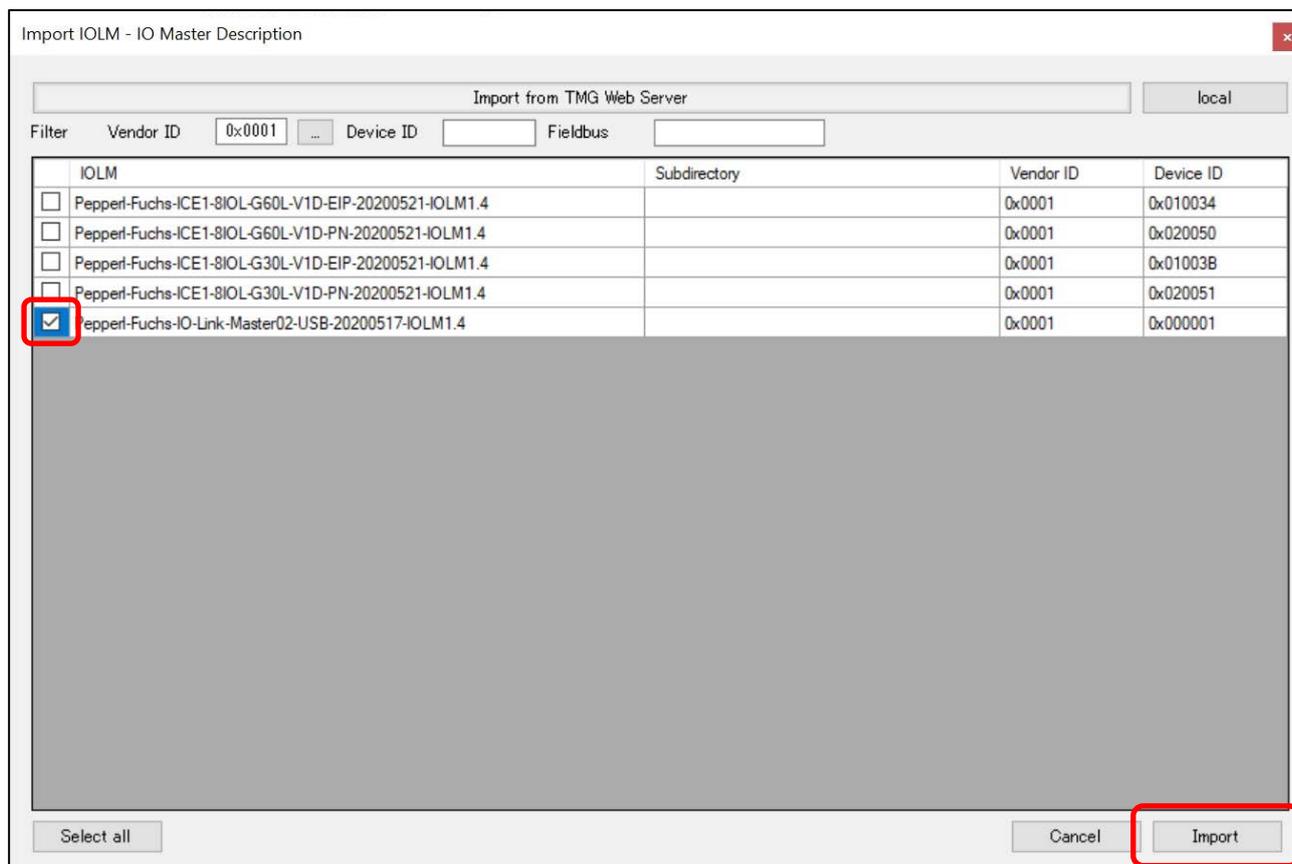


図 7-6 IO-Link Device Tool V5.1 – PE (IOLM ファイルのロード)

7. 「Import」 を選択します。

7.4 Catalog の更新の確認

7.2 章と 7.3 章の Catalog の更新が成功すると、Catalog の IO-Link Devices セクションに「RA2E1 Starterkit」として TMG TE GmbH ベンダーと EK-RA2E1 による IO-Link デバイスボードが表示されます。IO-Link-Master02-USB は Master の下に表示されます。

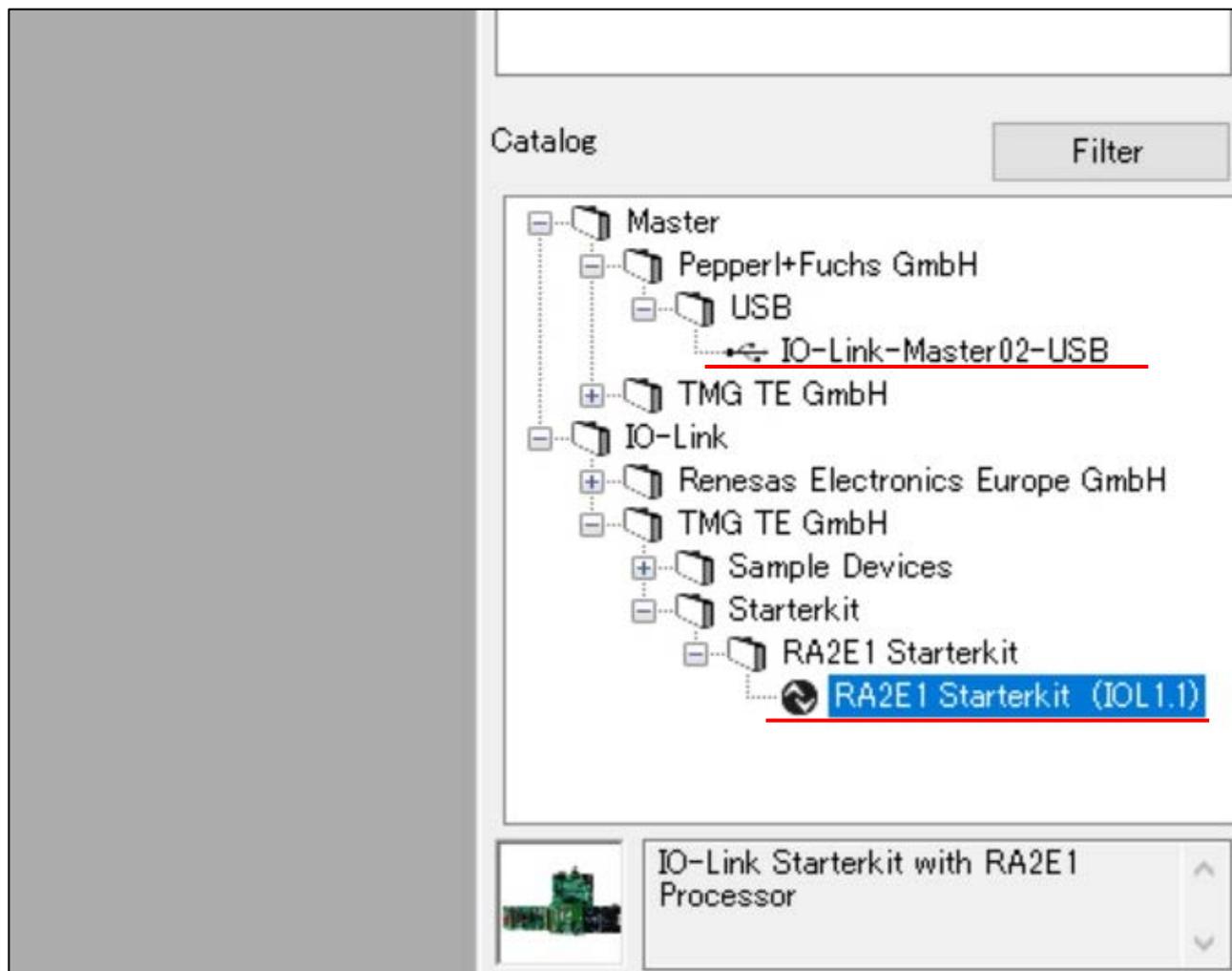


図 7-7 IO-Link Device Tool V5.1 – PE (Catalog の更新の確認)

7.5 IO-Link 通信のセットアップ

1. ウィンドウの右上にある「Search Master」ボタンをクリックします。図 7-8 のように、「IO-Link-Master02-USB」が「Master Discovery」ウィンドウに表示されます。

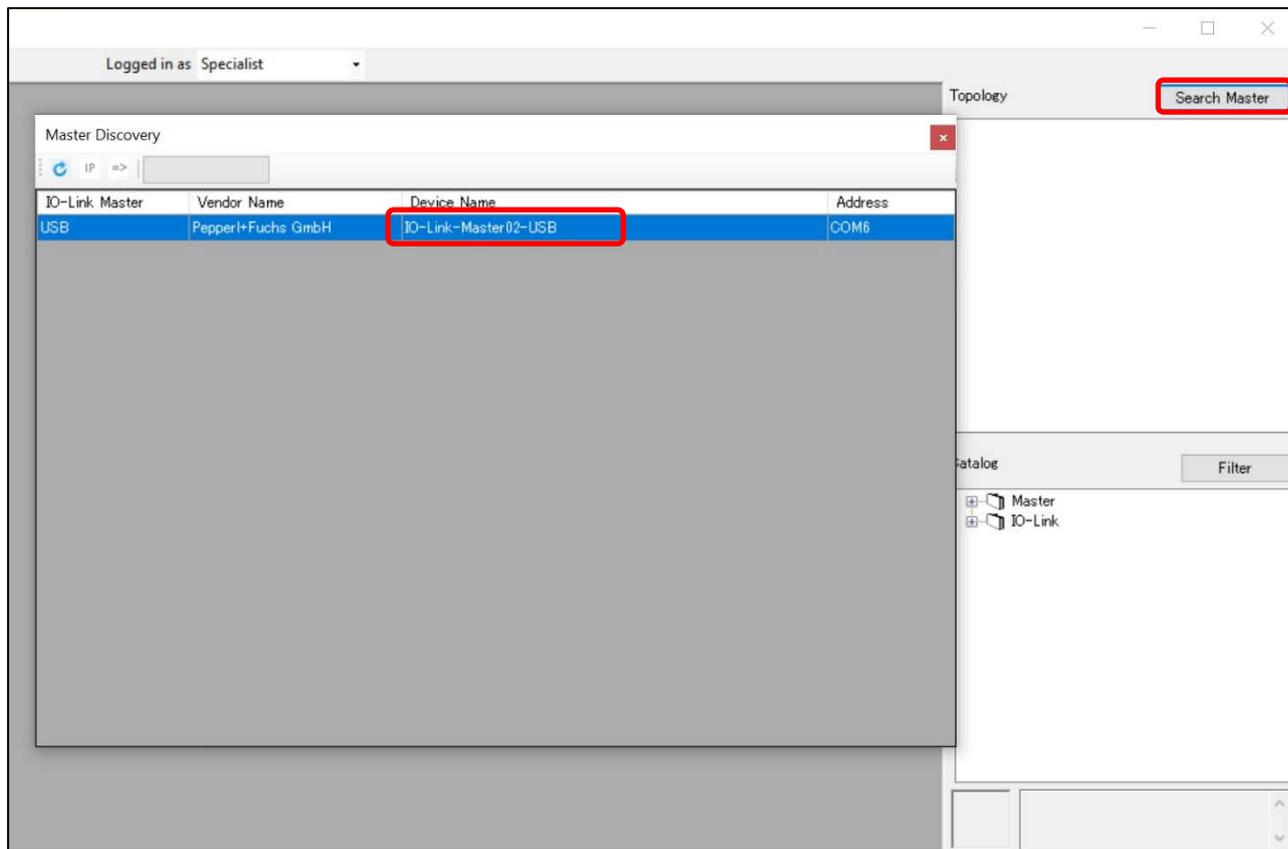


図 7-8 IO-Link Device Tool V5.1 – PE (IO-Link マスターの検索)

2. 「Master Discovery」ウィンドウに表示されているデバイス名をダブルクリックします。

3. 「Go Online」 ボタンをクリックして、マスターとデバイス間の接続をアクティブにします。

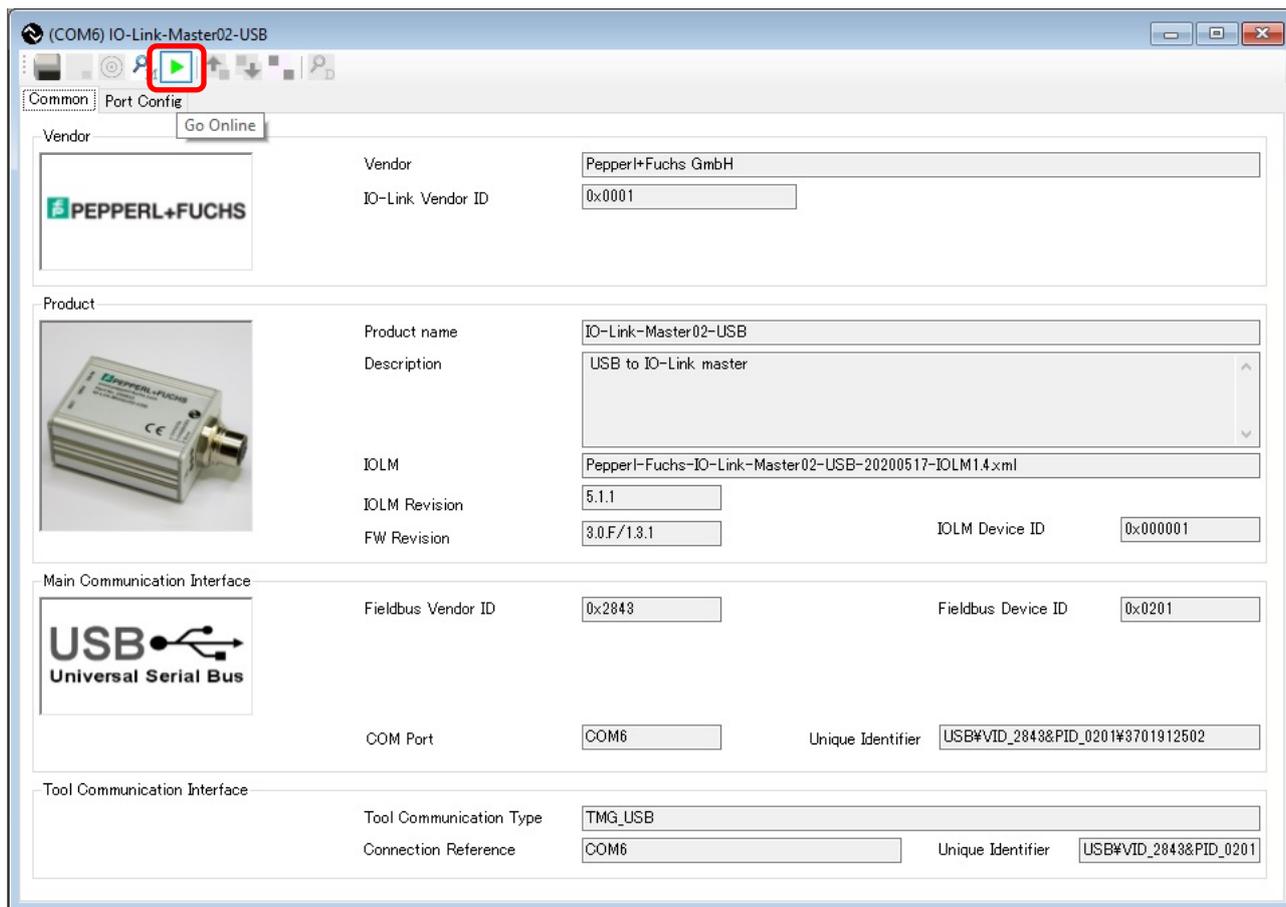


図 7-9 IO-Link Device Tool V5.1 – PE (Online 状態に設定)

マスターとデバイス間の接続がアクティブになると、「Go Online」ボタンは赤丸のボタンに置き換えられ、「Check Devices」ボタンが有効になります。

4. 「Check Devices」 ボタンをクリックして、接続されているデバイスを検出します。

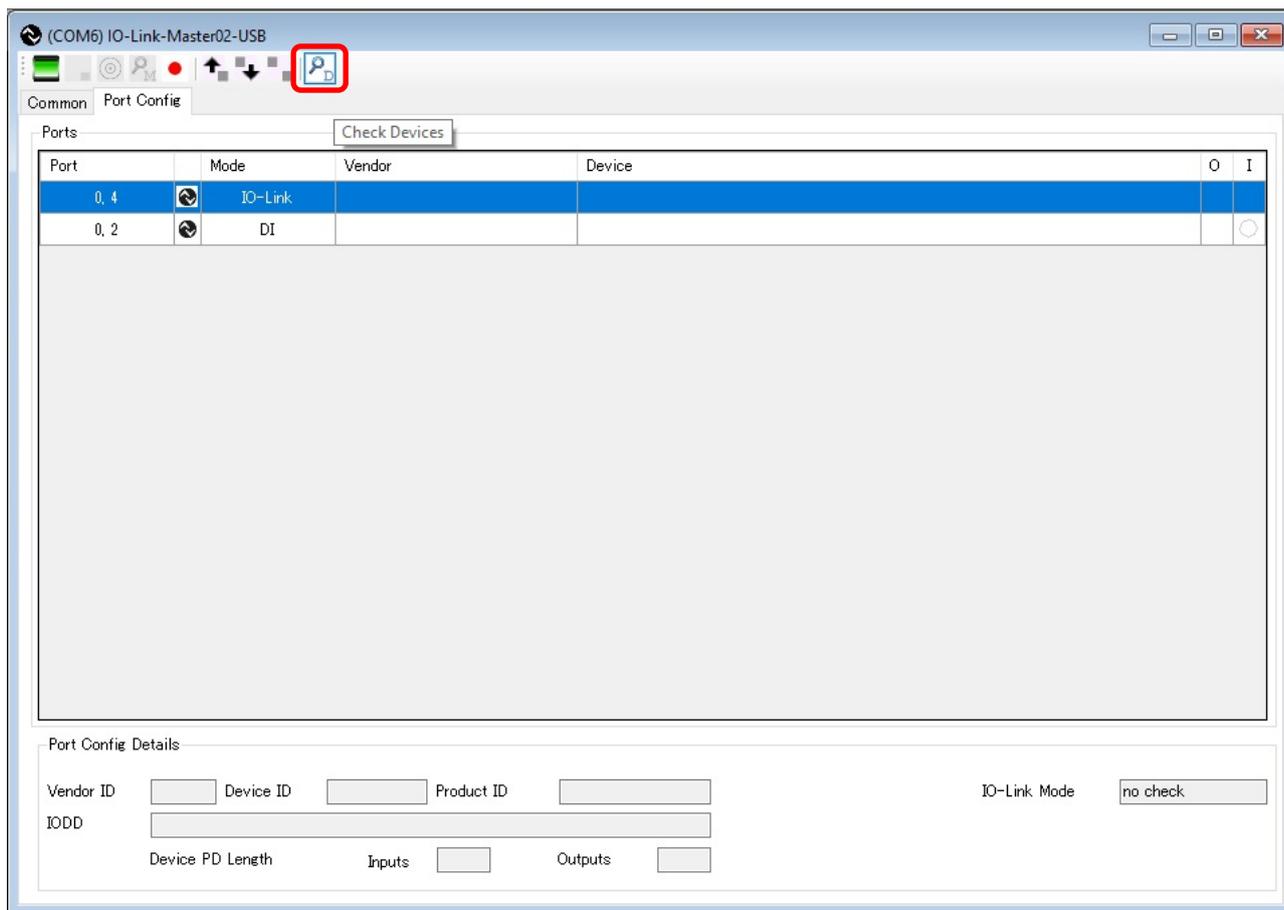


図 7-10 IO-Link Device Tool V5.1 – PE (IO-Link デバイスの確認)

検出が完了すると、「Check Devices」ウィンドウが表示され、IO-Link Master のポートに接続された RA2E1 Starterkit が表示されます。

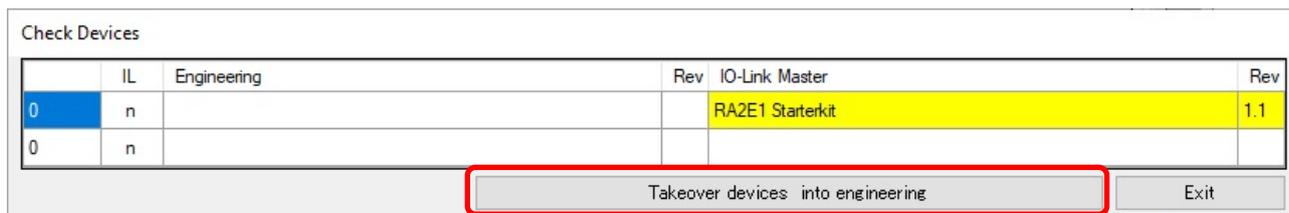


図 7-11 IO-Link Device Tool V5.1 – PE (IO-Link デバイスの検出)

5. 「Takeover devices to engineering」 ボタンをクリックします。

6. マスターとデバイス間の接続が成功すると、図 7-12 のように IO-Link Master に接続されているデバイスの一覧に"RA2E1 Starterkit"が追加されます。

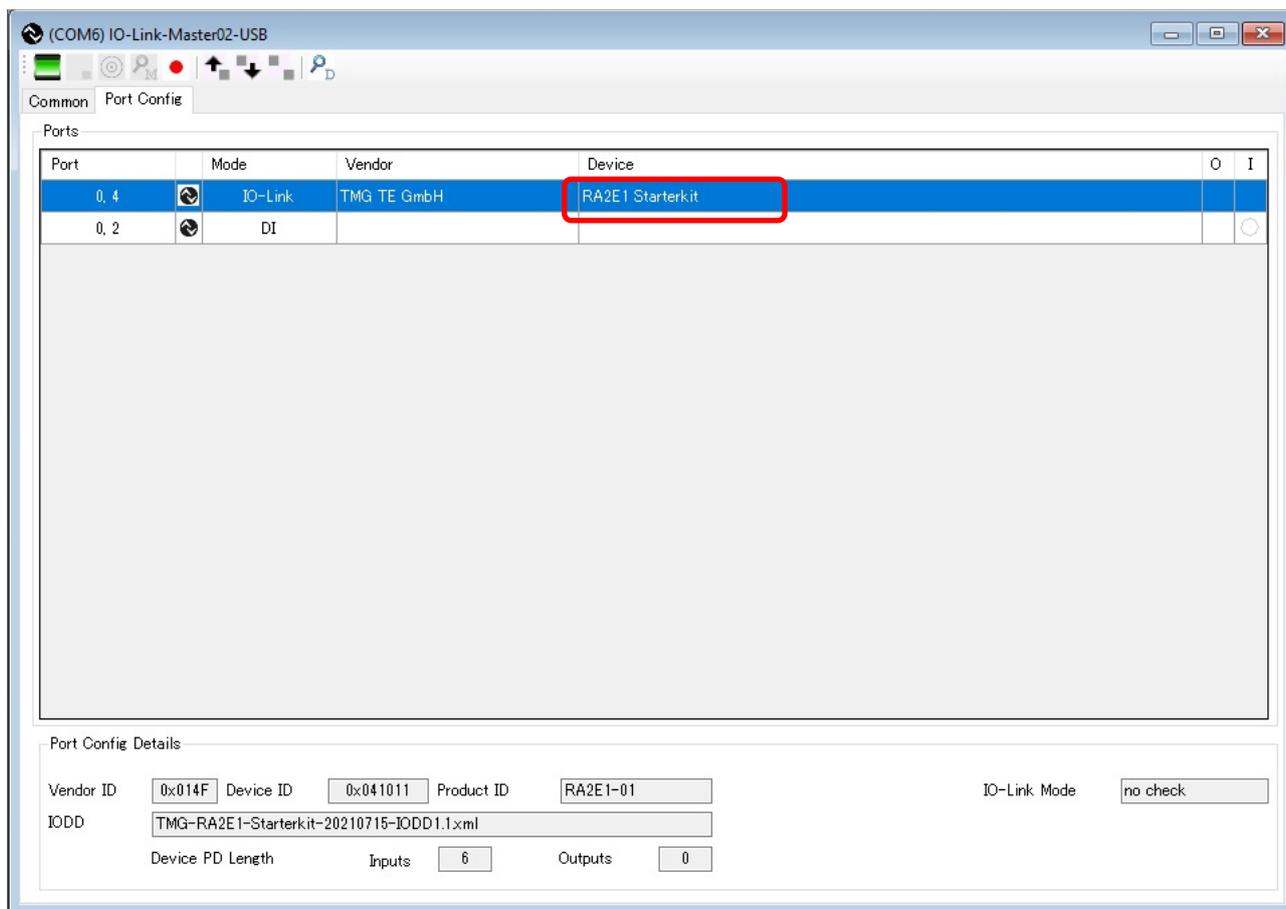


図 7-12 IO-Link Device Tool V5.1 – PE (IO-Link マスターとデバイスの接続成功)

7. 「RA2E1 Starterkit」をダブルクリックすると「7.6.1 Common タブ」が表示されます。

7.6 IO-Link デバイスツールの EK-RA2E1 ボードセンサーデモ

IO-Link デバイスツールから、EK-RA2E1 ボードを使用した IO-Link デバイスを操作する方法を説明します。デバイス操作ツールバーは図 7-13 sensor's description page の左上にあります。

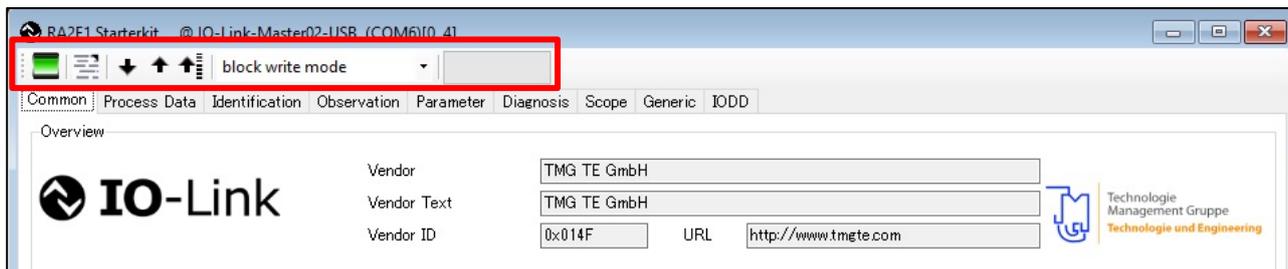


図 7-13 sensor's description page

図 7-14 に、デバイス操作ツールバーのボタンとその機能を示します。



図 7-14 IO-Link Device Tool V5.1 – PE (デバイス操作ツールバー)

【注】 マスターの通信状態が不可の場合「システム稼働状況」のアイコンがグレーで表示されることがあります。その際はマスターボードの電源を再投入し、デバイスツールを再起動すると改善します。

7.6.1 Common タブ

Common タブには以下のようにデバイスに関する一般的な説明情報が表示されます。

ベンダー名 : TMG TE GmbH

ベンダーテキスト : TMG TE GmbH

ベンダーID : 0x014F

URL : www.tmgte.com

デバイス名 : RA2E1 Starterkit

デバイスの説明 : IO-Link Starterkit with RA2E1 Processor

デバイス ID : 0x041011

IO-Link リビジョン : 1.1

SIO モードサポート : No

通信ボーレート : COM2

センサーの最小サイクル時間 : 6400[us]

IO-Link デバイスの写真

接続の説明

M12 コネクタのピン配列

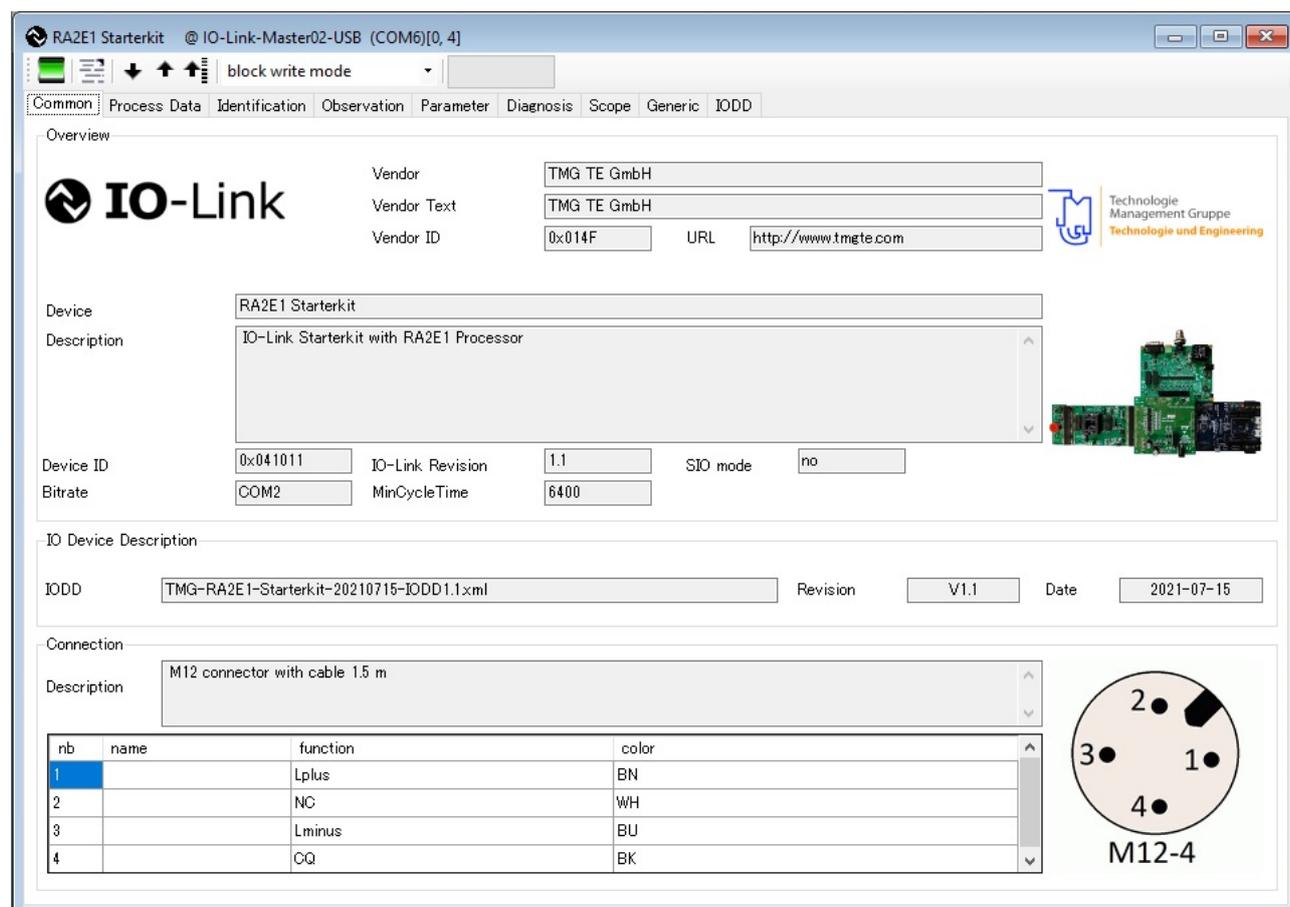


図 7-15 IO-Link Device Tool V5.1 – PE (Common タブ)

7.6.2 Process Data タブ

Process Data タブには、PD Input の情報が表示されます。

1. PD Input

- MDC – Measurement Value
センサー計測値
- MDC – Scale
非対応
- SSC.1 - Switching Signal ^注
スイッチング状態 (Low or High)
- SSC.2 - Switching Signal
非対応

The screenshot shows the 'Process Data' tab in the IO-Link Device Tool V5.1 - PE. The window title is 'RA2E1 Starterkit @ IO-Link-Master02-USB (COM6)[0, 4]'. The 'Process Data' tab is selected, and the 'block write mode' is set to 'block write mode'. The table below displays the PD Input parameters:

| Name | Value | Unit |
|--------------------------|-------|-----------------------|
| [-] PD Input | | |
| MDC - Measurement Value | 40658 | |
| MDC - Scale | 0 | |
| SSC.1 - Switching Signal | Low | <input type="radio"/> |
| SSC.2 - Switching Signal | Low | <input type="radio"/> |

図 7-16 IO-Link Device Tool V5.1 – PE (Process Data タブ)

【注】 SSC.1 - Switching Signal は緑色 LED にのみ連動します。赤色 LED は非対応です。

7.6.3 Identification タブ

Identification タブでは、ユーザーはデバイスに保存されている以下のような識別情報を読み取って確認することができます。

1. デバイス情報
2. アプリケーション固有の情報
3. リビジョン情報

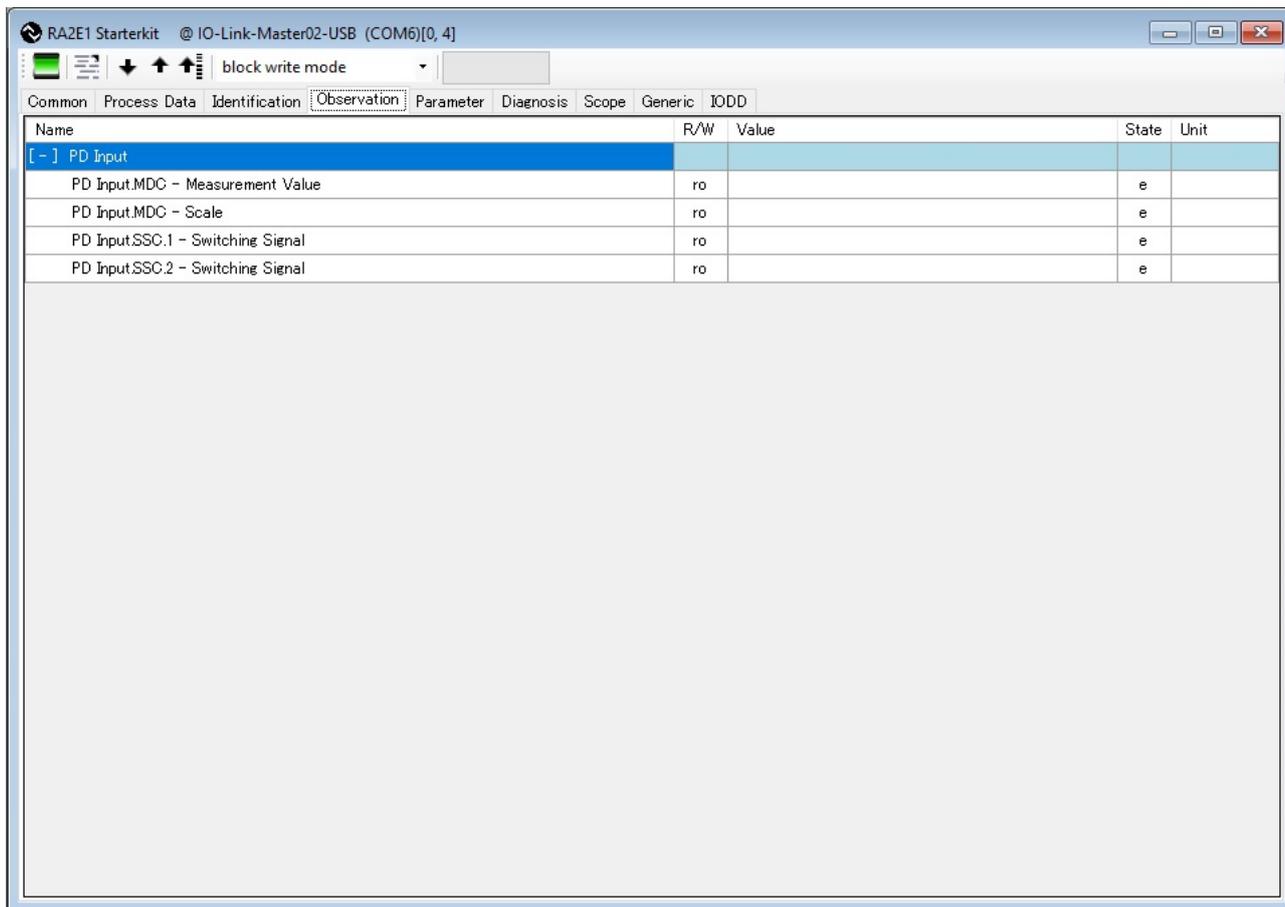
The screenshot shows the 'Identification' tab of the IO-Link Device Tool V5.1 - PE. The window title is 'RA2E1 Starterkit @ IO-Link-Master02-USB (COM6)[0, 4]'. The 'block write mode' is set to 'off'. The 'Identification' tab is selected, showing a table of device information.

| Name | R/W | Value | State | Unit |
|--------------------------|-----|---|-------|------|
| Vendor Name | ro | TMG TE GmbH | i | |
| Vendor Text | ro | www.tmete.com | i | |
| Product Name | ro | RA2E1 Starterkit | i | |
| Product Text | ro | IO-Link Starterkit with RA2E1 Processor | i | |
| Product ID | ro | RA2E1-01 | i | |
| Serial Number | ro | | e | |
| Hardware Revision | ro | | e | |
| Firmware Revision | ro | | e | |
| Application-specific Tag | rw | *** | i | |
| Function Tag | rw | *** | i | |
| Location Tag | rw | *** | i | |

図 7-17 IO-Link Device Tool V5.1 – PE (Identification タブ)

7.6.4 Observation タブ

Observation タブには、デバイスが測定した測定値が表示されます。ユーザーは、デバイスによって測定された情報を確認することができます。



| Name | R/W | Value | State | Unit |
|-----------------------------------|-----|-------|-------|------|
| [-] PD Input | | | | |
| PD Input.MDC - Measurement Value | ro | | e | |
| PD Input.MDC - Scale | ro | | e | |
| PD Input.SSC.1 - Switching Signal | ro | | e | |
| PD Input.SSC.2 - Switching Signal | ro | | e | |

図 7-18 IO-Link Device Tool V5.1 – PE (Observation タブ)

7.6.5 Parameter タブ

Parameter タブには、デバイスのパラメータの設定状態が表示され、ユーザーはここからデバイスの状態を確認することができます。また、デバイスに新しい設定を書き込むこともできます。パラメータの意味、機能については表 5-3 を参照してください。デバイスのパラメータ設定の方法については「7.6.7 デバイスのパラメータ変更 (Teach-In/Read)」を参照してください。

| Name | R/W | Value | State | Unit |
|--------------------------------|-----|---|-------|------|
| [-] General Settings | | | | |
| System Command | wo | Application Reset | | |
| [-] Device Parameterisation | | | | |
| User Variable UInt8_1 | rw | 0 | i | |
| User Variable UInt8_2 | rw | 0 | i | |
| User Variable UInt8_3 | rw | User Variable UInt8_3 Enumeration Value 0 | i | |
| User Variable UInt8_4 | rw | User Variable UInt8_4 Enumeration Value 0 | i | |
| User Variable UInt16_1 | rw | 0 | i | |
| User Variable UInt16_2 | rw | 0 | i | |
| User Variable UInt16_3 | rw | 0 | i | |
| User Variable UInt16_4 | rw | 0 | i | |
| User Variable UInt32_1 | rw | 0 | i | |
| User Variable UInt32_2 | rw | 0 | i | |
| User Variable UInt32_3 | rw | 0 | i | |
| User Variable UInt32_4 | rw | 0 | i | |
| [-] Switching Signal Channel 1 | | | | |
| SSC.1 Param.SP1 | rw | 55000 | i | |
| SSC.1 Param.SP2 | rw | 45000 | i | |
| SSC.1 Config.Logic | rw | High active | i | |
| SSC.1 Config.Mode | rw | Deactivated | i | |
| SSC.1 Config.Hyst | rw | 0 | i | |
| [-] Switching Signal Channel 2 | | | | |
| SSC.2 Param.SP1 | rw | 0 | i | |
| SSC.2 Param.SP2 | rw | 0 | i | |
| SSC.2 Config.Logic | rw | High active | i | |
| SSC.2 Config.Mode | rw | Deactivated | i | |
| SSC.2 Config.Hyst | rw | 0 | i | |
| [-] Teach | | | | |
| Teach Select | rw | Default Channel | i | |
| [-] Teach - Single Value | | | | |
| System Command | wo | Teach SP1 | | |
| System Command | wo | Teach SP2 | | |
| Teach Result.State | ro | Idle | i | |

図 7-19 IO-Link Device Tool V5.1 – PE (Parameter タブ)

7.6.6 Scope タブ

Scope タブでは、プロセスデータを視覚化することが可能です。

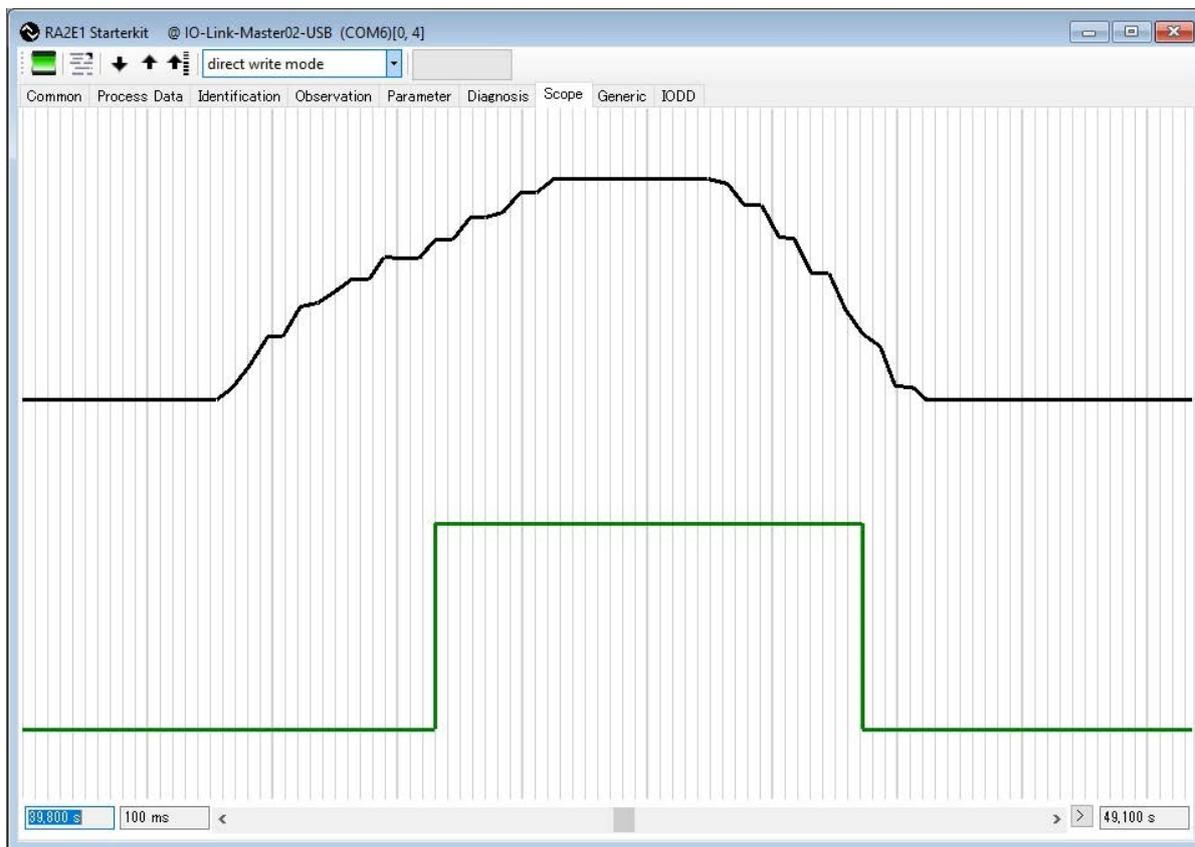


図 7-20 IO-Link Device Tool V5.1 – PE (Scope タブ)

ユーザーは Scope エリアで右クリックすることで、Scope 構成の設定を開くことができます。設定を編集することもできます。

Scope Configuration

| Visible | IO | Name | Digital | Value Min | Value Max | Display Min | Display Max | Color |
|-------------------------------------|----|--------------------------|-------------------------------------|-----------|-----------|-------------|-------------|-------|
| <input checked="" type="checkbox"/> | I | MDC - Measurement Value | <input type="checkbox"/> | 0 | 65535 | 25 | 90 | black |
| <input type="checkbox"/> | I | MDC - Scale | <input type="checkbox"/> | -100 | 100 | 25 | 50 | red |
| <input checked="" type="checkbox"/> | I | SSC.1 - Switching Signal | <input checked="" type="checkbox"/> | 0 | 1 | 10 | 40 | green |
| <input type="checkbox"/> | I | SSC.2 - Switching Signal | <input checked="" type="checkbox"/> | 0 | 1 | 75 | 100 | blue |

Time Resolution: 100 ms Grid Width: 10 Pixel [cancel] [take over] [ok]

図 7-21 IO-Link Device Tool V5.1 – PE (Scope タブ、Scale/Parameter 設定)

7.6.7 デバイスのパラメータ変更 (Teach-In/Read)

ユーザーが Parameter タブを開くと、デバイス固有のパラメータが Value 列に表示されます。これらの値は IODD ファイルに記録されています。Parameter タブの詳細については「7.6.5 Parameter タブ」を参照してください。

SetPoint パラメータは「Teach Values」パラメータを使用して設定するか、「Standard Command」を使用して設定することができます。

ユーザーは IO-Link デバイスツールにより、パラメータ設定を変更できます。設定は書き込みモードによって異なります。以下にその方法を示します。

7.6.7.1 block write 書き込みモードを使用したパラメータ設定

この block write mode では、パラメータの変更を行ってから「Download to device」ボタンをクリックすることで、デバイスにパラメータを設定します。

1. 書き込みモードを「block write mode」に変更します。
2. SetPoint を設定するには、パラメータ「SSC.1 Param.SP1」または「SSC.1 Param.SP2」の「Value」フィールドをクリックします。
3. 数値を入力し、キーボードの Enter キーを押下します。「Status」の黄色の背景色は、パラメータがまだデバイスに設定されていないことを示します。
4. パラメータを書き込むには「Download to device」ボタンをクリックします。
5. 「Status」の緑色の背景色は、パラメータがデバイスに設定され、マスターとデバイス間の同期が正常に実行されたことを示します。

The screenshot shows the 'Parameter' tab of the IO-Link Device Tool. The 'block write mode' dropdown is selected. The table below lists various parameters with their R/W status, current values, and states.

| Name | R/W | Value | State | Unit |
|--------------------------------|-----|---|-------|------|
| [-] General Settings | | | | |
| System Command | wo | Application Reset | | |
| [-] Device Parameterisation | | | | |
| User Variable UInt8_1 | rw | 0 | d | |
| User Variable UInt8_2 | rw | 0 | d | |
| User Variable UInt8_3 | rw | User Variable UInt8_3 Enumeration Value 0 | d | |
| User Variable UInt8_4 | rw | User Variable UInt8_4 Enumeration Value 0 | d | |
| User Variable UInt16_1 | rw | 0 | d | |
| User Variable UInt16_2 | rw | 0 | d | |
| User Variable UInt16_3 | rw | 0 | d | |
| User Variable UInt16_4 | rw | 0 | d | |
| User Variable UInt32_1 | rw | 0 | d | |
| User Variable UInt32_2 | rw | 0 | d | |
| User Variable UInt32_3 | rw | 0 | d | |
| User Variable UInt32_4 | rw | 0 | d | |
| [-] Switching Signal Channel 1 | | | | |
| SSC.1 Param.SP1 | rw | 50000 | c | |
| SSC.1 Param.SP2 | rw | 45000 | d | |
| SSC.1 Config.Logic | rw | High active | d | |
| SSC.1 Config.Mode | rw | Deactivated | d | |
| SSC.1 Config.Hyst | rw | 0 | d | |

図 7-22 IO-Link Device Tool V5.1 – PE (block write mode でのパラメータ設定)

7.6.7.2 direct write 書き込みモードを使用したパラメータ設定

この direct write mode では、パラメータの変更が自動的にデバイスに設定されます。

1. 書き込みモードを「direct write mode」に変更します。
2. SetPoint を設定するには、パラメータ「SSC.1 Param.SP1」または「SSC.1 Param.SP2」の「Value」フィールドをクリックします。
3. 数値を入力し、キーボードの Enter キーを押下します。
4. 「Status」が黄色に変わり、その後緑色に変わることで、パラメータがデバイスに設定され、マスターとデバイス間の同期が行われたことを示します。

| Name | R/W | Value | State | Unit |
|--------------------------------|-----|---|-------|------|
| [-] General Settings | | | | |
| System Command | wo | Application Reset | | |
| [-] Device Parameterisation | | | | |
| User Variable UInt8_1 | rw | 0 | d | |
| User Variable UInt8_2 | rw | 0 | d | |
| User Variable UInt8_3 | rw | User Variable UInt8_3 Enumeration Value 0 | d | |
| User Variable UInt8_4 | rw | User Variable UInt8_4 Enumeration Value 0 | d | |
| User Variable UInt16_1 | rw | 0 | d | |
| User Variable UInt16_2 | rw | 0 | d | |
| User Variable UInt16_3 | rw | 0 | d | |
| User Variable UInt16_4 | rw | 0 | d | |
| User Variable UInt32_1 | rw | 0 | d | |
| User Variable UInt32_2 | rw | 0 | d | |
| User Variable UInt32_3 | rw | 0 | d | |
| User Variable UInt32_4 | rw | 0 | d | |
| [-] Switching Signal Channel 1 | | | | |
| SSC.1 Param.SP1 | rw | 50000 | d | |
| SSC.1 Param.SP2 | rw | 40000 | c | |
| SSC.1 Config.Logic | rw | High active | d | |
| SSC.1 Config.Mode | rw | Deactivated | d | |
| SSC.1 Config.Hyst | rw | 0 | d | |

図 7-23 IO-Link Device Tool V5.1 – PE (direct write mode でのパラメータ設定)

7.6.7.3 動作モード変更

動作モードは Deactivated が初期値として設定されています。SetPoint を使用したスイッチング状態機能を有効にする場合には、動作モードを Window または Two point に変更してください。

動作モードについては「5.3 動作モードとスイッチング状態」を参照して下さい。

1. 書き込みモードを「block write mode」に変更します。
2. SSC.1 Config.Mode のリストで Window または Two point を選択します。
3. 「Download to device」 ボタンをクリックしてデバイスに設定します。

| | | | |
|--------------------------------|----|-------------|---|
| [-] Switching Signal Channel 1 | | | |
| SSC.1 Param.SP1 | rw | 55000 | d |
| SSC.1 Param.SP2 | rw | 45000 | d |
| SSC.1 Config.Logic | rw | High active | d |
| SSC.1 Config.Mode | rw | Deactivated | d |
| SSC.1 Config.Hyst | rw | Deactivated | d |
| [-] Switching Signal Channel 2 | | | |
| SSC.2 Param.SP1 | rw | 0 | d |
| SSC.2 Param.SP2 | rw | 0 | d |

図 7-24 IO-Link Device Tool V5.1 – PE（動作モード変更）

【注】 Single point は本サンプルアプリケーションで非対応となっています。選択しないでください。

7.6.7.4 デバイス計測値による SetPoint 設定

デバイスが計測した値を SetPoint として SP1, SP2 に設定します。

- Parameter タブの Value フィールドにある「Teach SP1」ボタンまたは「Teach SP2」ボタンをクリックします。
その時点の計測値が「SSC.1 Param.SP1」または「SSC.1 Param.SP2」に自動的に設定され、Teach Result State の結果も「SP1 Success」または「SP2 Success」となります。

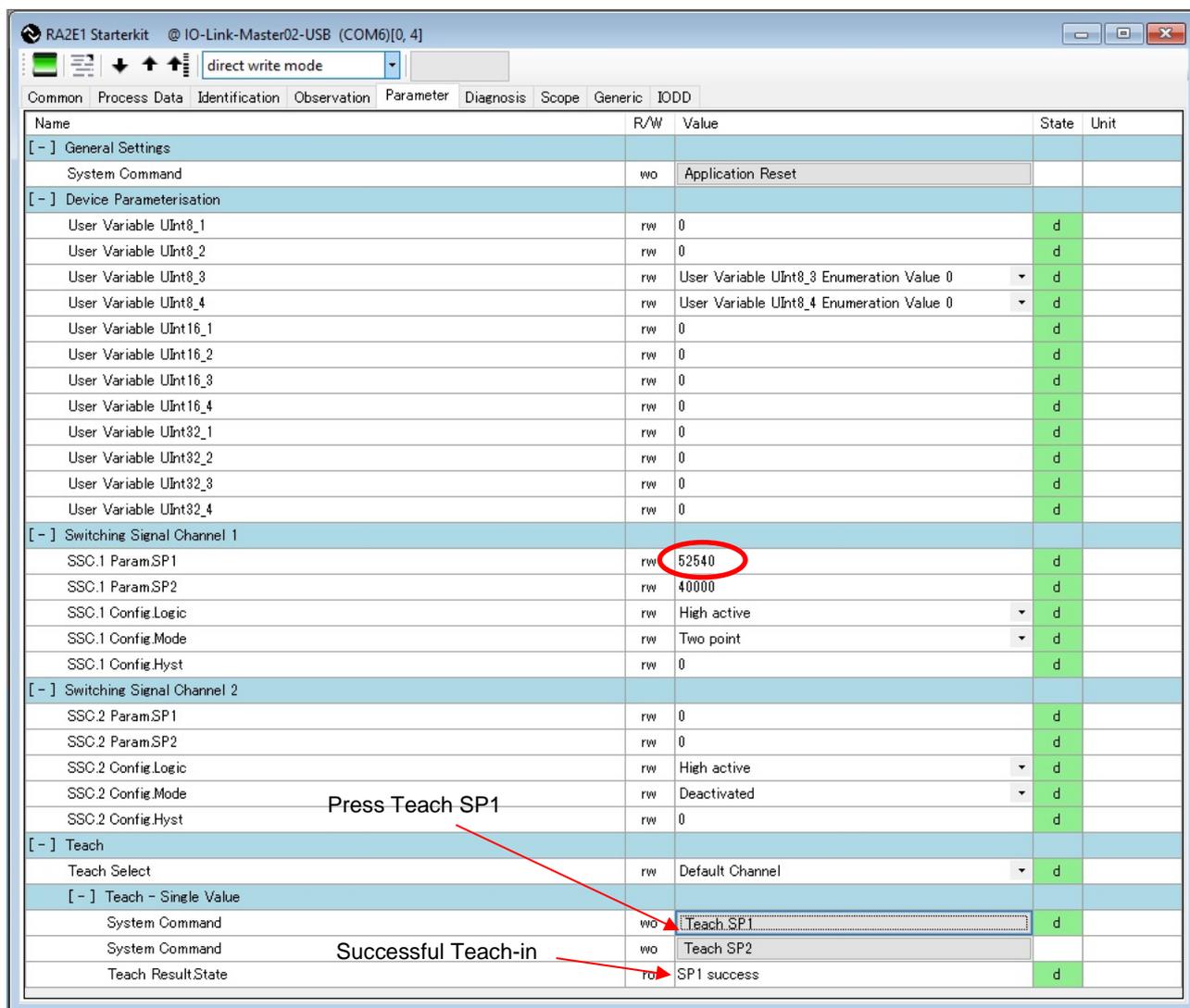


図 7-25 IO-Link Device Tool V5.1 – PE (デバイス計測値による SetPoint 設定)

【注】 Teach Select は本サンプルアプリケーションで非対応となっています。変更しないでください。

7.6.7.5 パラメータの読み取り

ユーザーは「Upload from device」ボタンをクリックして、デバイスに書き込まれている現在のパラメータを読み取ることができます。「Upload from device」ボタンについては「7.6 IO-Link デバイスツールの EK-RA2E1 ボードセンサーデモ」を参照してください。

7.6.7.6 パラメータをデフォルト値にリセット

デバイスの設定をデフォルト値に戻します。

1. 「Application Reset」ボタンをクリックします。

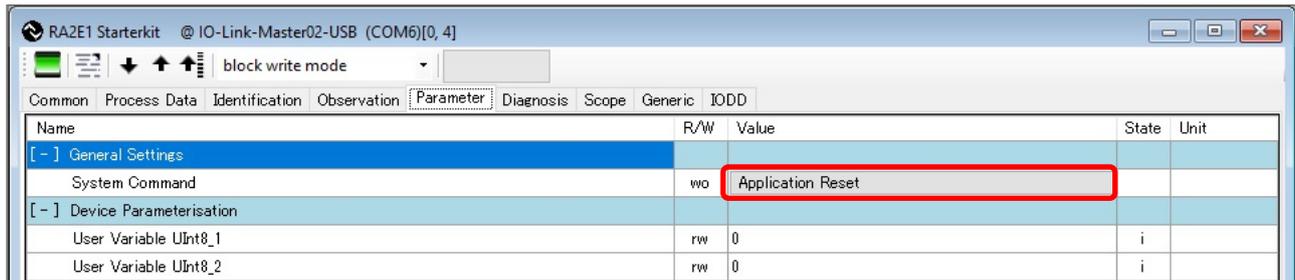


図 7-26 IO-Link Device Tool V5.1 – PE (Application Reset)

【注】 Application Reset を実施すると、動作モードはデフォルトの Deactivated に設定され、LED2/LED3 は常に消灯状態になります。動作モードの変更をする場合は「7.6.7.3 動作モード」を参照して操作してください。

7.6.8 IODD タブ

このタブには IODD 情報が表示されます。

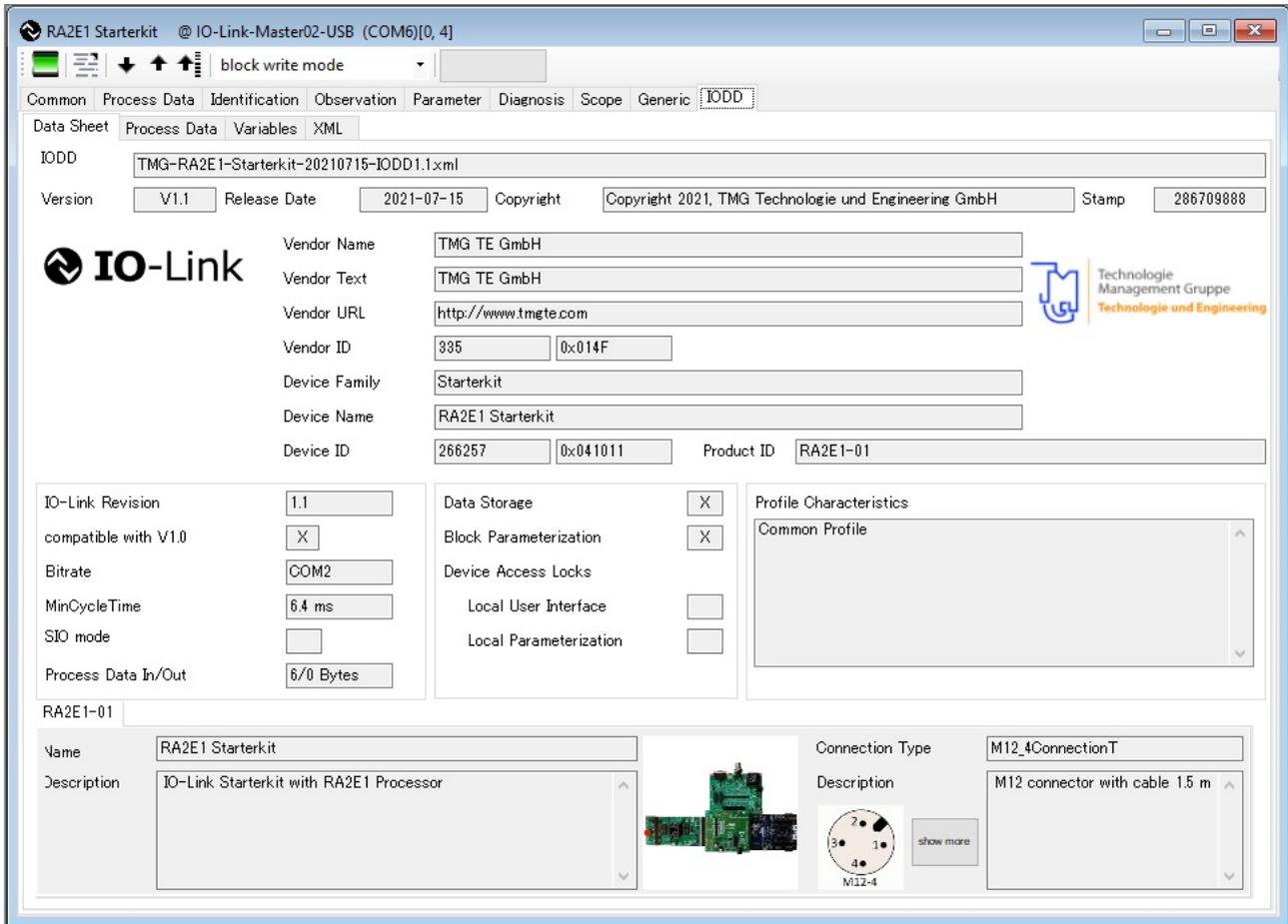


図 7-27 IO-Link Device Tool V5.1 – PE (IODD タブ)

8. 使用例

本章では、サンプルアプリケーションの使用例を解説します。

操作中の測定データ (Measurement Value) やスイッチング状態は IO-Link 通信を介して、IO-Link デバイスから IO-Link マスター、PC へと伝達されます。IO-Link Device Tool の「7.6.2 Process Data タブ」や「7.6.6 Scope タブ」を開き、以降の操作における値の変化をご確認ください。

8.1 IO-Link パラメータをデフォルト値にリセット

操作開始前に EK-RA2E1 の IO-Link パラメータのリセットを実施しておきます。

1. 「7.6.7.6 パラメータをデフォルト値にリセット」に従って、リセットを行います。

8.2 操作方法

本章では、以下に示す設定値での動作例を示します。

| | |
|----------|--------|
| 動作モード | Window |
| SP1 | 55,000 |
| SP2 | 45,000 |
| ZSSC 分解能 | 16bit |

- 「7.6.7.3 動作モード変更」に従って、動作モードを Window に設定します。
- SRB のダイヤルを反時計回りで最下限に設定します。

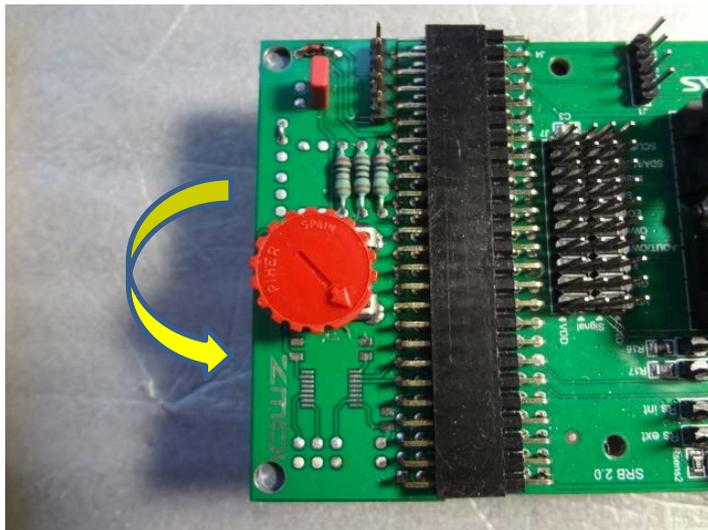


図 8-1 SRB 下限設定

測定値が SP2 未満である為 LED2 は消灯状態です。

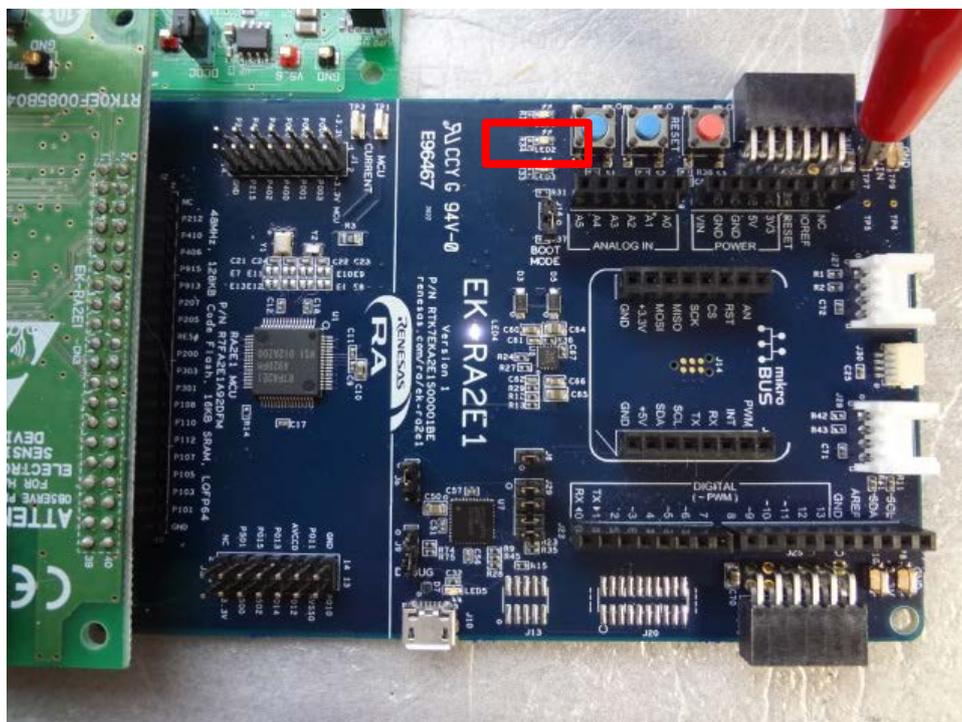


図 8-2 LED 状態 - SP2 未満

3. SRB のダイヤルを時計回りに回す。

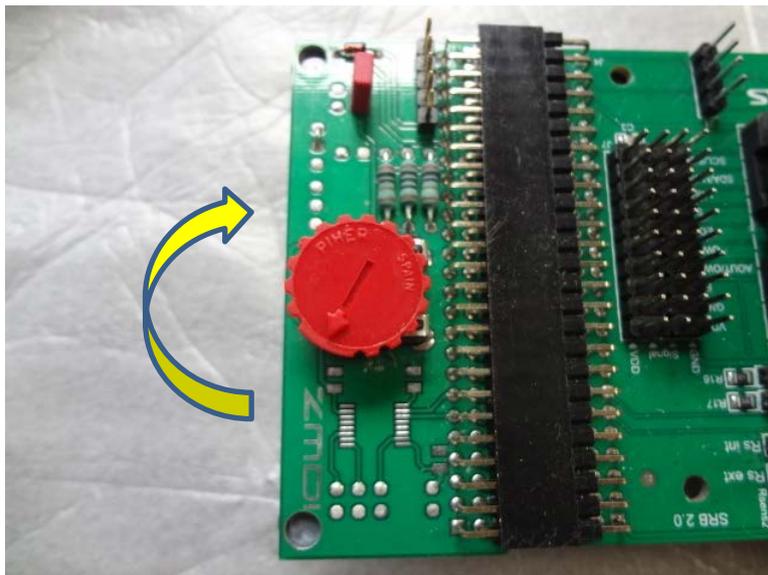


図 8-3 SRB - SP2 位置

測定値が SP2 以上になると LED2 が点灯します。この時のダイヤルの位置を図 8-3 に示します。

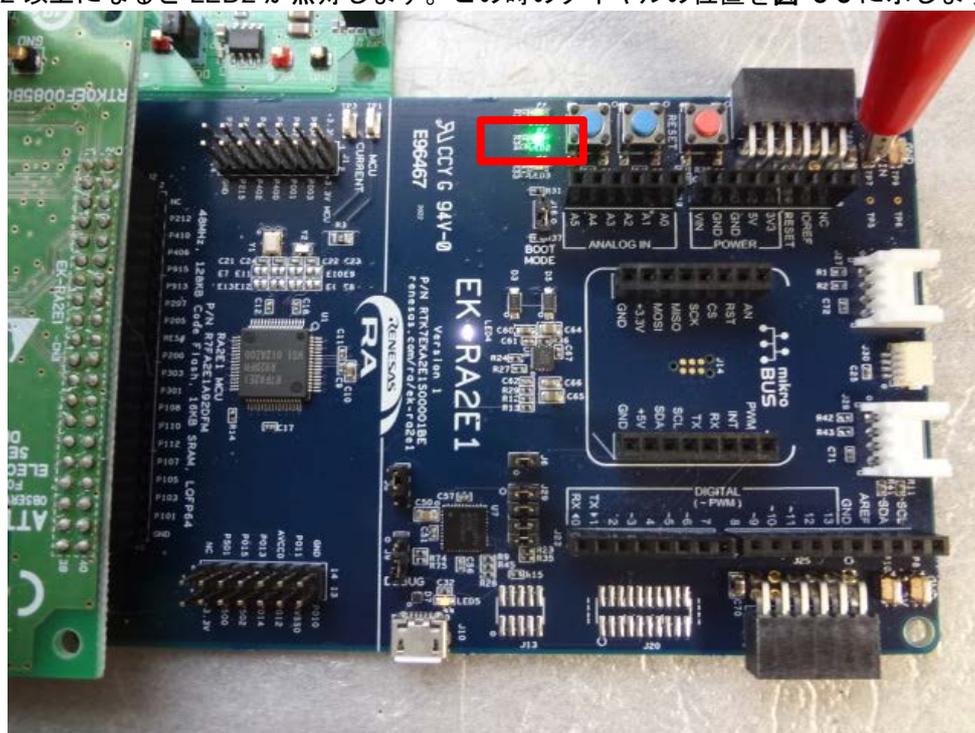


図 8-4 LED 状態 - SP2 以上

4. SRB のダイヤルをさらに時計回りに回す。

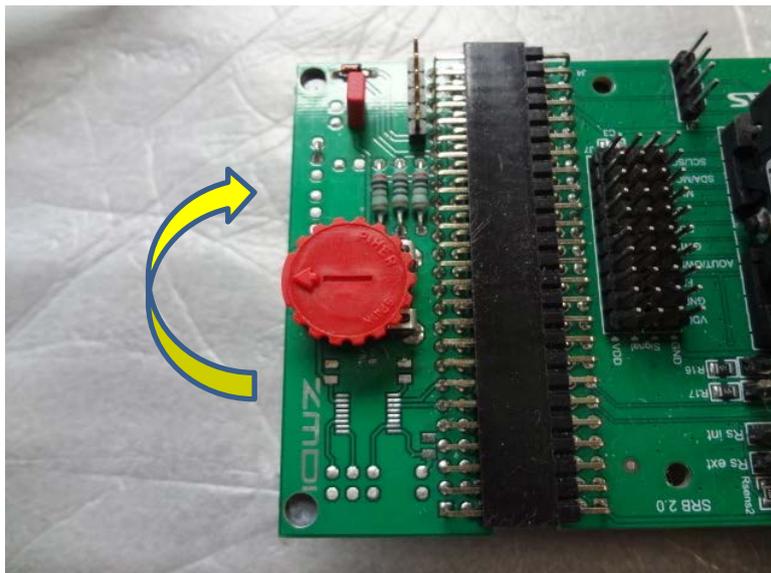


図 8-5 SRB - SP1 到達

測定値が SP1 を超過すると LED3(赤)に切り替わります。この時のダイヤルの位置を図 8-5 に示します。

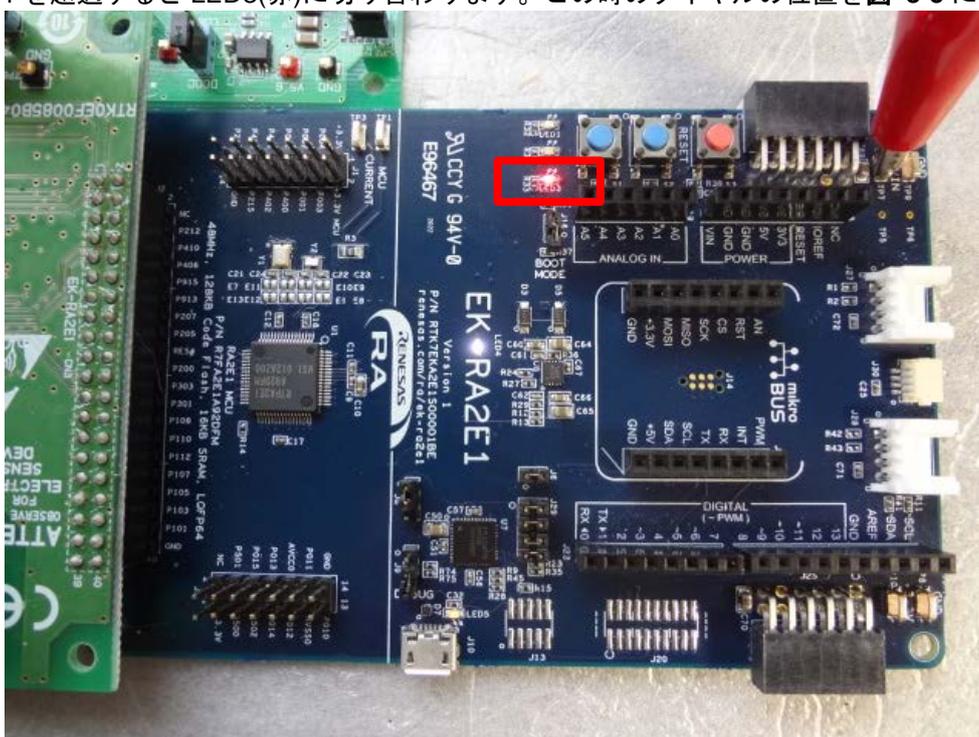


図 8-6 LED 状態 - SP1 超過

【注】 ZSSC での計測は ZSSC の Measurement Commands の 1 つである AAHEX コマンドで計測を行います。AAHEX コマンドについては以下の資料を参照してください。

- [ZSSC3240 Evaluation Kit User Manual](#) (Table 9. Measurement Commands)
- [ZSSC3240 DataSheet](#) (Table 33. Command List)

9. ZSSC の設定変更方法

ZSSC の設定は Windows PC と EK-RA2E1 ボードを USB-SERIAL 変換ケーブルで接続した状態で、PC 上の ZSSC GUI から行います。

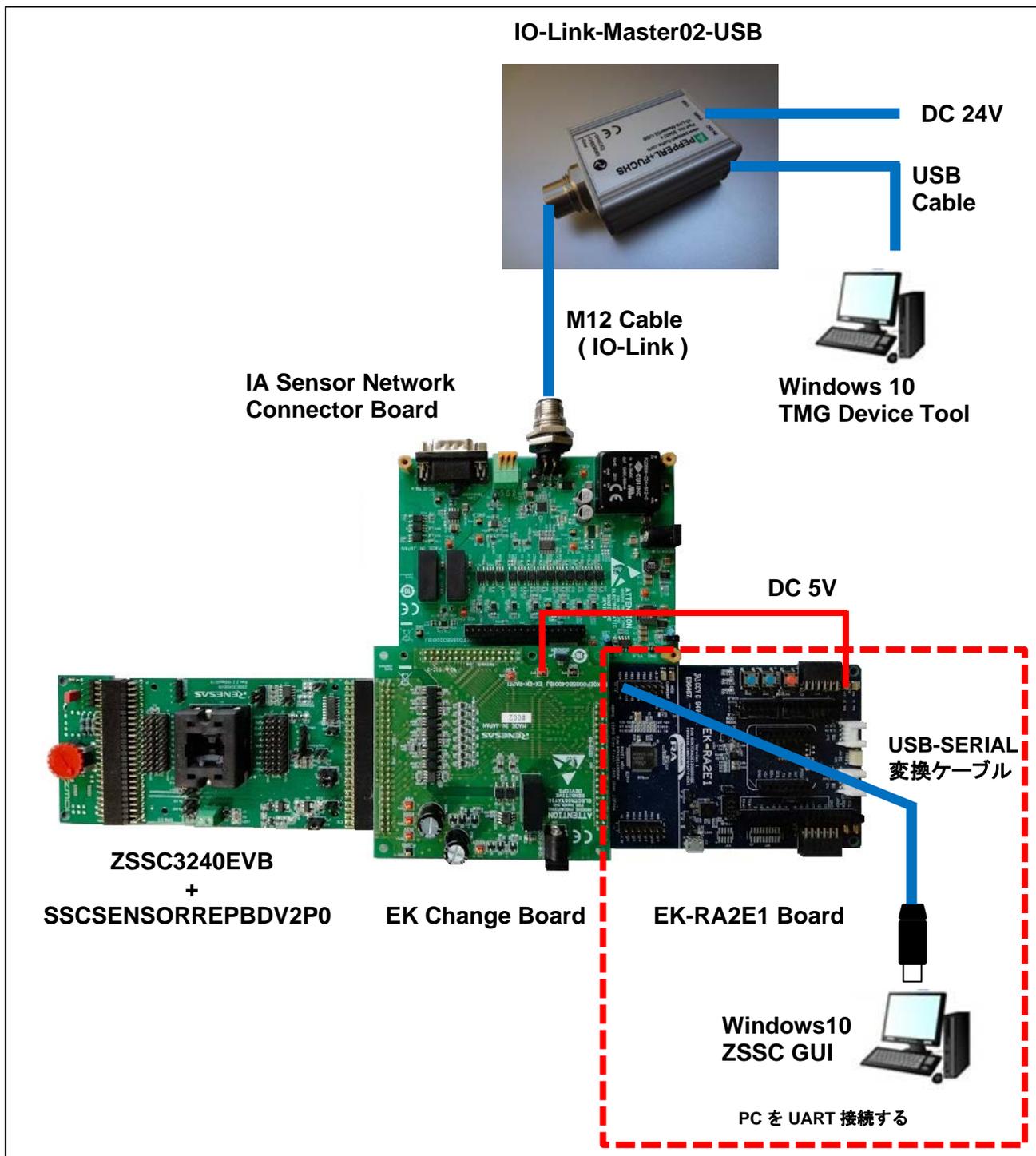


図 9-1 USB-SERIAL 変換ケーブル接続状態

【注】 本アプリケーションノートでは ZSSC GUI を使用した ZSSC3240 の設定と基本的な計測方法についてのみを解説します。ZSSC GUI 及び ZSSC3240 の詳細については以下の資料を参照してください

- ZSSC3240 Evaluation Kit User Manual
- ZSSC3240 DataSheet

9.1 USB-SERIAL 変換ケーブルの接続

EK ボードには PC と通信用の USB I/F が無いため、J1 コネクタの J1-9(P401)と J1-10(P402)、J1-14(GND)に USB-SERIAL 変換ケーブルを接続して仮想 COM で PC との通信を実現します。

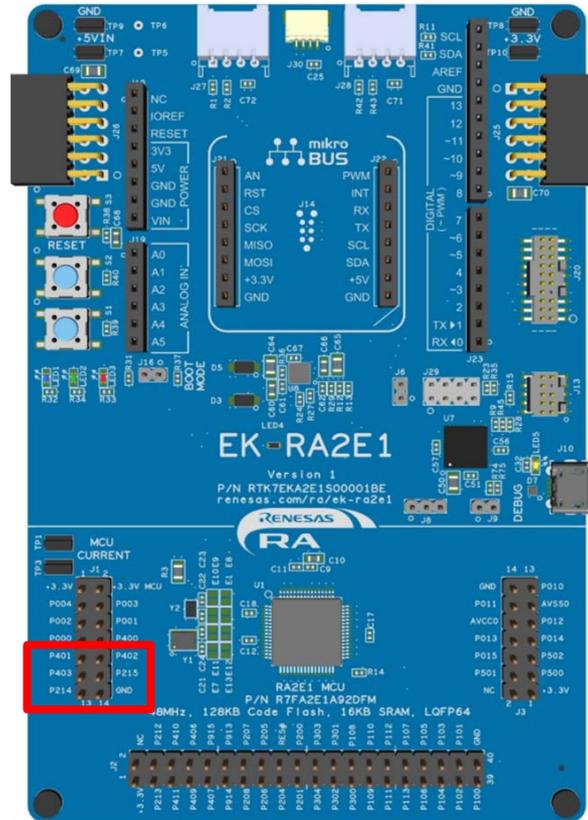


図 9-2 EK-RA2E1 ボード J1 コネクタ

FT232RL 内蔵 USB-SERIAL 変換ケーブルの RXD/TXD/GND 線を以下のように接続します。
 FT232RL と同等の機能内蔵のケーブルであれば使用することができます。

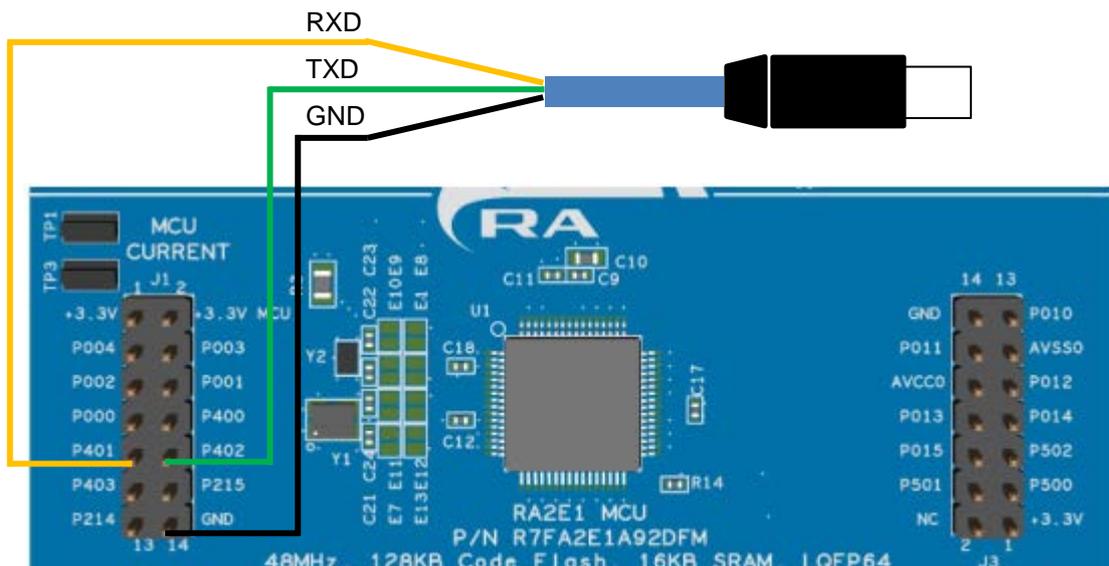


図 9-3 USB-SERIAL 変換ケーブル接続

9.2 仮想 COM ポート番号確認

USB-SERIAL 変換ケーブルを PC に接続すると自動的にドライバが選択され、COM ポートとして認識します^{注1}。

Windows のデバイスマネージャーで USB-SERIAL 変換ケーブルの仮想 COM ポート番号を確認します^{注2}。

以下は COM7 として認識した場合の例です。

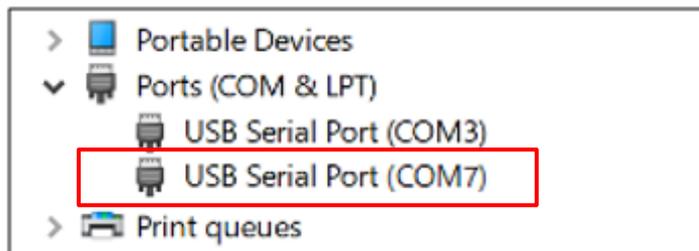


図 9-4 デバイスマネージャー

- 【注】
1. FT232RL 互換品のケーブルを使用した場合はドライバが自動的に設定されないものがあります。その場合はドライバを入手して手動でドライバ設定してください。
 2. 使用する USB-SERIAL 変換ケーブルによって表示される名称が異なる場合があります。

9.3 ZSSC GUI の起動

ZSSC GUI を起動します。以下は起動後の ZSSC GUI の画面です。

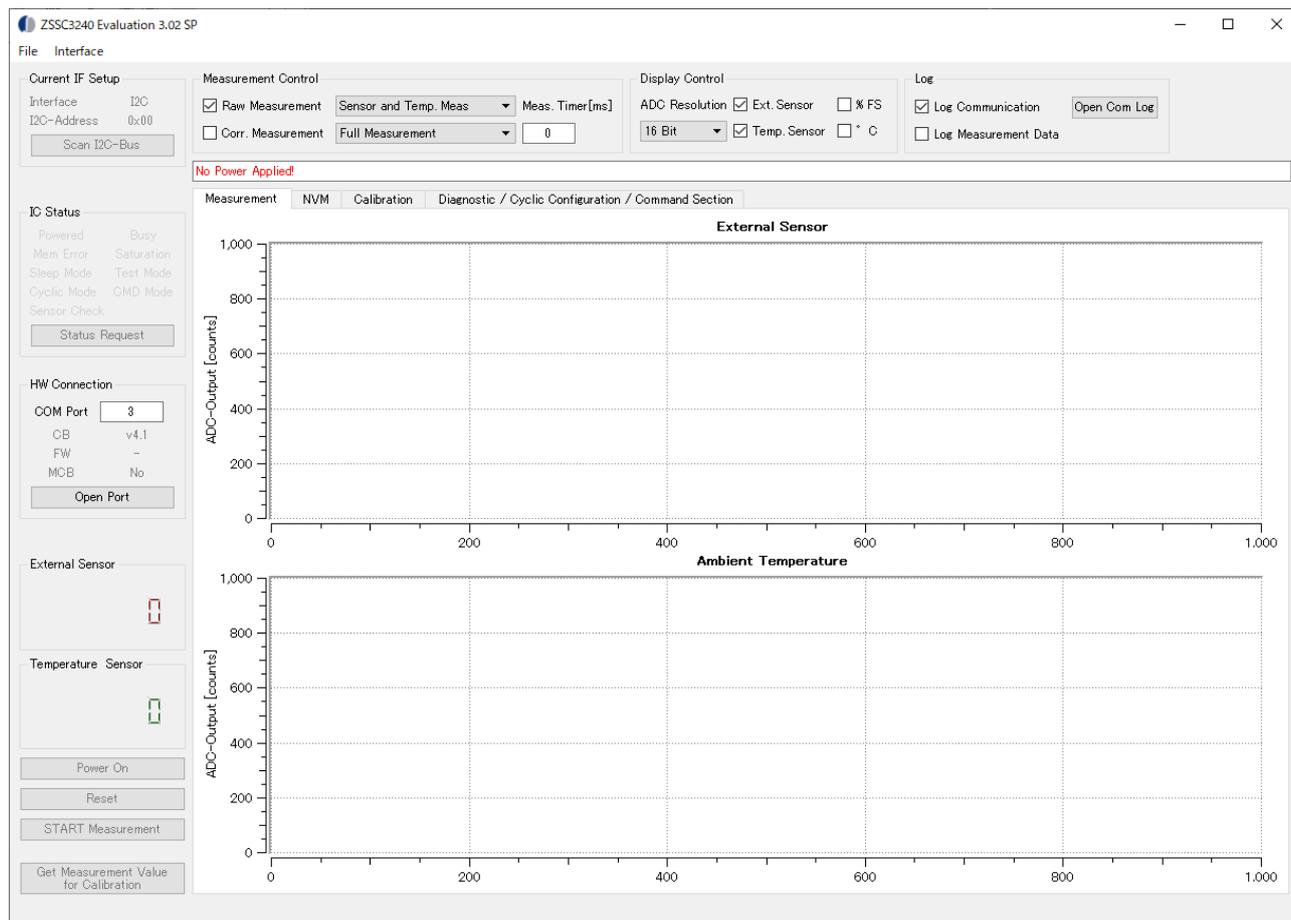


図 9-5 ZSSC GUI 起動後の画面

【注】 GUI は、約 10 秒間の通信ネゴシエーション後に表示されます。

【注】 通信ネゴシエーションは GUI の COM Port デフォルトの COM3 で行われ、通信が確認できなかった場合はエラーメッセージが表示されます。

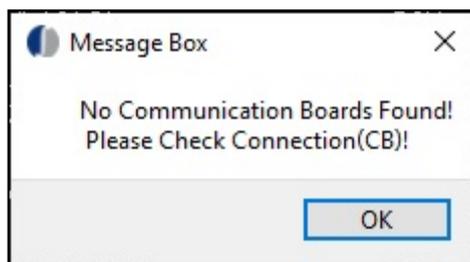


図 9-6 GUI 起動時の通信エラー表示

9.4 通信経路確立

9.4.1 Open Port

COM Port に、図 9-4 デバイスマネージャーで確認した COM ポート番号を設定し[Open Port]ボタンをクリックします。

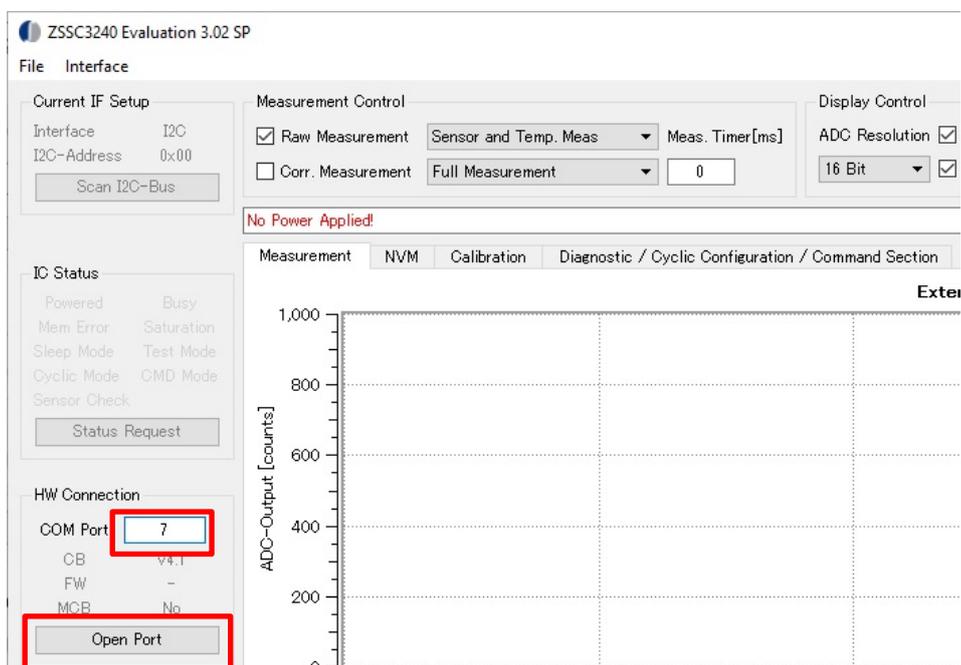


図 9-7 Open Port

【注】 使用する COM ポートが COM3 の場合、自動的に[Open Port]が選択されます。

Open Port に成功した場合、以下図 9-8 のように GUI 画面が変化します。

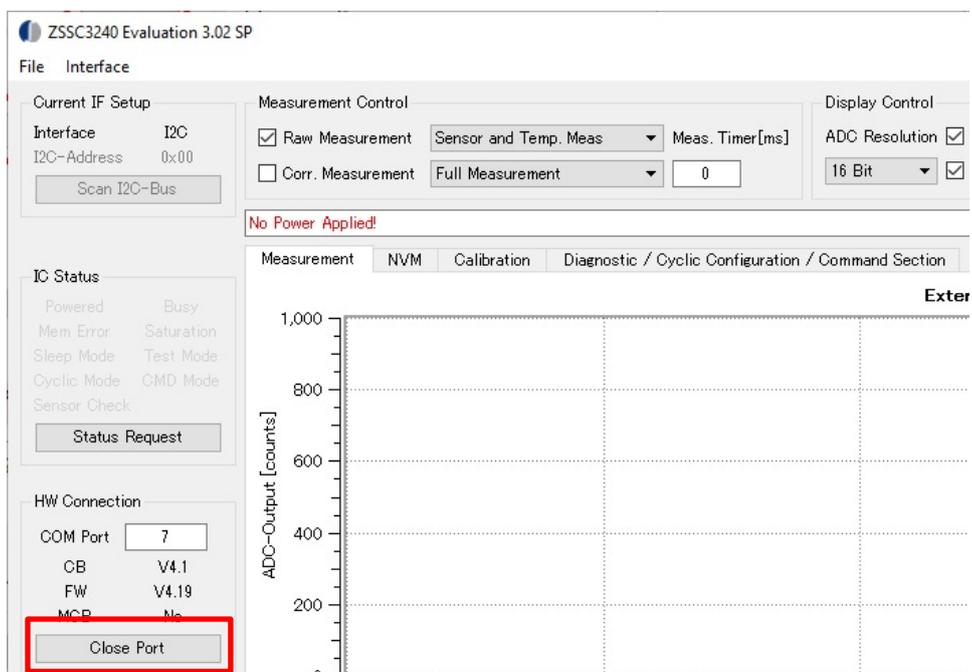


図 9-8 Open Port 成功

【注】 デフォルトの COM3 での通信が成功し、自動的に[Open Port]が選択された場合、IC Status などの表示は図 9-8 に示す例とは異なる場合があります。

9.4.2 Interface

[Interface]-[I2C]を選択します^{注1}。

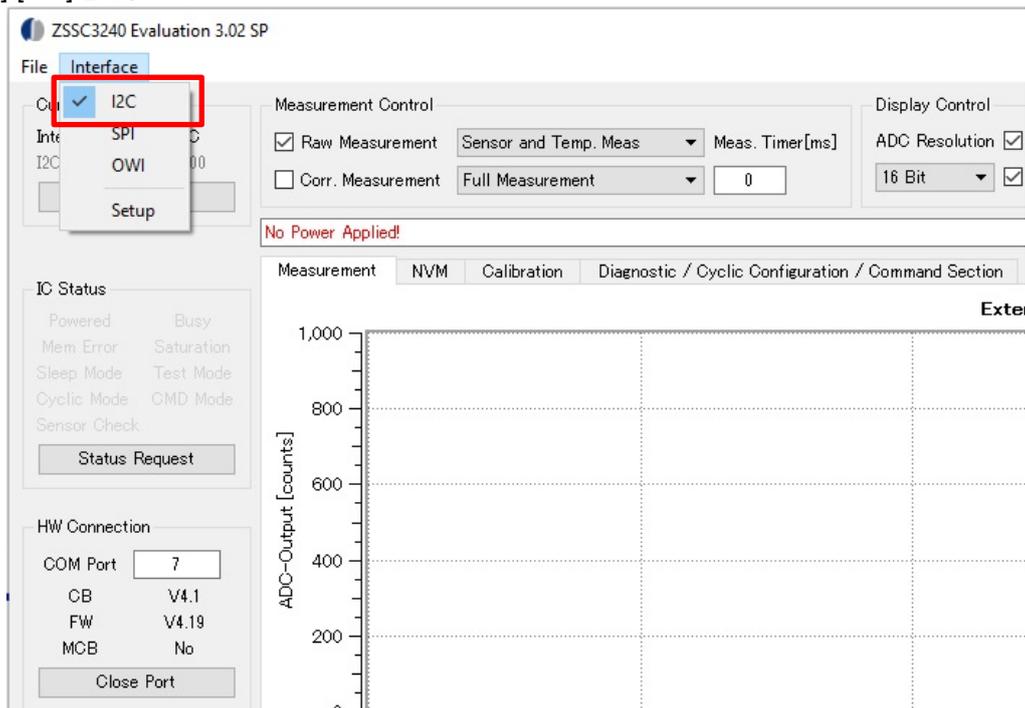


図 9-9 インターフェース選択

Current IF Setup の Interface に"I2C [400kHz]"が表示されていることを確認してください^{注2}。

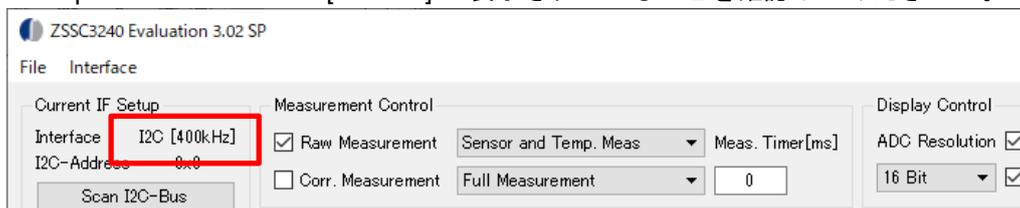


図 9-10 Current IF Setup の Interface

- 【注】
1. 本サンプルアプリケーションでは I2C のみを操作対象とします。SPI と OWI を使用する事はできません。
 2. 図 9-10 に示すように Interface の表示が"I2C [400kHz]"に変更されない場合、再度[Interface]メニューから[I2C]を選択してください。

9.4.3 Scan I2C-Bus

[Scan I2C-Bus] をクリックし ZSSC の I2C デバイスを検出します。

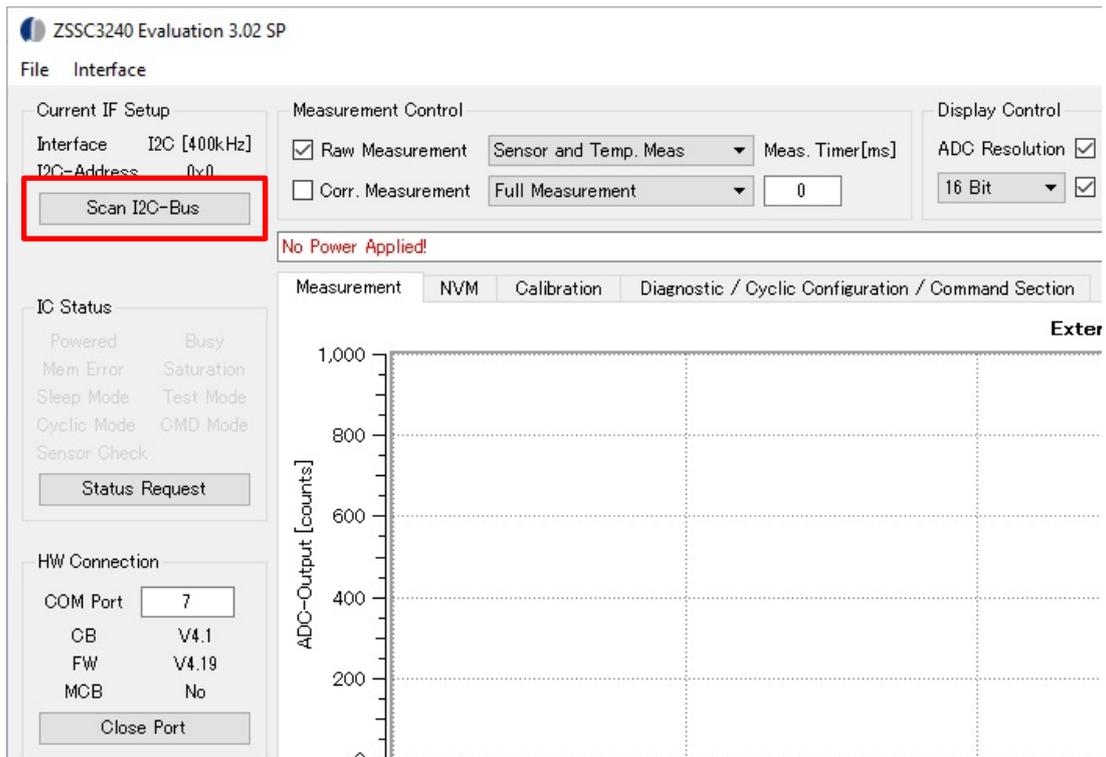


図 9-11 Scan I2C-Bus

I2C デバイスを検出すると図 9-12 に示すメッセージが表示されます。これで ZSSC3240 との通信が可能となりました。



図 9-12 I2C デバイス検出

9.5 測定(Corr.Measurement)

本章では、以下の表に示す設定値をもとに ZSSC GUI ツールを使用して測定を行う例を示します。

| | |
|----------|-----------|
| 動作モード | Two Point |
| SP1 | 55,000 |
| SP2 | 45,000 |
| ZSSC 分解能 | 16bit |

【注】 各グラフ画像は、[Start Measurement]後に説明の為にそれぞれ個別にキャプチャしたものです。連続した計測によるものではありません。

- 「7.6.7.3 動作モード変更」に従って、動作モードを Two Point に設定します。
- [Corr. Measurement]をチェックし、[Full Measurement]を選択します。[Temp.Sensor]がチェックされている場合はチェックを解除します。

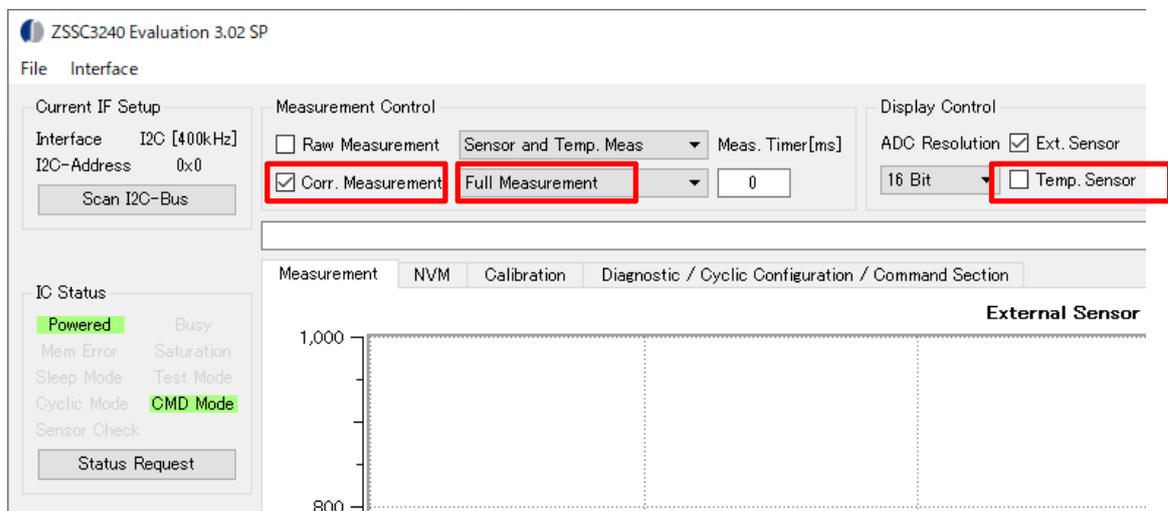


図 9-13 Corr. Measurement

3. SRB のダイヤルを反時計回しで最下限に設定します。

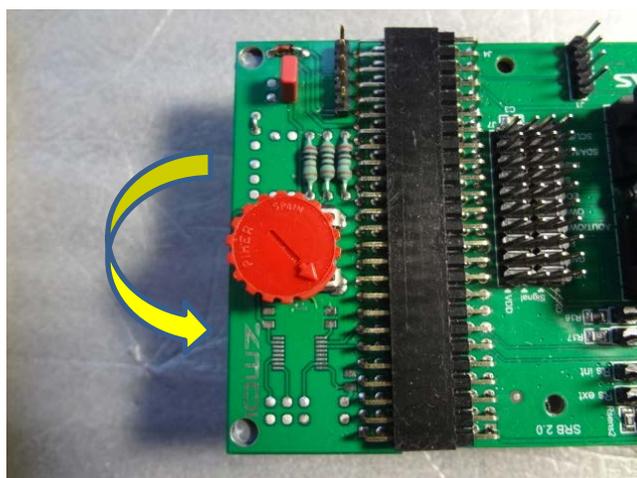


図 9-14 SRB 下限設定

4. 測定開始。

[START Measurement]クリックで ZSSC3240EVb に接続されたセンサーでの測定が開始されグラフ表示が開始されます。クリック操作後、同ボタンは[STOP Measurement]表示に切り替わります。

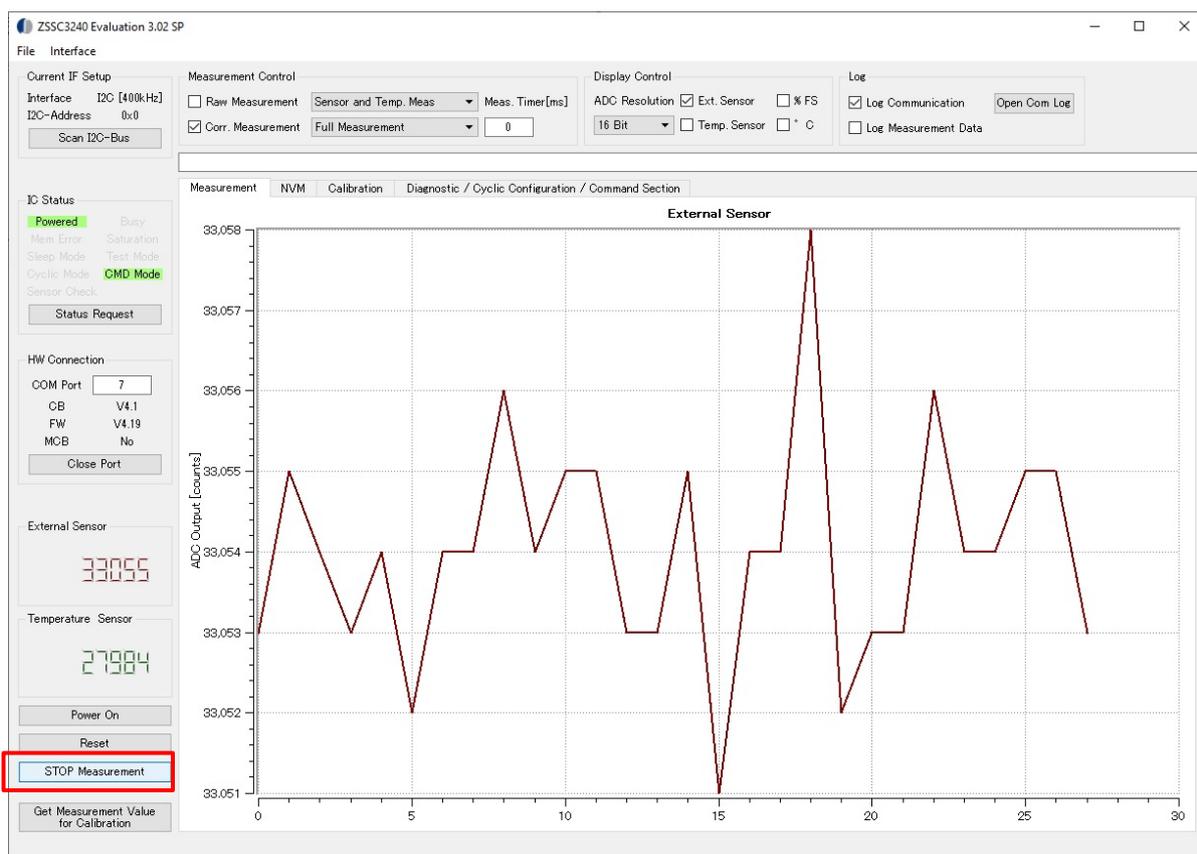


図 9-15 測定開始

【注】 グラフスケールは測定値に従って自動的に調整されます。測定開始直後のダイヤルが未操作の状態では測定値の変化が少ないためグリッド単位が1のスケールで表示されます。

5. SRB のダイヤルを時計回りに、External Sensor 値が SP1(55000)の少し手前になるまで回します。

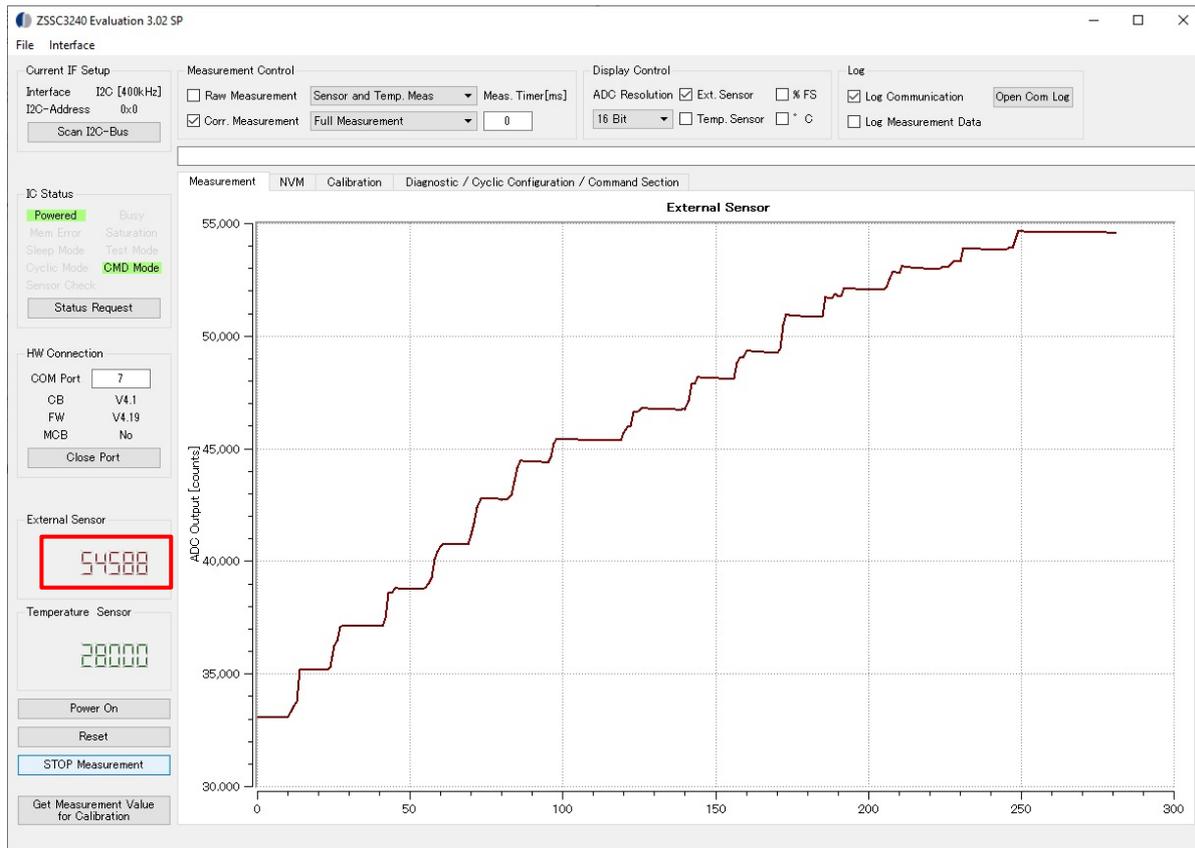


図 9-16 Measurement - SP1 手前

測定値が SP1 以下である為 LED2 は消灯状態です。

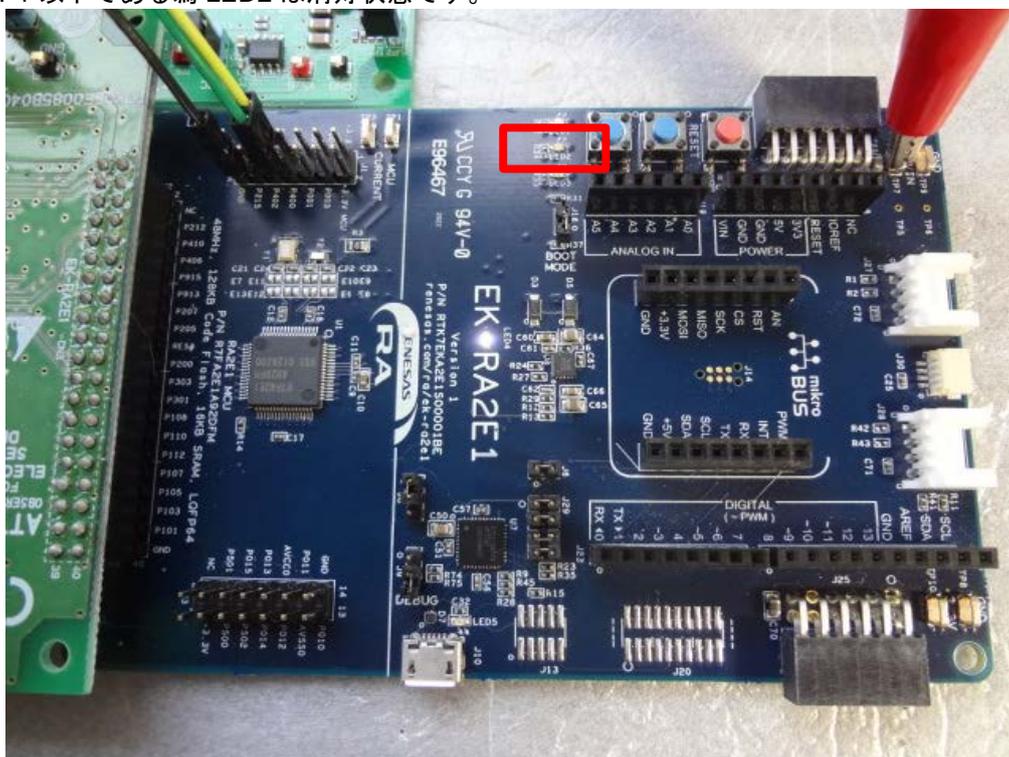


図 9-17 LED 状態 - SP1 以下

6. SRB のダイヤルを時計回りに、External Sensor 値が SP1(55000)超過になるまで回します。

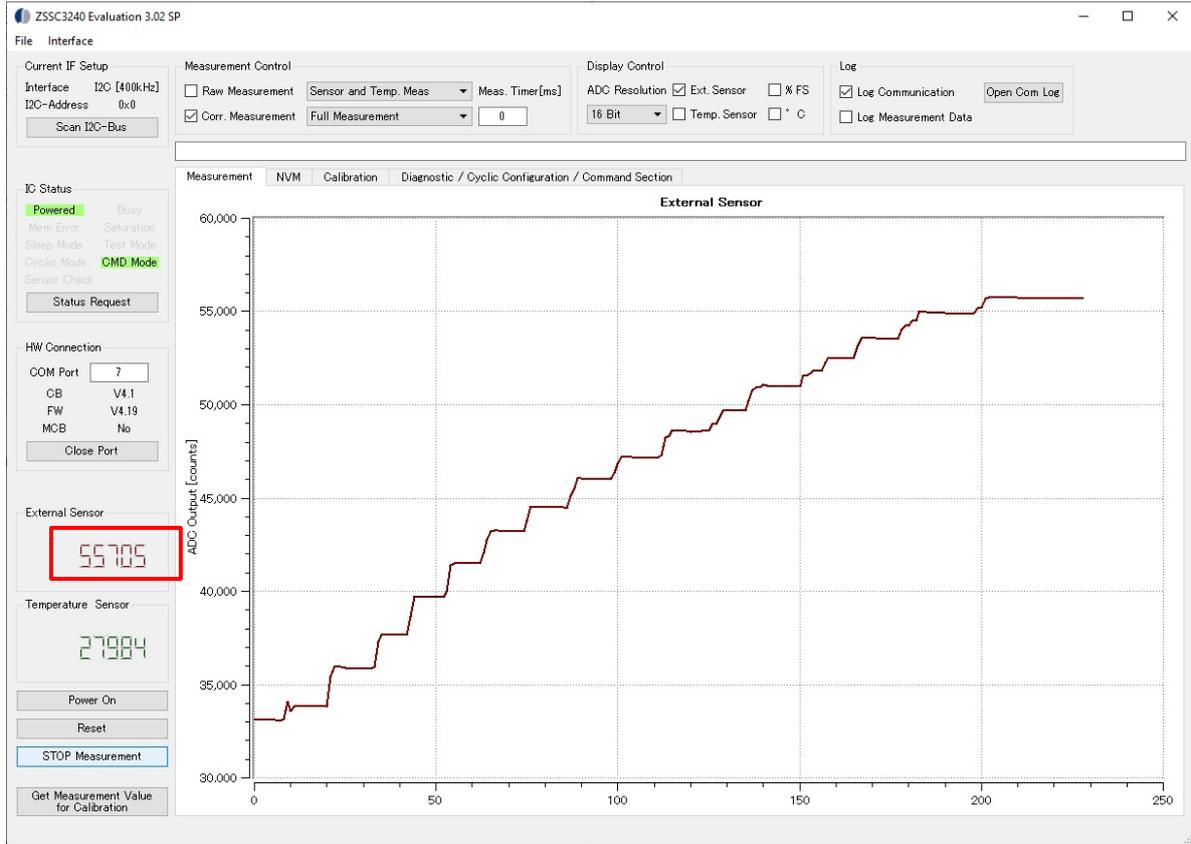


図 9-18 Measurement - SP1 超過

測定値が SP1 超過で LED2 が点灯します。

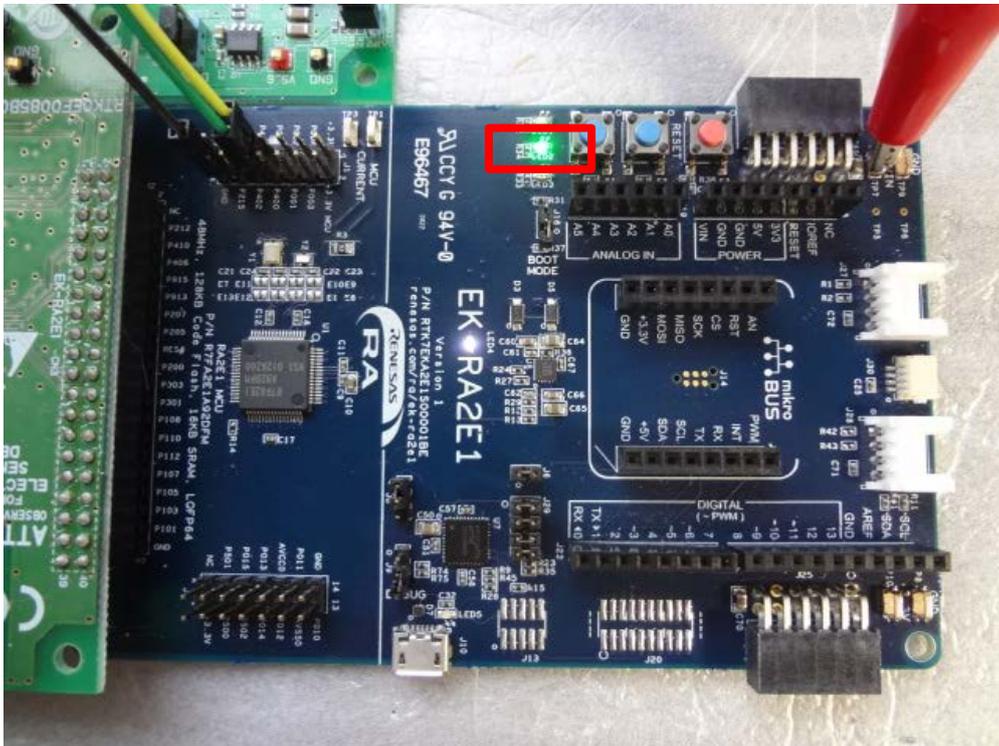


図 9-19 LED 状態 - SP1 超過

7. SRB のダイヤルを反時計回りに、External Sensor 値が SP2(45000)未満になるまで回します。

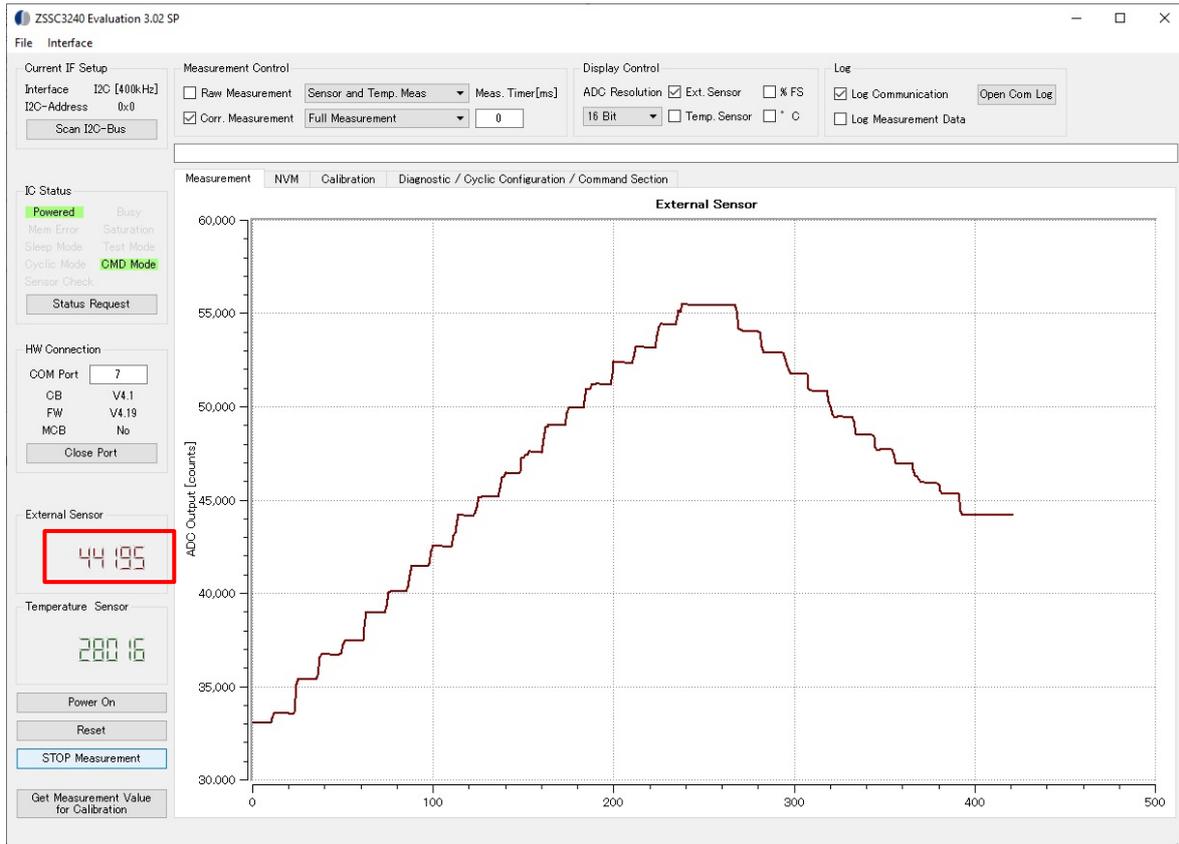


図 9-20 Measurement – SP2 未満

測定値が SP2 未満になった時点で LED2 が消灯します。

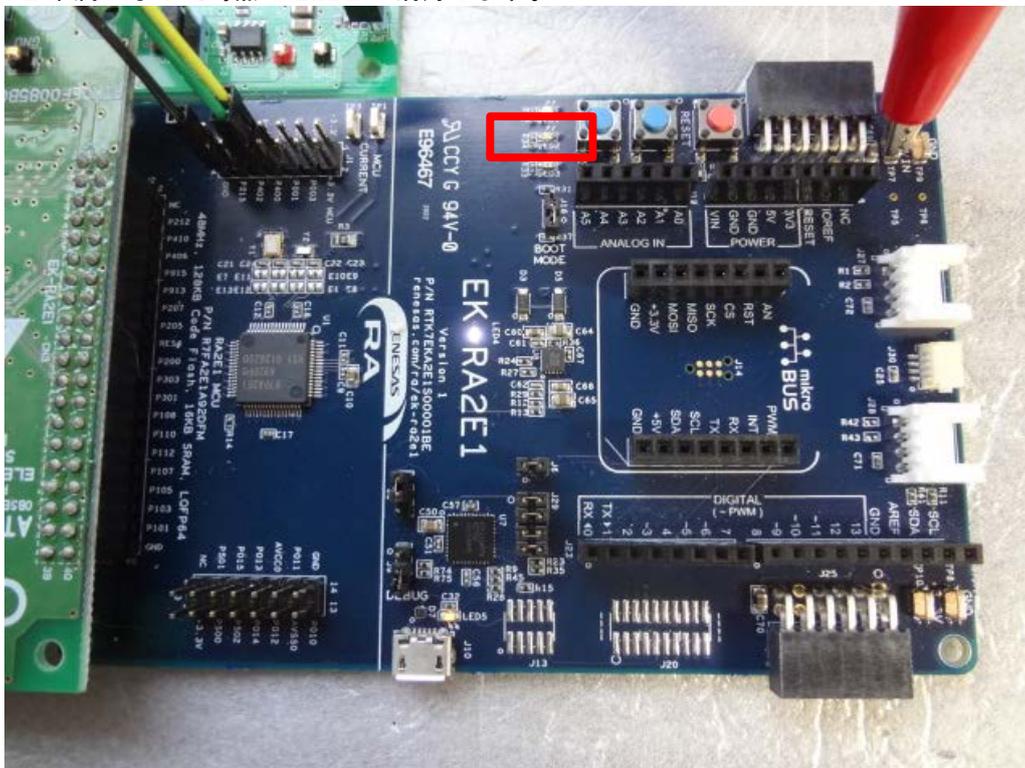


図 9-21 LED 状態 – SP2 未満

9.6 NVM 操作

ZSSC に搭載された NVM(不揮発性メモリ)レジスタの操作方法を解説します。

9.6.1 Read NVM

[NVM]タブを選択します。初期状態では値は読み込まれていません。

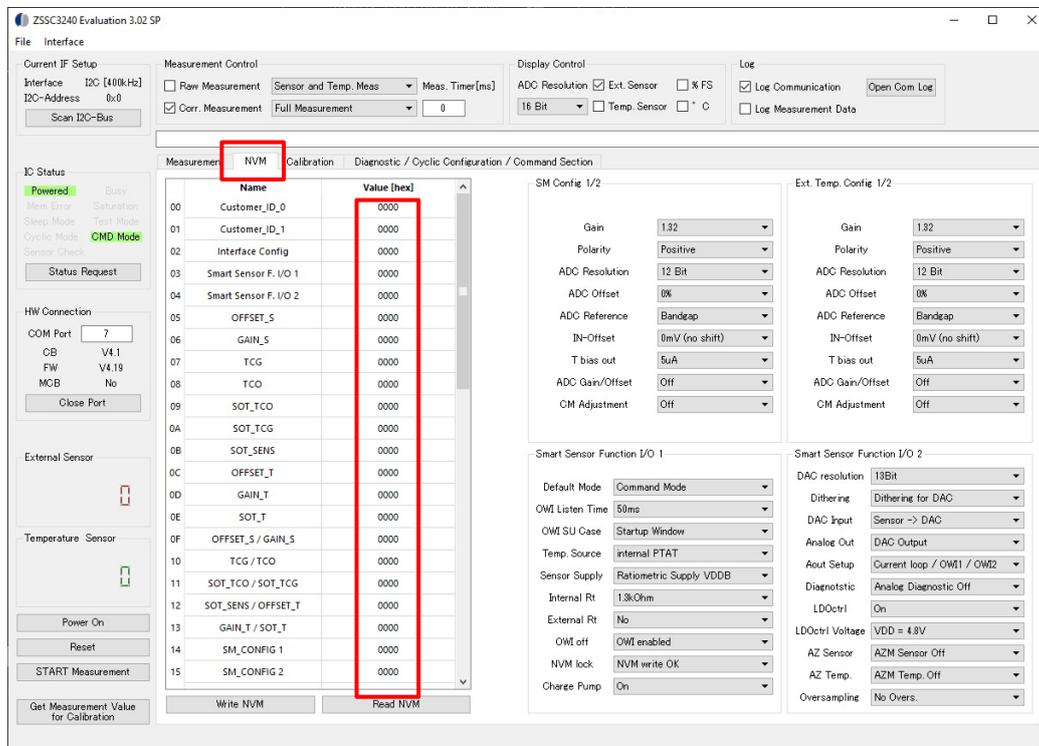


図 9-22 NVM タブ初期状態

[Read NVM]をクリックすると、ZSSC と通信を行い値が読み込まれます。

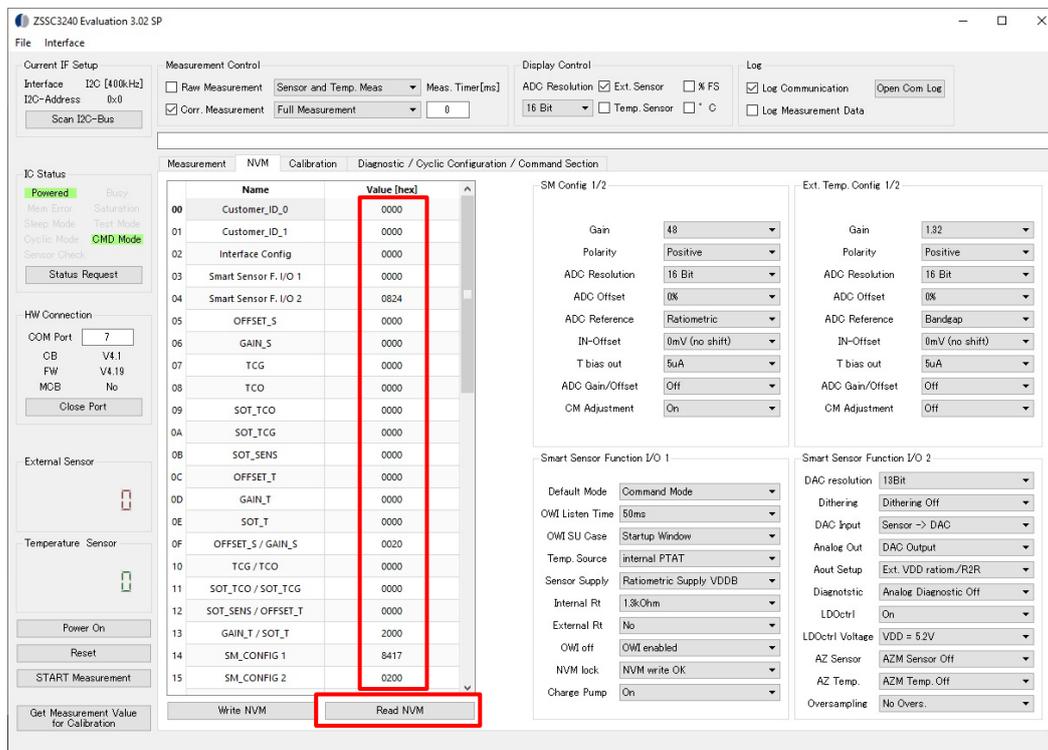


図 9-23 Read NVM 実行後の NVM タブの状態

9.6.2 Write NVM

設定値を変更し、Write NVM を実行することで、ZSSC に設定内容を転送します。以下では ADC 分解能を変更する例を示します。

[SM Config 1/2] の [ADC Resolution] で 24Bit を選択します。変更内容は、**図 9-24** に示すようにレジスタ一覧で赤色背景でハイライトされます。

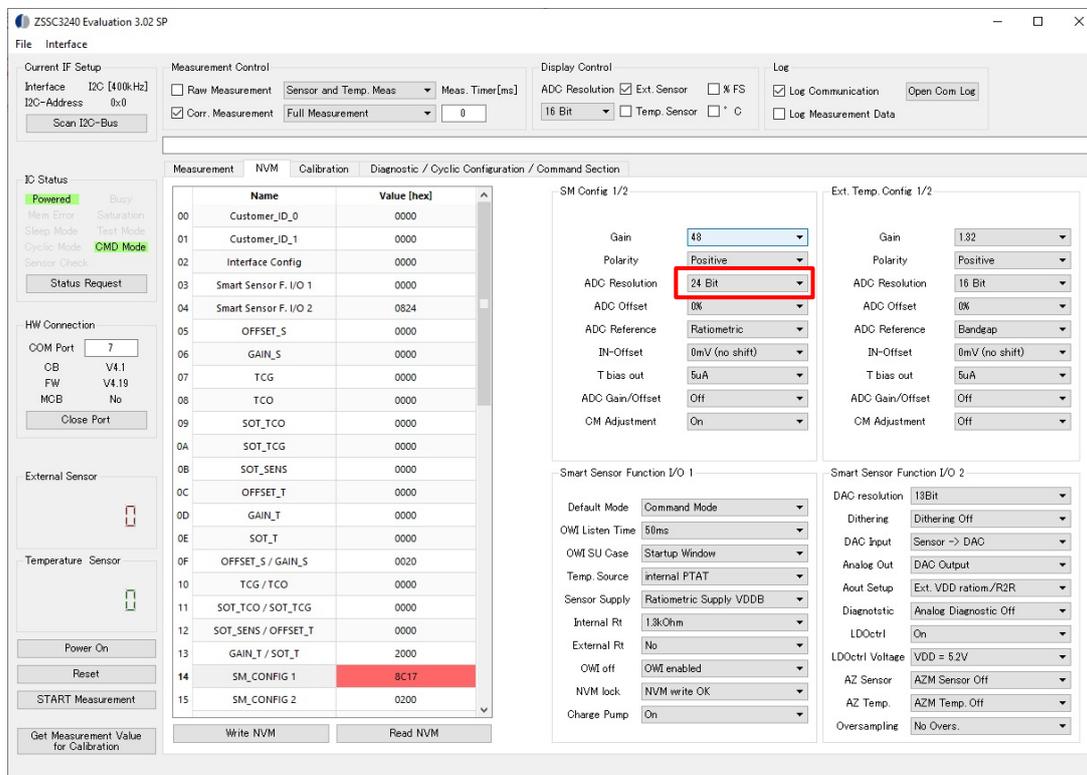


図 9-24 ADC 分解能変更

[Write NVM]をクリックします。ZSSC の NVM レジスタに変更内容が転送されます。転送が完了すると、ハイライトされていた変更箇所の赤色背景が消灯します。

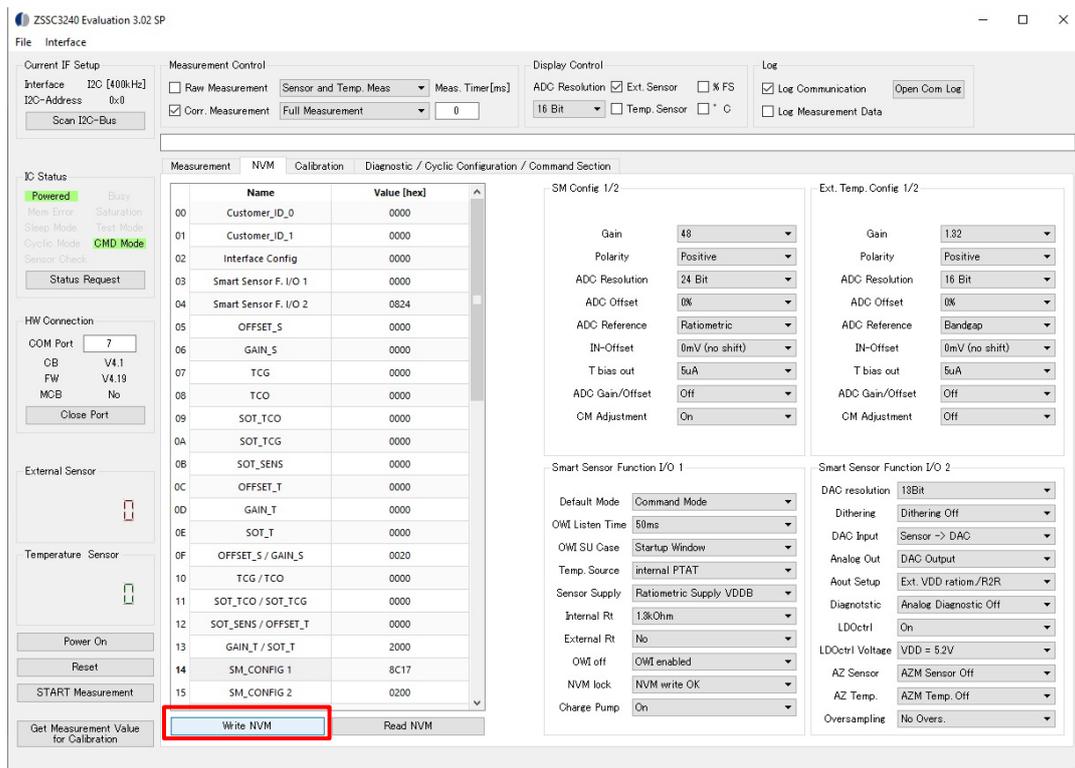


図 9-25 Write NVM

【注】 [Read NVM]のよる設定値読み込みを行っていない状態で、[Write NVM]を実行した場合、既存の値を意図せず上書きしてしまう場合があります。必ず[Write NVM]の実行前には[Read NVM]で既存の設定値を読み込んでください。

[Reset]をクリックして ZSSC EVB をリセットします。この操作によって NVM 設定値がシステムに反映されます。

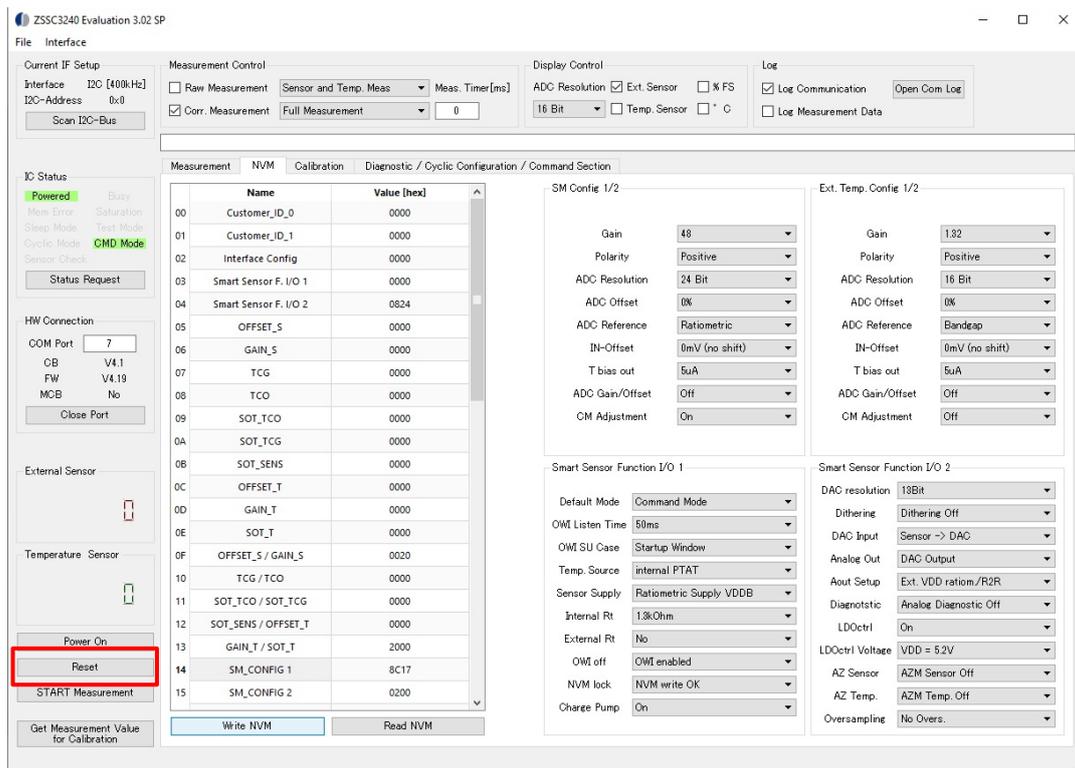


図 9-26 Reset

【注】 [Write NVM]実行のみでは設定が有効になりません。必ず[Reset]ボタンで ZSSC EVB をリセットしてください。

9.6.3 Write CRC

NVM 変更で CheckSum 不整合の場合、[Reset]ボタン操作後に IC Status の Mem Error が点灯します。

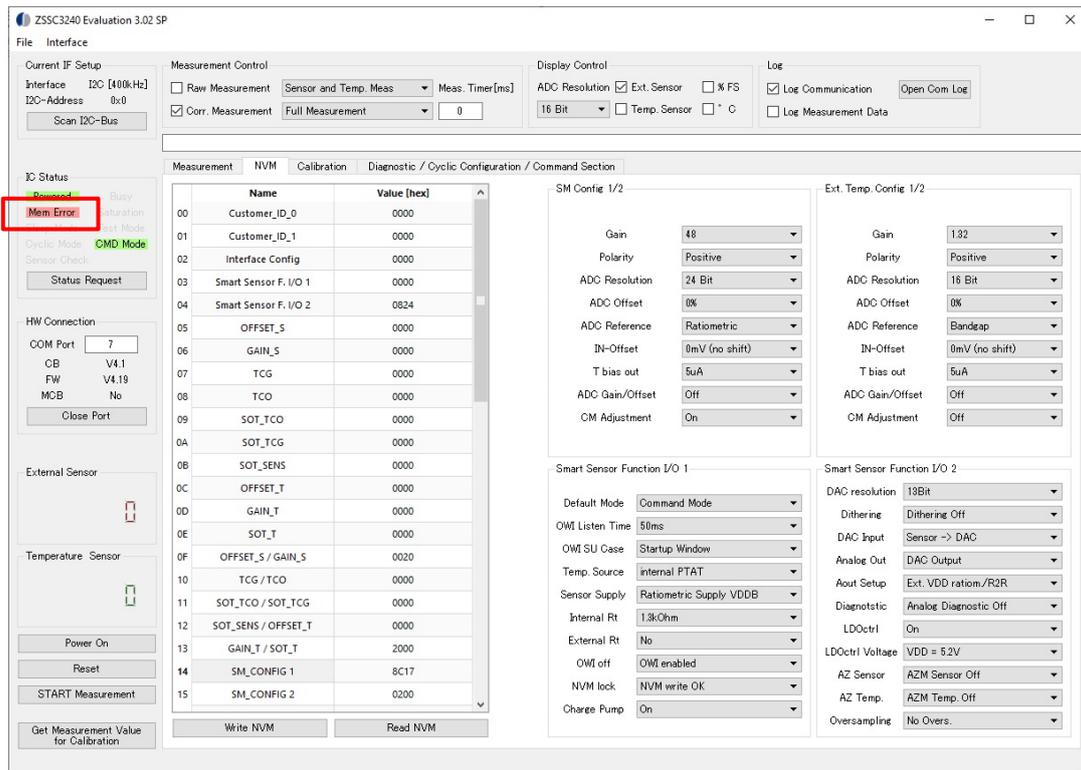


図 9-27 Mem Error

その場合は[Diagnostic / Cycle Configuration / Command Section]タブを選択し[Write CRC]をクリックすることで CheckSum が再計算され NVM に書き込まれます。ZSSC の NVM で CheckSum が正常に再設定された場合、[Reset]ボタンのクリック後に、MemError が消灯します。

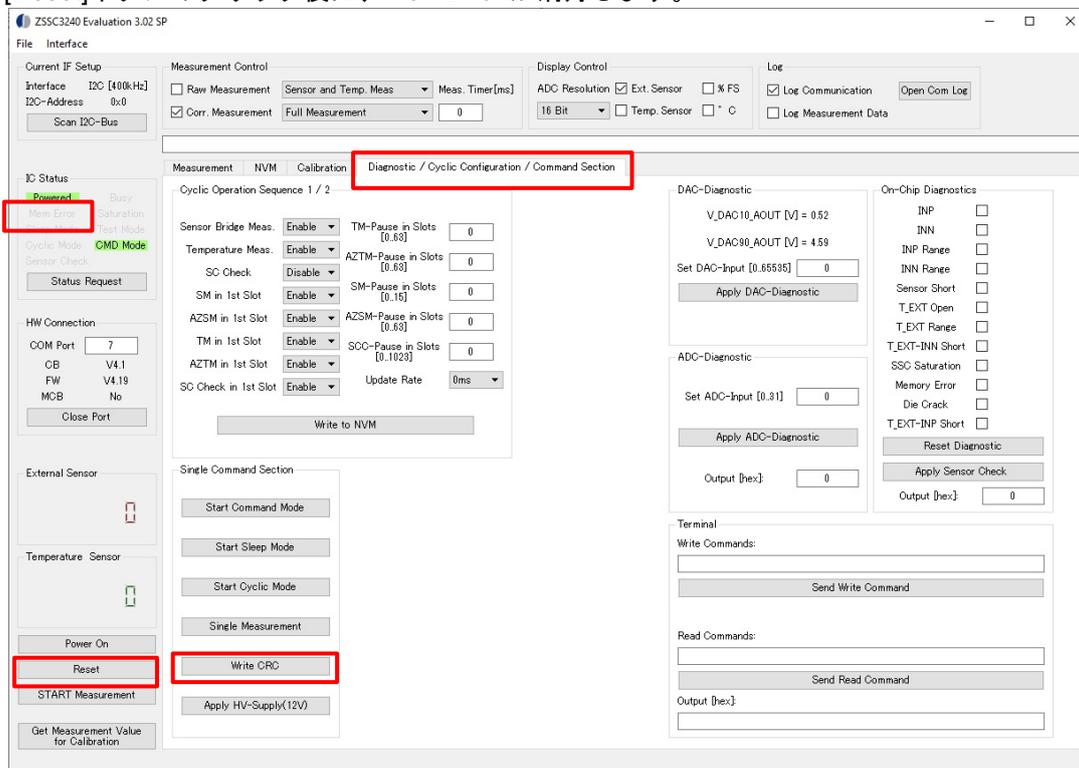


図 9-28 Write CRC

9.6.4 ADC 分解能設定値の戻し

「9.6.2 Write NVM」では NVM 変更の例として、ADC 分解能を 24Bit に変更する操作を解説しました。「8.2 操作方法」用に、変更前の 16Bit に戻しておきます。以下の手順で ADC 分解能を 16Bit に設定してください。

1. 「9.6.2 Write NVM」の手順に従い ADC 分解能を 16Bit に変更します。
2. 「9.6.3 Write CRC」の手順に従い MemError の修正とリセットを行います。

改訂記録

| Rev. | 発行日 | 改訂内容 | |
|------|-----------|------|------------|
| | | ページ | ポイント |
| 1.00 | Sep.16.22 | - | 第 1.00 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まれます。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、変更、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、変更、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）、交通管制（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、これに限られません。）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た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。