

## RX23E-A グループ

### IO-Link デバイス（温度センサー）例

---

#### 要旨

本アプリケーションノートでは、Renesas Solution Starter Kit for RX23E-A（RSSKRX23E-A）と IA Sensor Network Connector Board を使用し、RX23E-A で IO-Link 通信を実現するサンプルプログラムについて説明します。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

RX23E-A, ZIOL2401 (IO-Link PHY)

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	5
2. 関連ドキュメント	6
3. 動作確認環境	7
4. サンプルアプリケーション概要	10
4.1 全体処理フロー概要	11
4.2 App_Temperature 関数フローチャート	13
4.3 App_Cmd 関数フローチャート	19
4.4 動作モードとスイッチング状態	21
4.4.1 Window モード	22
4.4.2 Two point モード	23
4.5 IO-Link 通信仕様	24
4.5.1 ビットレート	24
4.5.2 SIO モード	24
4.5.3 プロセスデータ (Input)	24
4.5.4 パラメータ	26
4.6 使用する周辺機能と端子	28
4.7 プログラム構成	29
4.7.1 ファイル構成	29
4.7.2 関数一覧	30
4.7.2.1 main.c	30
4.7.2.2 r_temperature_measurementc	31
4.7.2.3 r_led_api.c	32
4.7.2.4 IOlinkMain.c	33
5. 動作確認手順	35
5.1 PC の IP アドレス設定	35
5.2 IO-Link Device Tool V5.1 – PE を起動	36
5.3 IO-Link Device Catalog の更新	37
5.4 IO-Link Master Catalog の更新	38
5.5 Catalog の更新の確認	39
5.6 IO-Link 通信のセットアップ	40
5.7 IO-Link デバイスツールの RSSKRX23E-A ボードセンサーデモ	45
5.7.1 Common タブ	46
5.7.2 Process Data タブ	47
5.7.3 Identification タブ	48
5.7.4 Observation タブ	49
5.7.5 Parameter タブ	50
5.7.6 Scope タブ	51
5.7.7 デバイスのパラメータ変更 (Teach-In/Read)	52
5.7.7.1 Teach Values (block write mode) でのパラメータ設定	52
5.7.7.2 Teach Values (direct write mode) でのパラメータ設定	53

---

5.7.7.3 Standard Command でのパラメータ設定 .....	54
5.7.7.4 パラメータの読み取り .....	55
5.7.8 工場出荷設定への復元 .....	55
5.7.9 Generic タブ .....	55
5.7.10 IODD タブ .....	55

## 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 (R01AN4747JJ0100) をベースにし、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 RSSKRX23E-A ボードと IA Sensor Network Connector Board を IO-Link デバイスとし、ルネサスエレクトロニクス製 IO-Link Master Development Kit (RZ/N1S-IO-LINK-M ボード) 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。

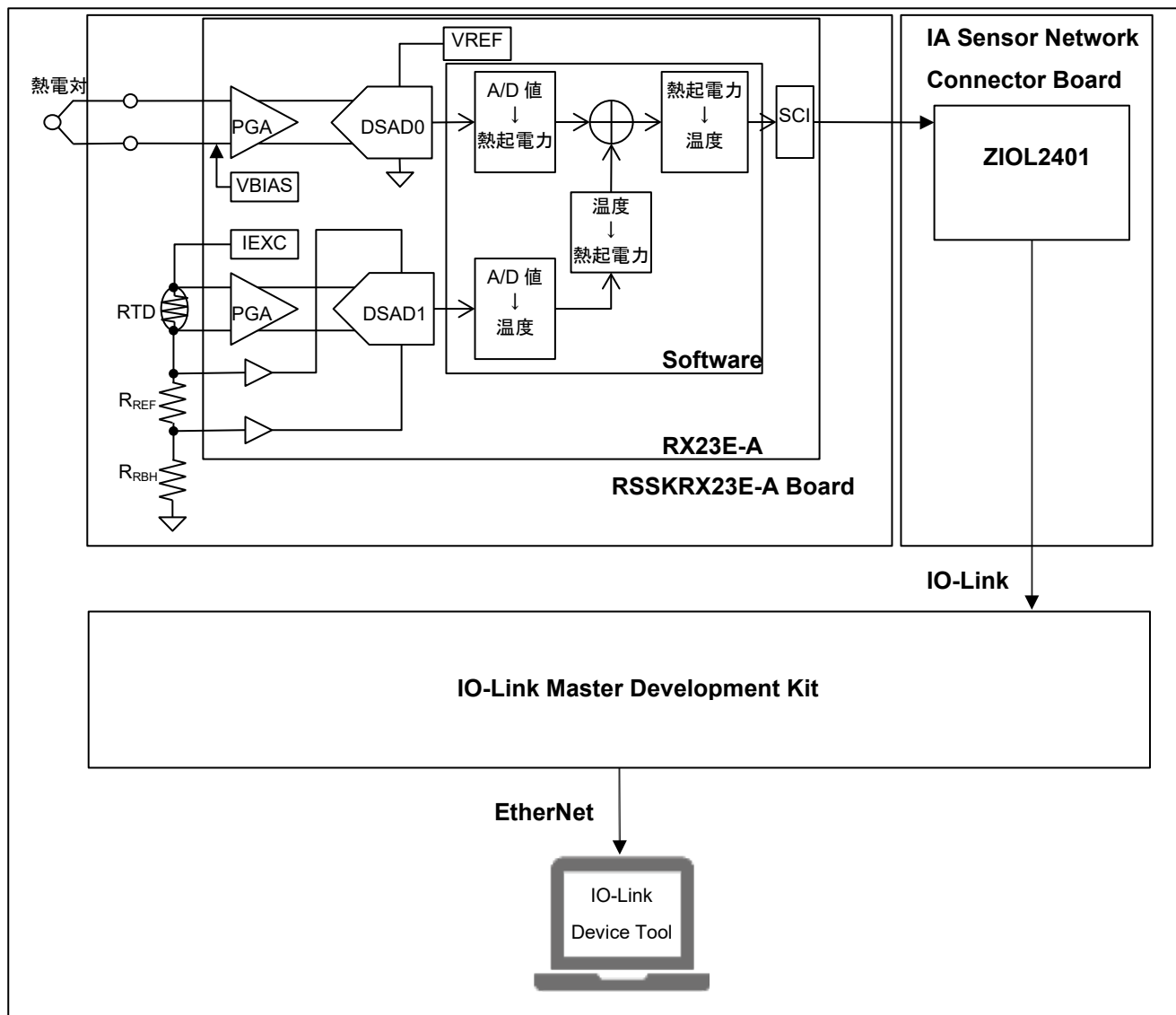


図 1-1 本アプリケーションノートにおけるシステム構成

## 2. 関連ドキュメント

- RX23E-A グループ ユーザーズマニュアル ハードウェア編 R01UH0801JJ0100
- RSSKRX23E-A ユーザーズマニュアル ボード編 R20UT4542JJ0100
- RX23E-A グループ 熱電対を使用した温度計測例 R01AN4747JJ0100

## 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、この構成の機器の画像を図 3-1 に記載します。

表 3-1 動作確認環境

項目	説明
CPU ボード	RSSKRX23E-A ボード (RTK0ESXB10C00001BJ) IA Sensor Network Connector Board と接続し、IO-Link デバイスとして動作します。
MCU	RX23E-A (R5F523E6ADFL) 電源電圧(VCC, AVCC0) : 5V 動作周波数(ICLK) : 32MHz 周辺動作周波数(PCLKB) : 32MHz DSAD 動作周波数( $f_{DR}$ ) : 4MHz DSAD モジュレータクロック周波数( $f_{MOD}$ ) : 0.5MHz
RTD (on board)	Vishay PTS060301B100RP100
熱電対	Labfacility Ltd XE-3505-001
IDE	Renesas e <sup>2</sup> studio 2020-07 Renesas Smart Configurator 20.7.0.v20200629-0858
Tool Chain	Renesas CC-RX V3.02.00
ライブラリ	TMG 社製 IO-Link スタック
エミュレータ	E1 エミュレータ
IO-Link 通信ボード	IA Sensor Network Connector Board (RTK0EF0085B00001BJ) RSSKRX23E-A 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	RZ/N1S-IO-LINK-M ボード
IO-Link ツール	TMG 社製 IO-Link Device Tool V5.1 - PE
IO-Link ツール用ホスト PC	Windows10 Professional

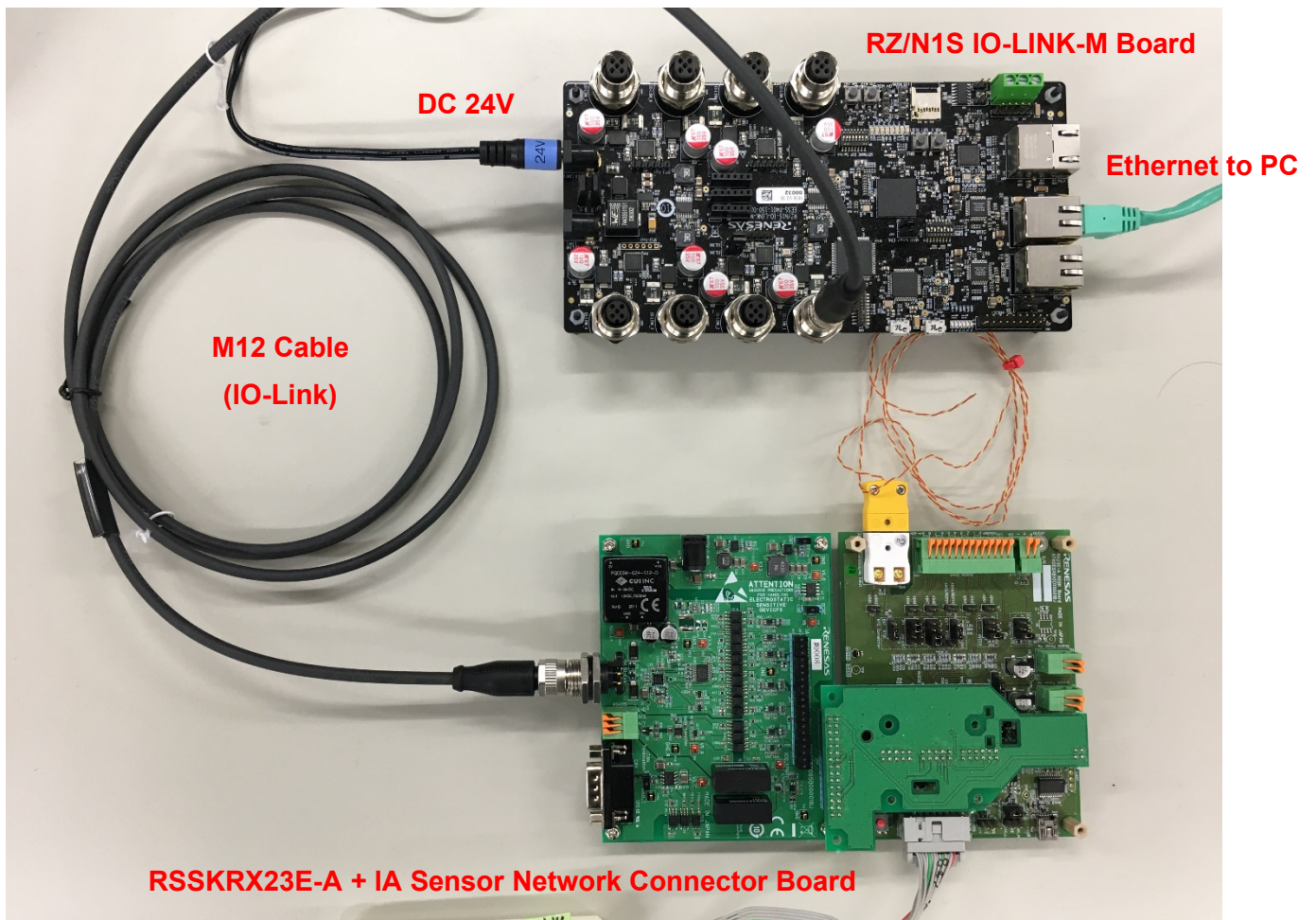


図 3-1 システム全体図



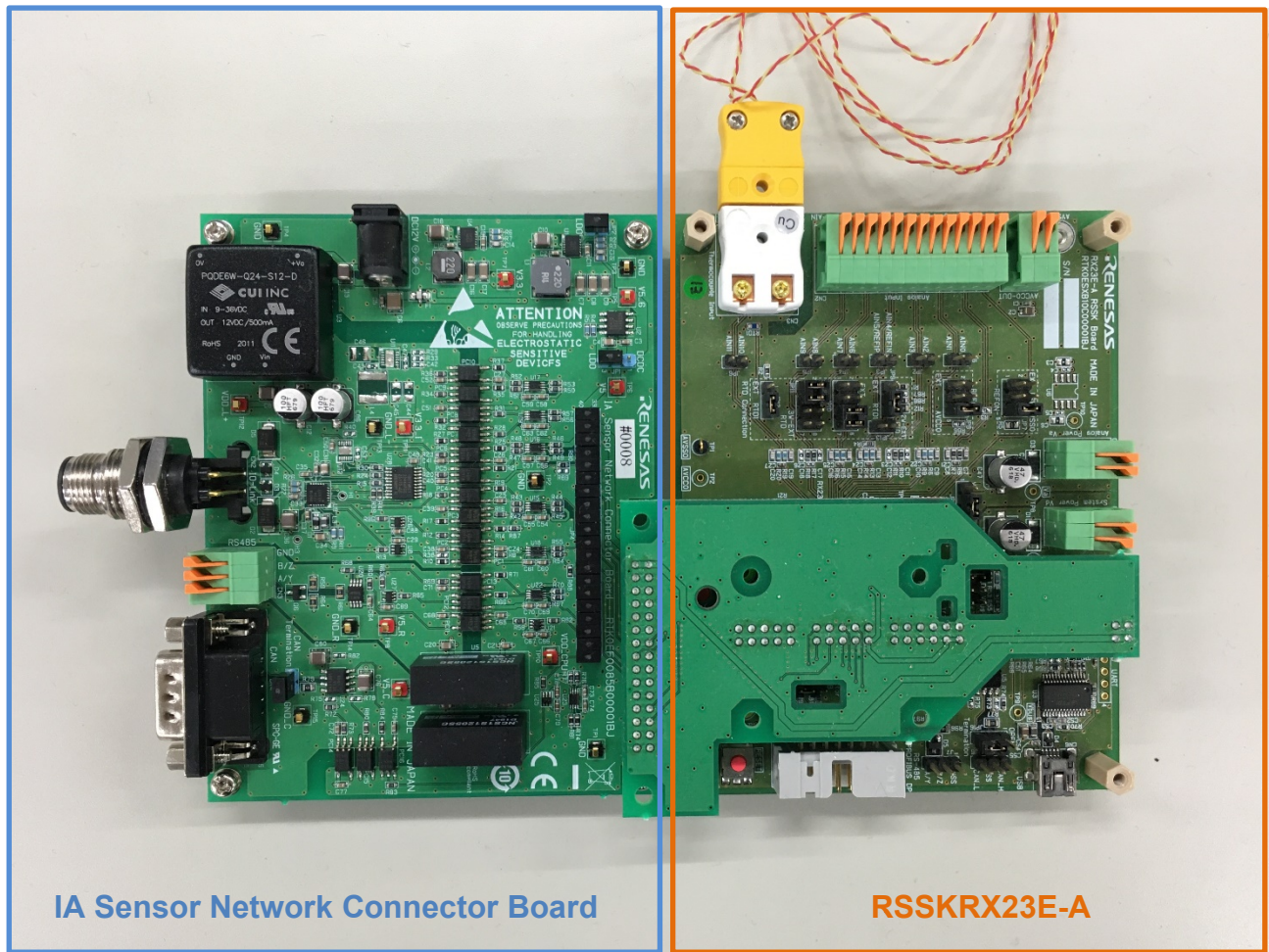


図 3-2 RSSKRX23E-A+IA Sensor Network Connector Board

#### 4. 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 デバイスは温度の測定と閾値判定を定期的に（100[ms]に 1 回）実行し、IO-Link で情報を送信します。送信する情報（プロセスデータ）は、温度データ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。

4.1 全体処理フロー概要

図 4-1 にサンプルアプリケーションの処理フローを記載します。

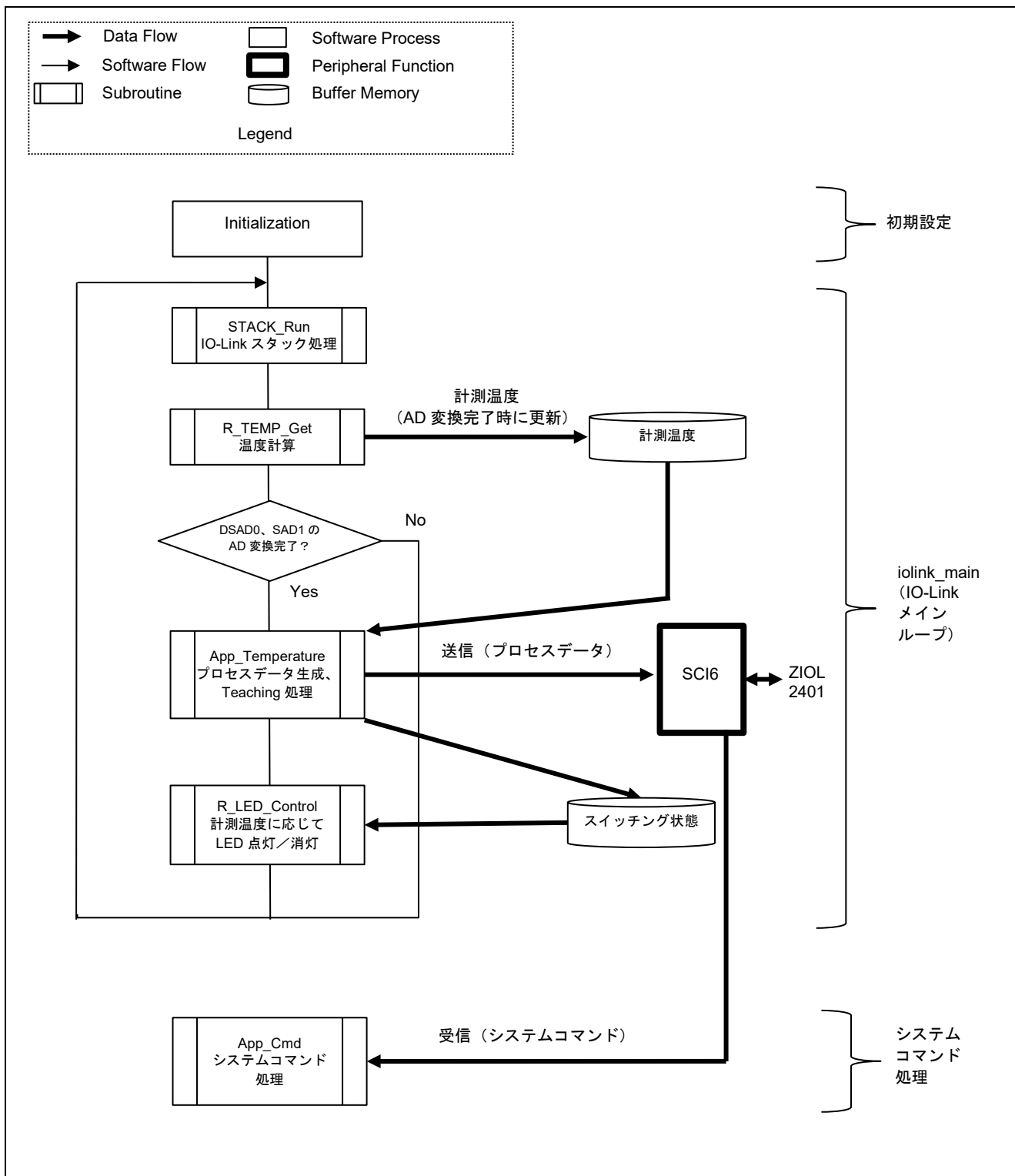


図 4-1 サンプルアプリケーションの処理フロー

各処理の概略を記載します。

### 1. 初期設定

DSAD0 と DSAD1 の同期スタートを有効にします。

DSAD0 と DSAD1 の A/D 変換を開始します。

FIT の Flash モジュールの初期化処理を行います。

データフラッシュからデータを読み出し、変数に格納します。

### 2. iolink\_main

IO-Link サンプルアプリケーションのメインループです。各処理の概要を記載します。

#### A) IO-Link スタックの Run 関数の実行 (STACK\_Run)

IO-Link スタックが提供している API、STACK\_Run を実行します。この関数は定期的に行う必要があります。

#### B) 温度計算 (R\_TEMP\_Get)

DSAD0 と DSAD1 の A/D 変換が完了していたら、A/D 値から温度を計算して、C)、D)の処理を行います。A/D 変換が完了していない場合には、A)に戻ります。

#### C) プロセスデータ生成、Teaching 処理 (App\_Temperature)

計測温度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。

動作モードと計測温度から、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。

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、ここまでの処理で作成したプロセスデータのポインタを渡します。

この処理の詳細については図 4-2~図 4-5 を参照してください。

#### D) LED 点灯/消灯 (R\_LED\_Control)

C)の処理でスイッチング状態が戻り値で取得できるため、R\_LED\_Control の引数にこれを渡します。スイッチング状態が ON であれば LED1 が点灯、OFF であれば、ED1 が消灯します。

### 3. システムコマンド処理 (App\_Cmd)

IO-Link でシステムコマンドを受信した場合に、IO-Link スタックから実行されます。Teaching コマンドが実行中でない場合 (IDLE 状態) のみ受け付けます。対応しているコマンドは SP1 の Teaching コマンド (65)、SP2 の Teaching コマンド (66) の 2 つのみです。これらを受信した場合には、SP1 または SP2 の Teaching コマンド実行中とします。

4.2 App\_Temperature 関数フローチャート

図 4-2~図 4-5 に App\_Temperature 関数のフローチャートを記載します。

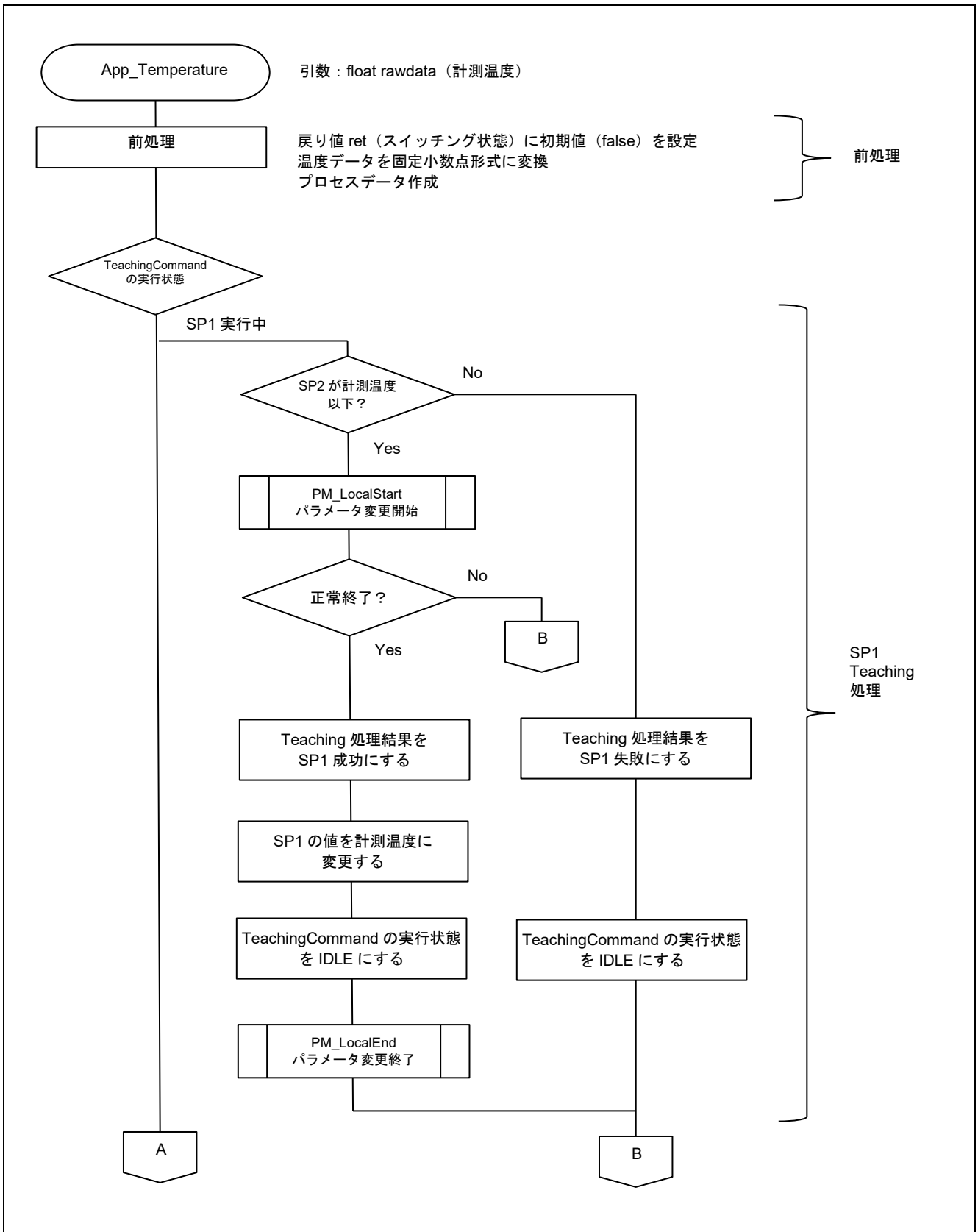


図 4-2 App\_Temperature フローチャート (1/4)

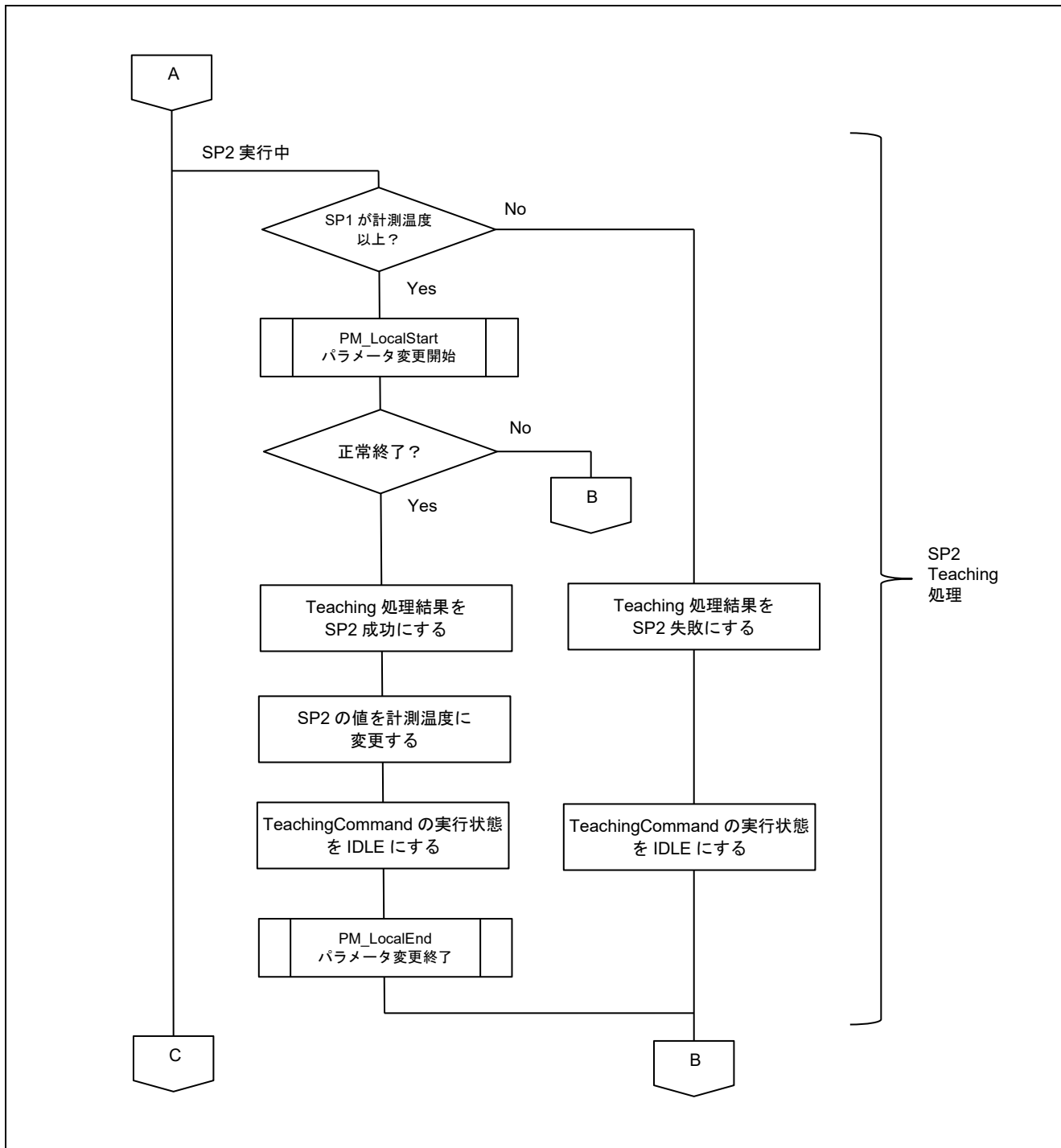


図 4-3 App\_Temperature フローチャート (2/4)

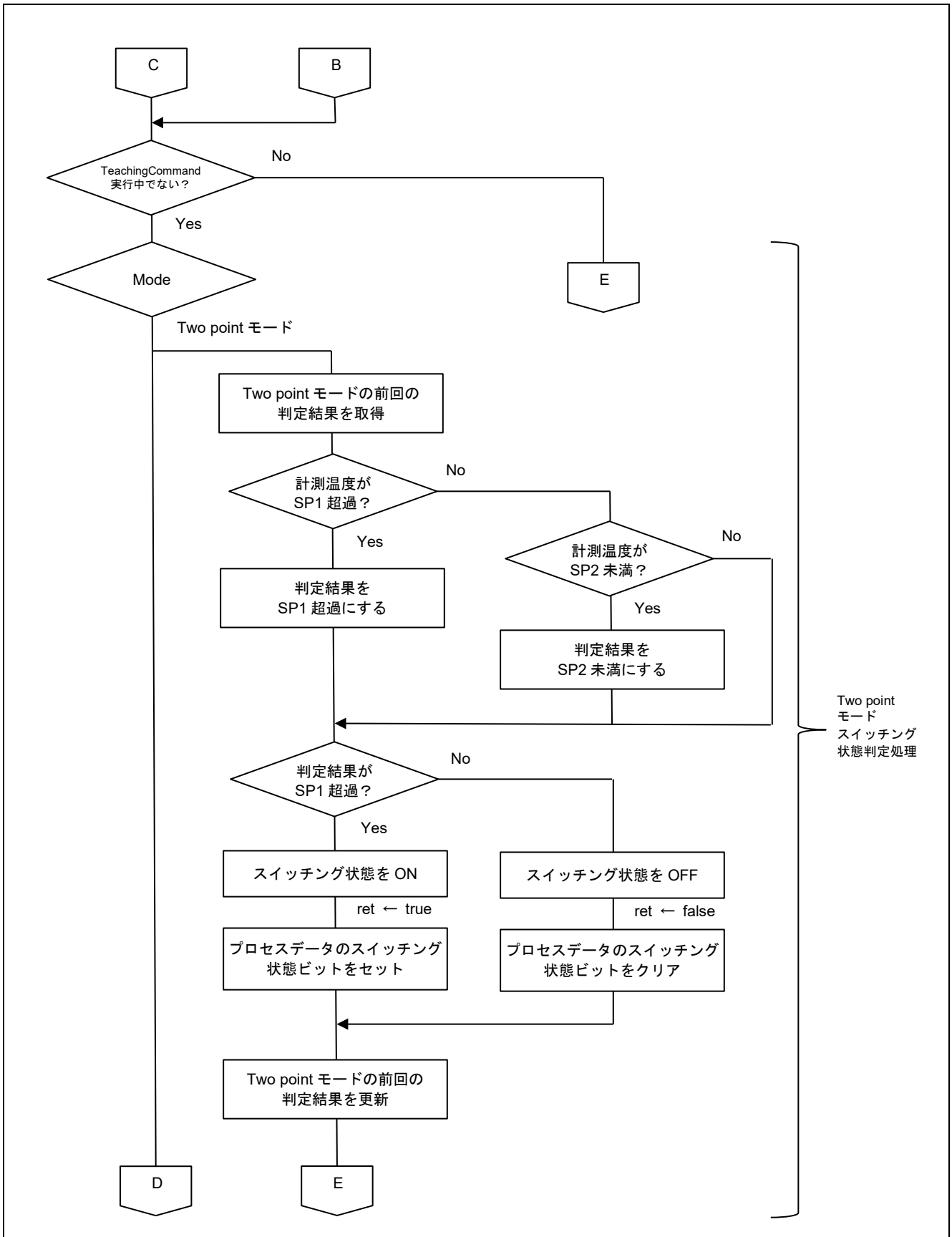


図 4-4 App\_Temperature フローチャート (3/4)

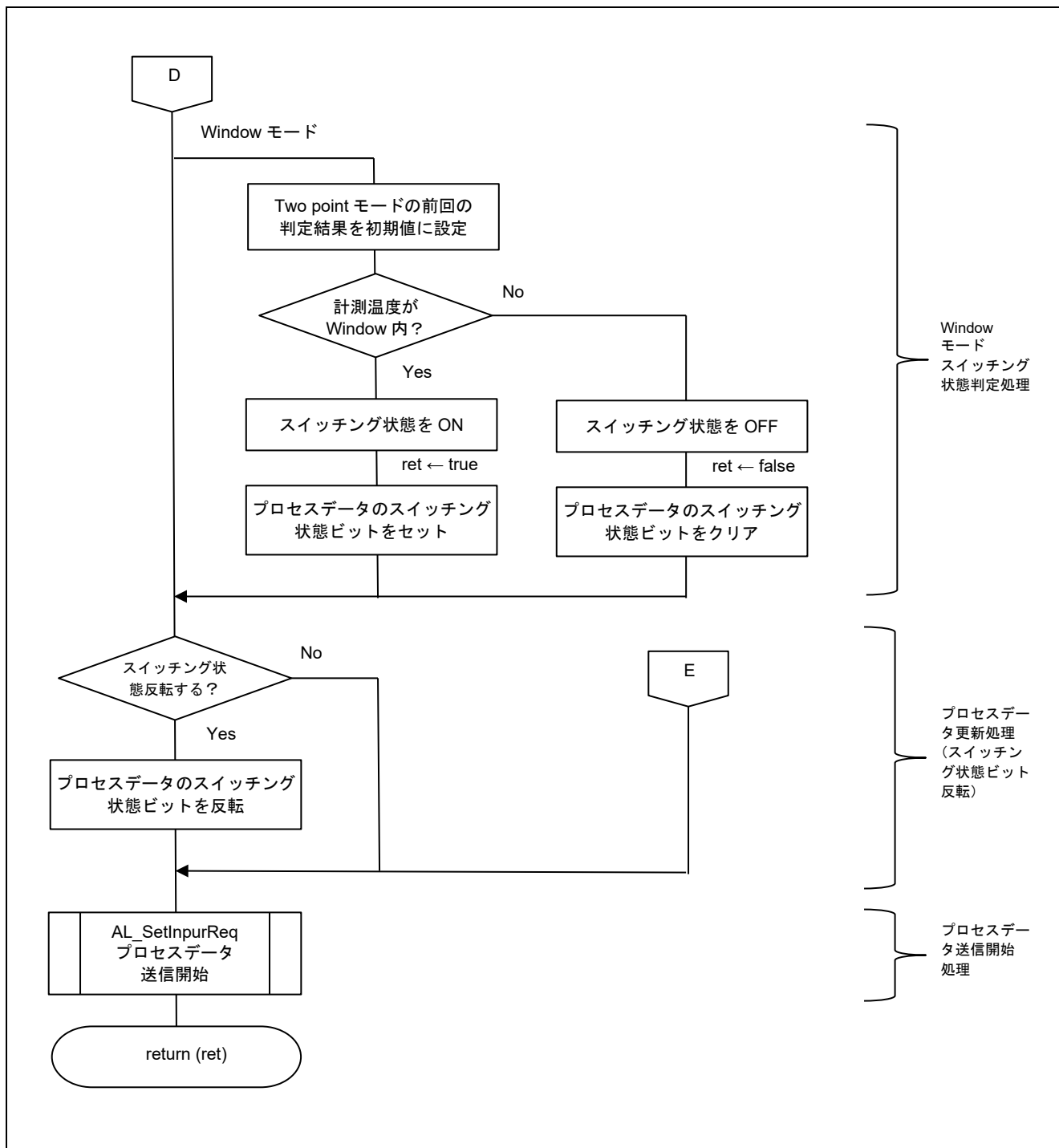


図 4-5 App\_Temperature フローチャート (4/4)



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。

引数で渡された計測温度 (float) に 100 を掛けて、int16\_t にキャストし、0.01[°C] 単位の固定小数点形式に変換します。

プロセスデータを作成します。

ProcessData.PDIn[16] : 温度データの上位 8bit を格納します。

ProcessData.PDIn[17] : 温度データの下部 8bit を格納します。

ProcessData.PDIn[18] : 最下位 bit がスイッチング状態です。初期値は 0 とします。

#### 2. SP1 Teaching 処理<sup>※注意</sup>

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、計測温度 < 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) に計測温度 (int16\_t) 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<sup>※注意</sup>

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 < 計測温度の場合には、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) に計測温度 (int16\_t) 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 < 計測温度の場合には、判定結果を SP1 超過とします。

SP2 > 計測温度の場合には、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、Switching 状態を ON として、ProcessData.PDIn[18] を 1 にします。

判定結果が SP2 未満の場合には、Switching 状態を OFF として、ProcessData.PDIn[18] 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 スイッチング状態判定処理

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、初期値の SP2 未満に設定します。SP1 $\geq$ 計測温度 T[°C] $\geq$ SP2 の場合には、スイッチング状態を ON として、ProcessData.PDIn[18]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[18]を 0 にします。

#### 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、以下の処理を行います。ProcessData.PDIn[18]の最下位ビットを反転します。

#### 7. プロセスデータ送信開始処理

ここまでで作成したプロセスデータ（ProcessData.PDIn）のポインタを、IO-Link スタックが提供する AL\_SetInputReq 関数に渡し、IO-Link マスターへのプロセスデータの送信を開始します。

#### ※注意：

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\_Temperature 関数を抜け、次の温度計測後の処理で Teaching 処理をリトライします。

4.3 App\_Cmd 関数フローチャート

図 4-6 に App\_Cmd 関数のフローチャートを記載します。

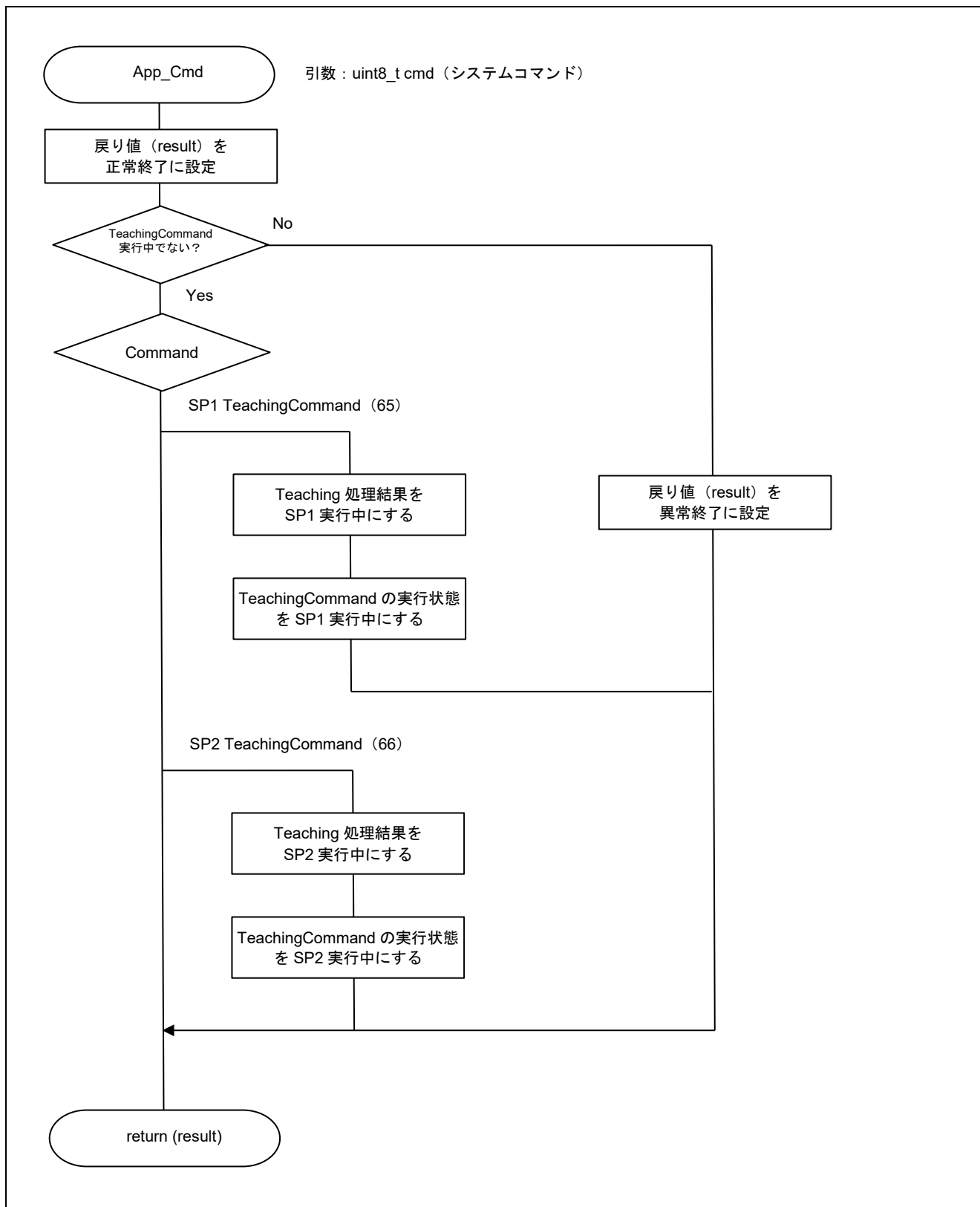


図 4-6 App\_Cmd フローチャート

処理の概略を記載します。

TeachingCommand 実行中の場合（IDLE 以外）には処理を行わず、異常終了を返します。

TeachingCommand 実行中でない場合（IDLE）には以下の処理を行います。

SP1 の TeachingCommand（65）の場合には、Teaching 処理の結果を「SP1 実行中」として、TeachingCommand の実行状態も「SP2 実行中」とします。

SP2 の TeachingCommand（66）の場合には、Teaching 処理の結果を「SP2 実行中」として、TeachingCommand の実行状態も「SP2 実行中」とします。

App\_Temperature 関数で TeachingCommand の実行状態を監視し、必要に応じて Teaching 処理を実行しているため、IO-Link でシステムコマンドを受信して App\_Cmd 関数が実行された後の App\_Temperature 関数で Teaching 処理が実行されます。

#### 4.4 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、Windows 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。

## 4.4.1 Window モード

温度[°C]が指定された範囲内（SP1～SP2の間）であればスイッチング状態 ON となる動作モードです。具体的な判定は以下のようにになります。動作イメージ図を図 4-7 に記載します。

$\text{Param1(SP1)} \geq \text{計測温度 } T[^\circ\text{C}] \geq \text{Param2(SP2)}$  : スwitchング ON

$\text{Param1(SP1)} < \text{計測温度 } T[^\circ\text{C}]$ 、または  $\text{Param2(SP2)} > \text{計測温度 } T[^\circ\text{C}]$  : スwitchング OFF

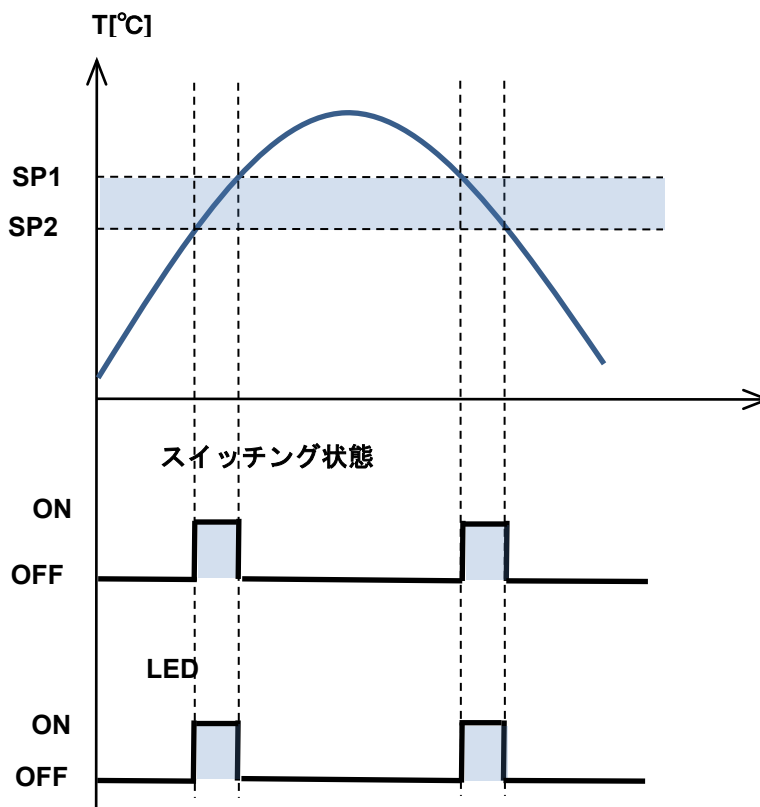


図 4-7 動作イメージ図 (Window モード)

## 4.4.2 Two point モード

温度[°C]が指定された値（SP1）を超過したらスイッチング状態 ON、指定された値（SP2）を下回ったらスイッチング状態 OFF となる動作モードです。具体的な判定は以下のようになります。動作イメージ図を図 4-8 に記載します。

Param1(SP1) < 計測温度 T[°C] : スwitching 状態 ON

Param2(SP2) > 計測温度 T[°C] : スwitching 状態 OFF

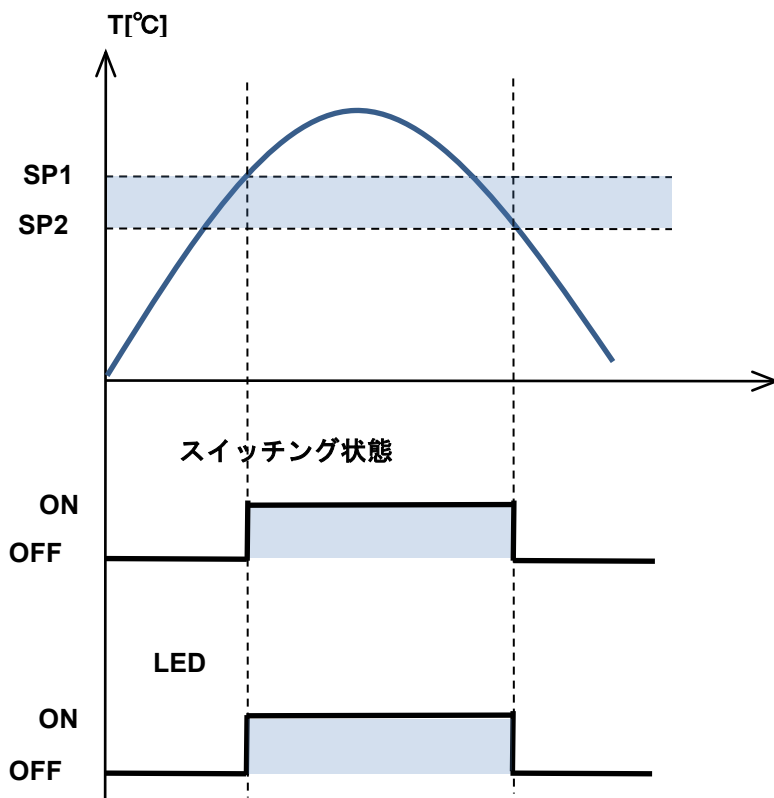


図 4-8 動作イメージ図 (Two point モード)

## 4.5 IO-Link 通信仕様

IO-Link の通信仕様について記載します。

### 4.5.1 ビットレート

ビットレートは COM2 (38.4[kbps]) です。

### 4.5.2 SIO モード

SIO モードは非対応です。

### 4.5.3 プロセスデータ (Input)

プロセスデータ (Input : IO-Link デバイスから IO-Link マスターへ送信されるデータ) の形式を表 4-1 に記載します。

表 4-1 プロセスデータ (Input) 形式

プロセスデータ (Input) データ長 : 3[bytes] (24[bits])				
サブ インデックス	名称	ビット オフセット	ビット数	データ
1	Switch Point 1	0	1	スイッチング状態
2	Temperature	8	16	温度 (0.01[°C]単位)



プロセスデータ (Input) の例を表 4-2 に記載します。

表 4-2 プロセスデータ (Input) 例

PDIn[16] : 16 オクテット								
サブ イン デックス	2							
値	0	0	0	0	1	0	0	1
例	温度データ上位 8bits (0x09)							
PDIn[17] : 17 オクテット								
サブ イン デックス	2							
値	0	0	1	1	0	1	0	0
例	温度データ下位 8bits (0x43) 上位+下位=0x0934 (2356) →23.56[°C]							
PDIn[18] : 18 オクテット								
サブ イン デックス	1							
値	0	0	0	0	0	0	0	1
例	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	スイッチ ング状態

## 4.5.4 パラメータ

IO-Link でマスターと送受信するパラメータ一覧を表 4-3、表 4-4 に記載します。

表 4-3 設定パラメータ一覧 (1/2)

Index (Sub Index)	名称 (型)	ビット オフセット (ビット数)	値域 (初期値)	R/W	単位	概要
Switching Signal Channel1						
60 (1)	Param SP1 (Integer)	16 (16)	-4000 to 15000 (2500) ※4	RW	0.01°C ※4	Switching Point1 (SP1) 設定※1 SP1 ≥ SP2 を満たす必要があります。
60 (2)	Param SP2 (Integer)	0 (16)	-4000 to 15000 (2000) ※4	RW	0.01°C ※4	Switching Point2 (SP2) 設定※1 SP1 ≥ SP2 を満たす必要があります。
61 (1)	Config Logic (UInteger)	24 (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
61 (2)	Config Mode (UInteger)	16 (8)	0, 2, 3 (3)	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 2 : Window モード※2 3 : Two point モード※3

※1. SP1, SP2 については 4.4 動作モードとスイッチング状態を参照してください。

※2. Window モードについては 4.4.1 Window モードを参照してください。

※3. Two point モードについては 4.4.2 Two point モードを参照してください。

※4. Device Tool では-40.00~150.00 の範囲で設定します。初期値は SP1=25.00, SP2=20.00 で表示されます。表示イメージについては 5.7.5 Parameter タブを参照してください。

表 4-4 設定パラメーター一覧 (2/2)

Index (Sub Index)	名称 (型)	ビット オフセット (ビット数)	値域 (初期値)	R/W	単位	概要
Teach-In Single Value						
2 (-)	Standard Command (UInteger)	- (-)	65 (-)	WO	-	現在の計測温度を SP1 に設定する。
2 (-)	Standard Command (UInteger)	- (-)	66 (-)	WO	-	現在の計測温度を SP2 に設定する。
59 (1)	Teach-In Result : State (UInteger)	0 (4)	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
Device Locks						
12 (3)	Device Access Locks Local Parameterization Lock (Boolean)	2 (1)	0, 1 (0)	RW	-	センサーのパラメータ化機能の有効／無効設定 0 : 無効 1 : 有効

## 4.6 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 4-5 に、使用端子一覧を表 4-6 に記載します。

表 4-5 使用する周辺機能一覧

周辺機能	用途
AFE、DSAD0、DSAD1	熱電対、RTD の駆動、熱電対の A/D 変換(DSAD0)、RTD の A/D 変換(DSAD1)
IRQ1	ZIOL2401 の DC/DC Ready 状態検出
IRQ4	IO-Link の WakeUp 検出
SCI6	ZIOL2401 との SPI 通信 (レジスタ設定) 及び UART 通信 (IO-Link)
P17	ZIOL2401 のリセット制御 (L 出力でリセット解除)
PB1	ZIOL2401 の SPI 通信有効/無効制御 (L 出力で有効)
PC5	ZIOL2401 の UART 通信有効/無効制御 (H 出力で有効)
PH2	LED1 の制御

表 4-6 使用端子一覧

端子名	入出力	用途
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
P17	出力	ZIOL2401 のリセット制御端子 (L 出力でリセット解除)
P31/IRQ1	入力	ZIOL2401 の DC/DC Ready 信号入力端子
PB0/IRQ4	入力	IO-Link の WakeUp 検出信号入力端子
PB1	出力	ZIOL2401 の SPI 通信有効/無効制御端子 (L 出力で有効)
PC5/SCK6	出力	ZIOL2401 の UART 通信有効/無効制御端子 (H 出力で有効) SPI 通信時は CLK 出力端子
PC6/RXD6/SMISO6	入力	UART6 受信端子 SPI 通信時はデータ受信端子
PC7/TXD6/SMOSI6	出力	UART6 送信端子 SPI 通信時はデータ送信端子
PH2	出力	LED1 の制御

## 4.7 プログラム構成

## 4.7.1 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、Library フォルダ内の Manuals フォルダ内に配置されていますので、必要に応じて参照してください。src2 フォルダ、StackExtensionsApp フォルダ内のファイル構成を表 4-7 に記載します。

表 4-7 ファイル構成

フォルダ名、ファイル名	説明
src2	
main.c	メイン処理
r_led_api.c	LED 制御処理プログラム
r_led_api.h	LED 制御処理 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_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_temperature_measurement_api.c	温度計測処理プログラム
r_temperature_measurement_api.h	温度計測処理 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
└smc_gen	Smart Configurator 生成
└Config_CMT0	
└Config_DMAC0	
└Config_DMAC3	
└Config_DSAD0	
└Config_DSAD1	
└Config_IOL_MTU0	
└Config_PORT	
└Config_SCI1	
└Config_SCI6	
└Config_SCI61	
└general	
└r_bsp	
└r_config	
└r_flash_rx	
└r_pincfg	
IO-Link	
└StackExtensionsApp	
└BSPExtensions.h	BSP 機能拡張ヘッダ
└IOLinkMain.c	IO-Link アプリケーションプログラム
└IOLinkMain.h	IO-Link アプリケーション定義
└MemoryManager.h	メモリマネージャーヘッダ
└ParameterSet.h	スタック拡張の定義
└ProductionSettings.h	プロダクション設定ヘッダ

#### 4.7.2 関数一覧

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 (R01AN4747JJ0100) から追加、変更を加えた関数を記載します。

##### 4.7.2.1 main.c

[関数名] main

---

概要	main 関数
ヘッダ	なし
宣言	void main (void)
説明	DSAD0 と DSAD1 の動作を開始し、IO-Link スタックと IO-Link 関連パラメータの初期化を行います。以降は IO-Link アプリケーション処理 (iolink_main 関数) を定期的にコールします。
引数	なし
リターン値	なし
備考	なし

## 4.7.2.2 r\_temperature\_measurementc

[関数名] R\_TEMP\_Start

---

概要	温度計測の開始
ヘッダ	r_temperature_measurement.h
宣言	void R_TEMP_Start (void)
説明	DSAD0 と DSAD1 のユニット間同期スタートを許可し、トリガをソフトウェアトリガに設定した後で DSAD0 と DSAD1 の動作を開始します。
引数	なし
リターン値	なし
備考	なし

[関数名] R\_TEMP\_Stop

---

概要	温度計測の停止
ヘッダ	r_temperature_measurement.h
宣言	void R_TEMP_Stop (void)
説明	DSAD0 と DSAD1 の動作を停止します。
引数	なし
リターン値	なし
備考	なし

[関数名] R\_TEMP\_Get

---

概要	温度の取得
ヘッダ	r_temperature_measurement.h
宣言	bool R_TEMP_Get (float *p_temperature)
説明	DSAD0 と DSAD1 の A/D 変換が完了していれば（温度計測が完了していれば）、引数で渡されたアドレスに計測温度を格納します。
引数	*p_temperature : 温度[°C]格納領域のポインタ
リターン値	計測状態 false : 計測中 true : 計測完了
備考	なし

## 4.7.2.3 r\_led\_api.c

[関数名] R\_LED\_Control

---

概要	LED1 の制御
ヘッダ	r_led_api.h
宣言	void R_LED_Control (bool state)
説明	スイッチング状態に応じて LED を点灯／消灯します。
引数	スイッチング状態 false : OFF true : ON
リターン値	なし
備考	なし



## 4.7.2.4 IOLinkMain.c

[関数名] iolink\_main

---

概要	IO-Link アプリケーションメイン処理
ヘッダ	IOLinkMain.h
宣言	uint8_t iolink_main (void)
説明	IO-Link スタックの定期処理を実行 (STACK_Run 関数をコール) し、温度の取得を行います。温度計測が完了していれば、IO-Link でプロセスデータの送信を行います。
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 : 接続断状態、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

## [関数名] App\_Cmd

---

概要	IO-Link システムコマンド処理
ヘッダ	IOLinkMain.h
宣言	uint16_t App_Cmd (uint8_t command)
説明	システムコマンド処理を行います。
引数	システムコマンド
リターン値	システムコマンド処理結果 RESULT_OK : 正常終了 RESULT_ERR_FUNCTION_TEMP_NA : 無効なコマンド
備考	なし

## [関数名] App\_Temperature

---

概要	IO-Link 温度センサーアプリケーション処理
ヘッダ	なし
宣言	static bool App_Temperature (float rawValue)
説明	以下の 4 つの処理を実行します。 1. プロセスデータの作成 2. Teaching 処理（閾値の設定処理） 3. 閾値判定処理+プロセスデータの更新 4. プロセスデータの送信開始
引数	計測温度[°C]
リターン値	閾値判定結果 false : 有効範囲外（OFF） true : 有効範囲内（ON）
備考	本関数は、iolink_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. 動作確認手順

WindowsPC 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、動作確認の手順を記載します。各機器の接続については図 3-1 を参照してください。

### 5.1 PC の IP アドレス設定

PC と RZ/N1S-IO-LINK-M ボードは同じサブネットワークに属している必要があります。RZ/N1S-IO-LINK-M ボードの IP アドレスは 192.168.0.225 なので、PC は 192.168.0.12 等に設定します。

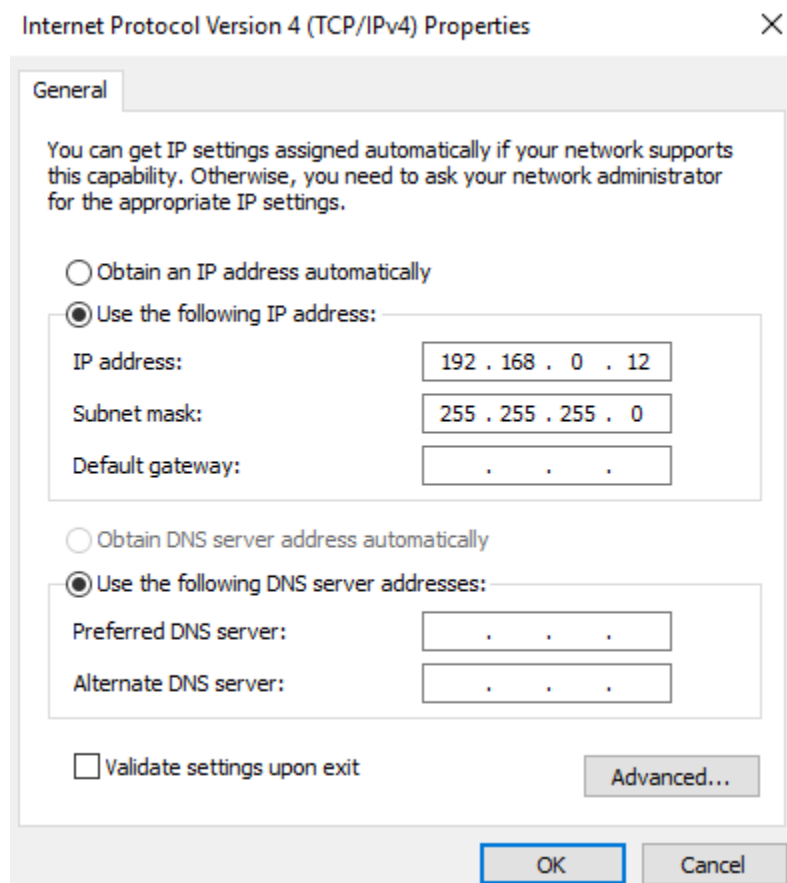


図 5-1 PC の IP アドレス設定

## 5.2 IO-Link Device Tool V5.1 – PE を起動

Topology には PC から IO-Link デバイスのトポロジーが表示されます。

Device Catalog にはインストールされた全てのデバイスが表示されます。



図 5-2 IO-Link Device Tool V5.1 – PE

### 5.3 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. パスを入力するか、IODD ファイルがあるフォルダを選択します。本サンプルアプリケーションの IODD ファイルは、ダウンロードしたサンプルプロジェクトファイル内の「IODD」フォルダにあります。図 5-3 のように、IODD ファイルは自動検出されます。

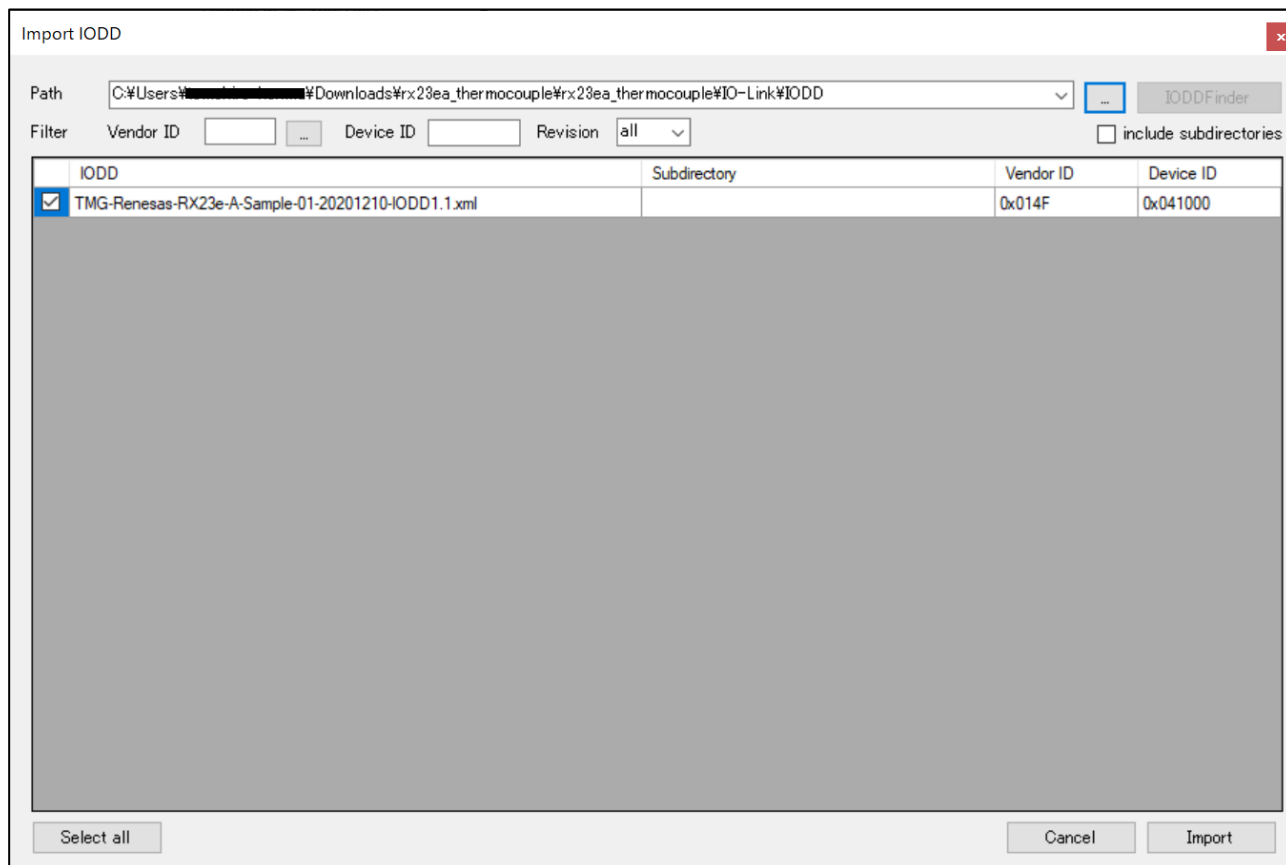


図 5-3 IO-Link Device Tool V5.1 – PE (IODD ファイルのロード)

3. 「Import」を選択します。

## 5.4 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. パスを入力するか、IOLM の zip ファイルがあるフォルダを選択します。RZ/N1S-IO-LINK-M ボードの IOLM ファイルは、ダウンロードしたサンプルプロジェクトファイル内の「IOLM」フォルダにあります。図 5-4 のように、IOLM ファイルは自動検出されます。

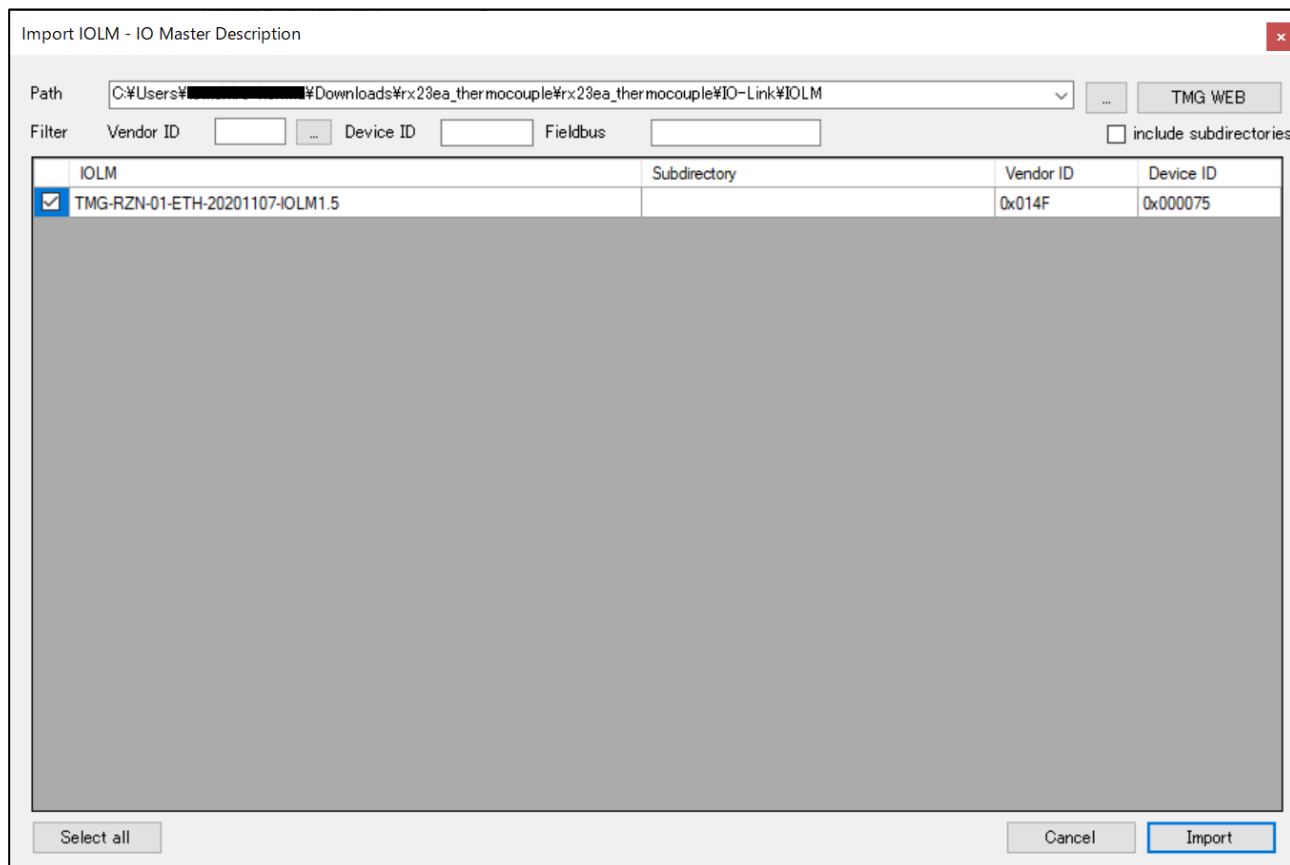


図 5-4 IO-Link Device Tool V5.1 – PE (IOLM ファイルのロード)

**注意:**

Path には解凍していない IOLM の zip ファイルがあるフォルダを指定してください。解凍したフォルダを指定した場合は IOLM ファイルが検索されません。

3. 「Import」を選択します。

## 5.5 Catalog の更新の確認

Catalog の更新が成功すると、Catalog の IO-Link Devices セクションに「RX23e-A Starterkit」として TMG TE GmbH ベンダーと RSSKRX23E-A ボードが表示されます。RZ/N1S IO-Link Master は PROFINET デバイスとして Master の下に表示されます。

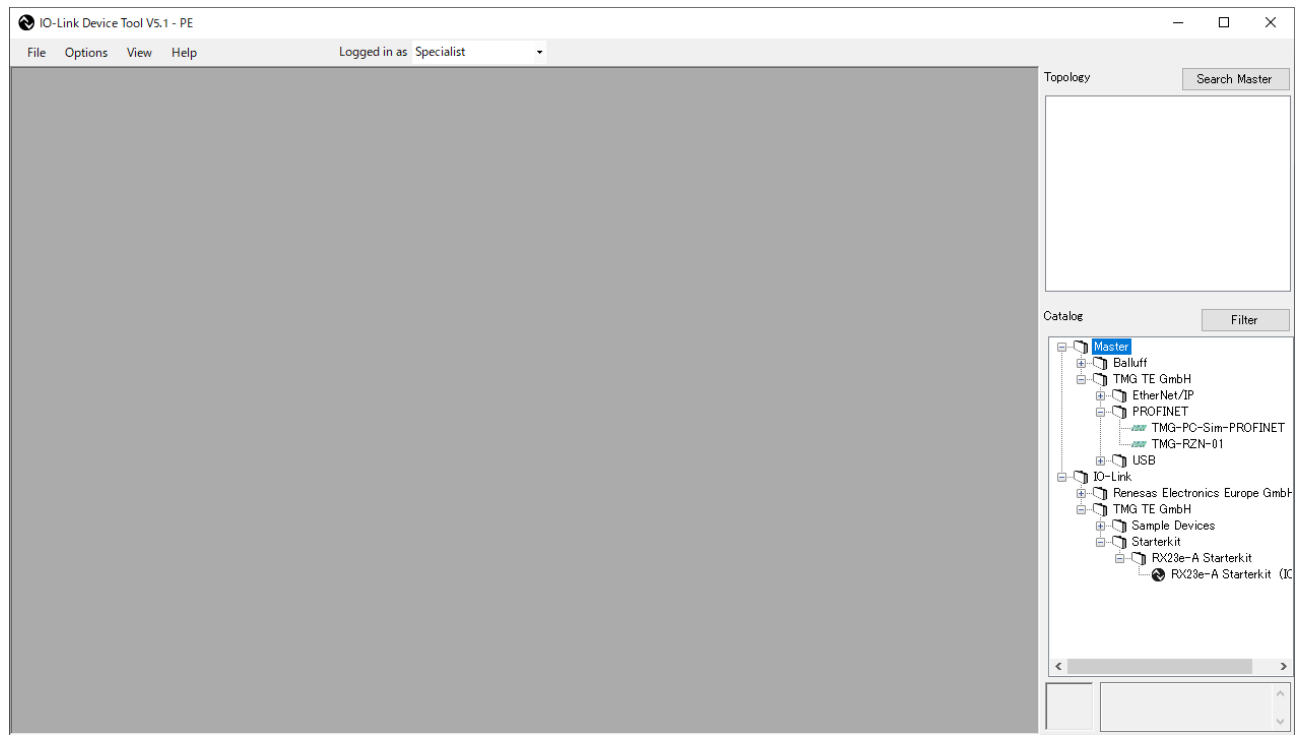


図 5-5 IO-Link Device Tool V5.1 – PE (Catalog の更新の確認)

## 5.6 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 5-6 のように、RZ/N1S-IOlink が「Master Discovery」ウィンドウに表示されます。

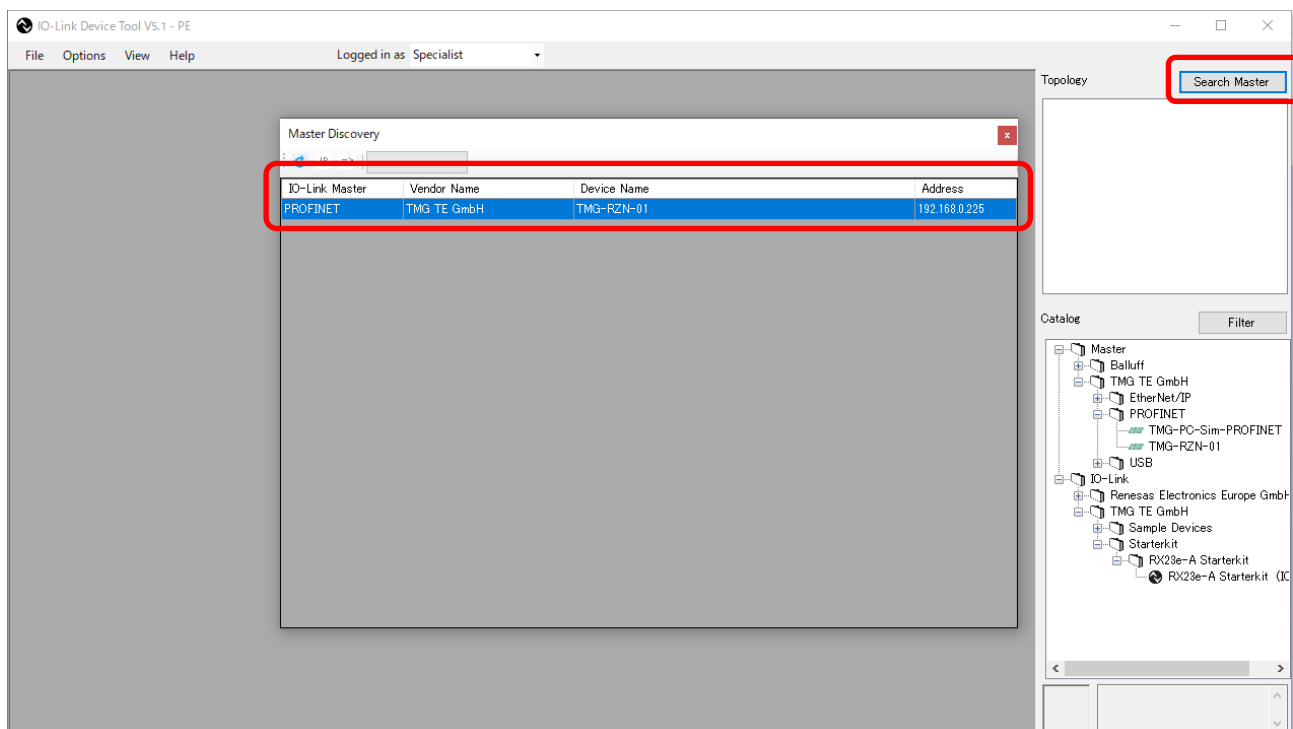


図 5-6 IO-Link Device Tool V5.1 – PE (IO-Link マスターの検索)

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。

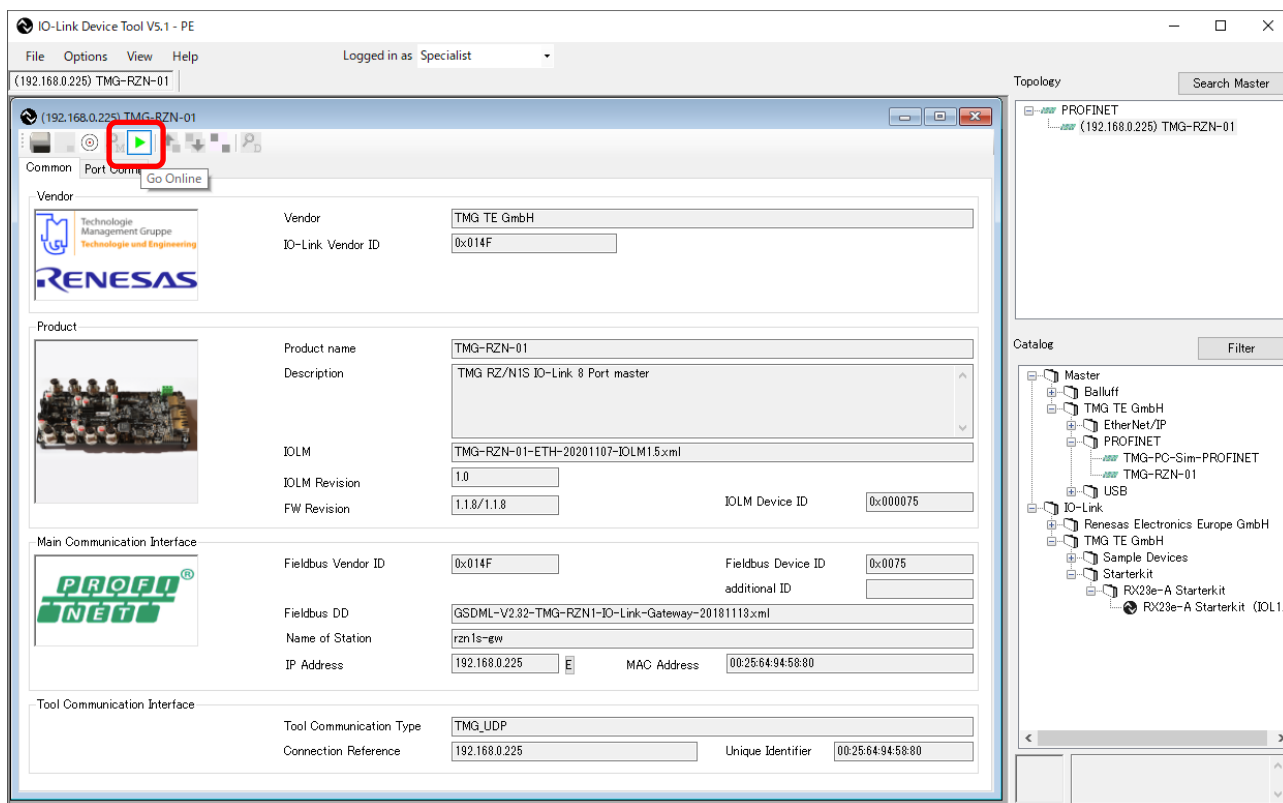


図 5-7 IO-Link Device Tool V5.1 – PE (Online 状態に設定)

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。

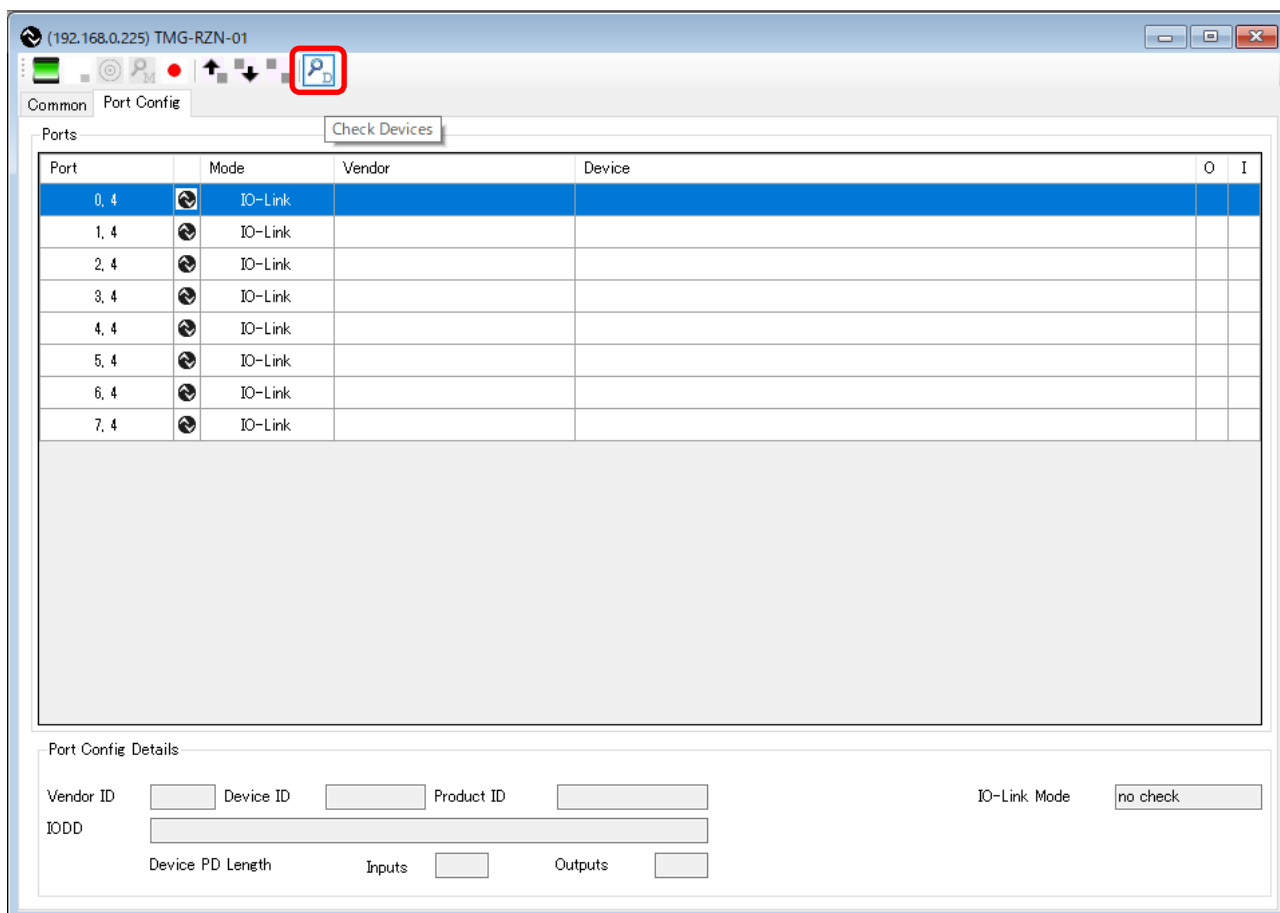


図 5-8 IO-Link Device Tool V5.1 – PE (IO-Link デバイスの確認)

「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 RX23e-A Starterkit が表示されます。

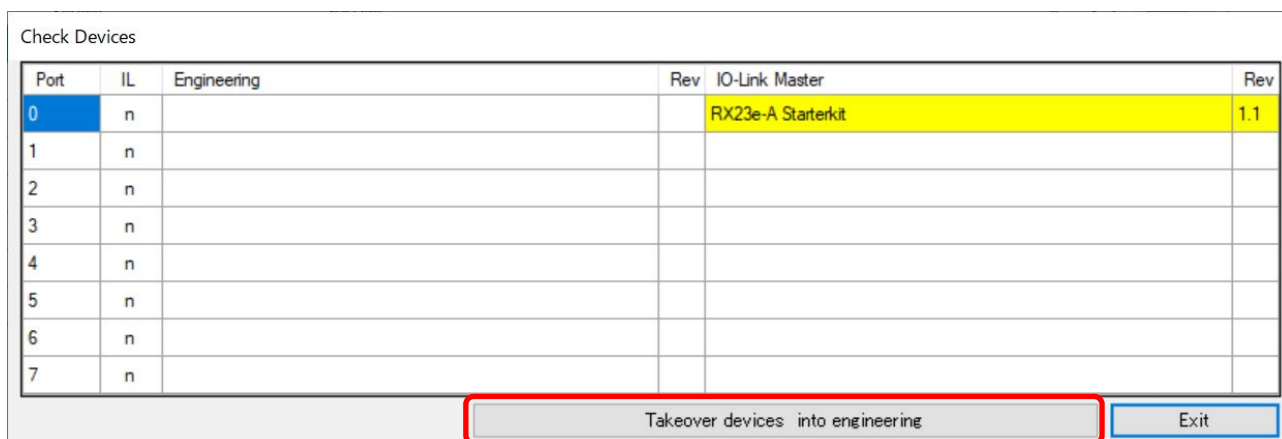


図 5-9 IO-Link Device Tool V5.1 – PE (IO-Link デバイスの検出)

- 「Takeover devices to engineering」ボタンをクリックします。マスターとデバイス間の接続が成功すると、IO-Link Master のポートに接続されているデバイスの表示が更新されます。

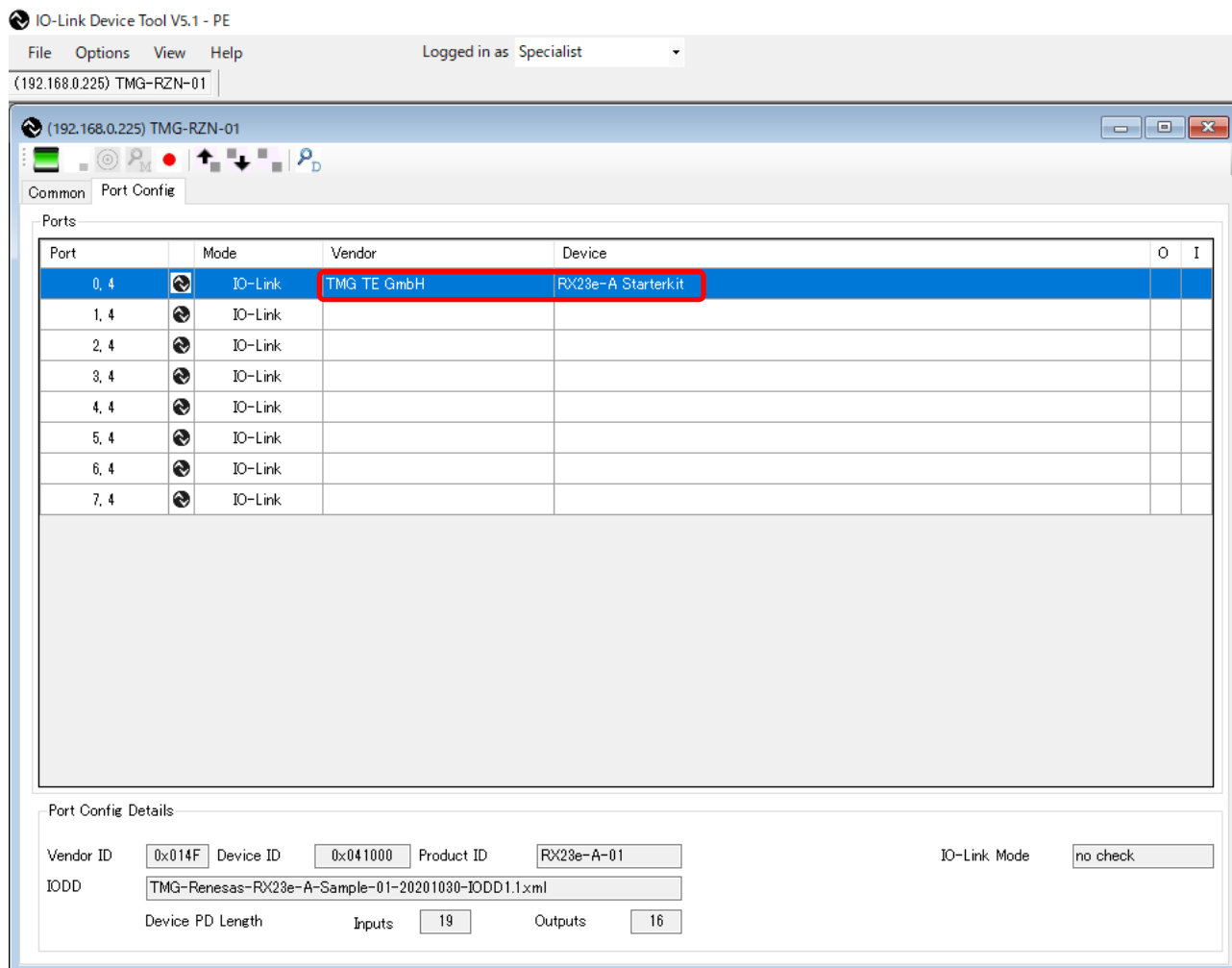


図 5-10 IO-Link Device Tool V5.1 – PE (IO-Link マスターとデバイスの接続成功)

6. 「RX23e-A Starterkit」をダブルクリックして、センサーの説明ページにアクセスします。

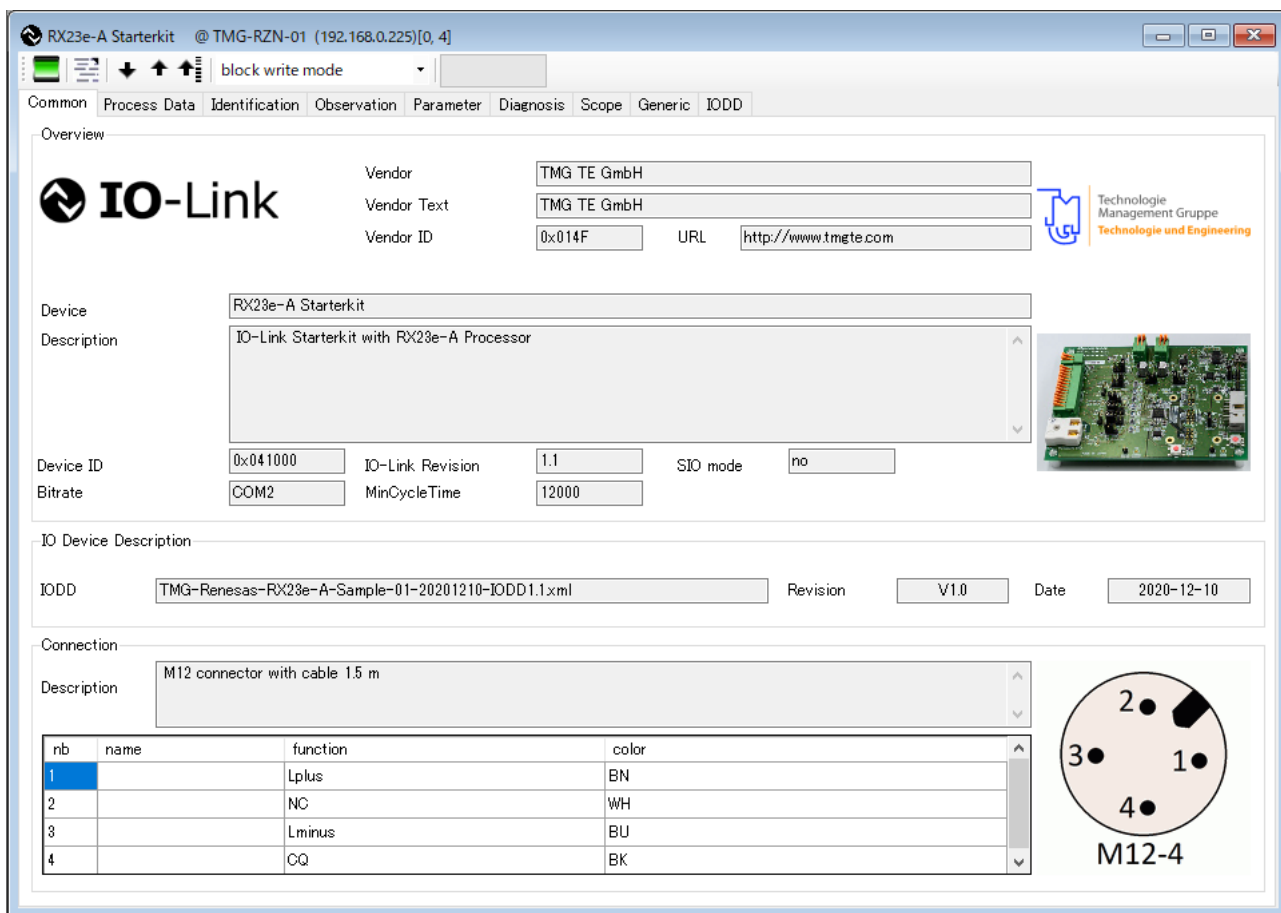


図 5-11 IO-Link Device Tool V5.1 – PE (RX23e-A Starterkit Common タブ)

## 5.7 IO-Link デバイスツールの RSSKRX23E-A ボードセンサーデモ

IO-Link デバイスツール GUI から、RSSKRX23E-A ボードを使用した IO-Link デバイス（温度センサー）を操作する方法を確認します。図 5-12 のボタンは IODD ウィンドウの左上にあります。

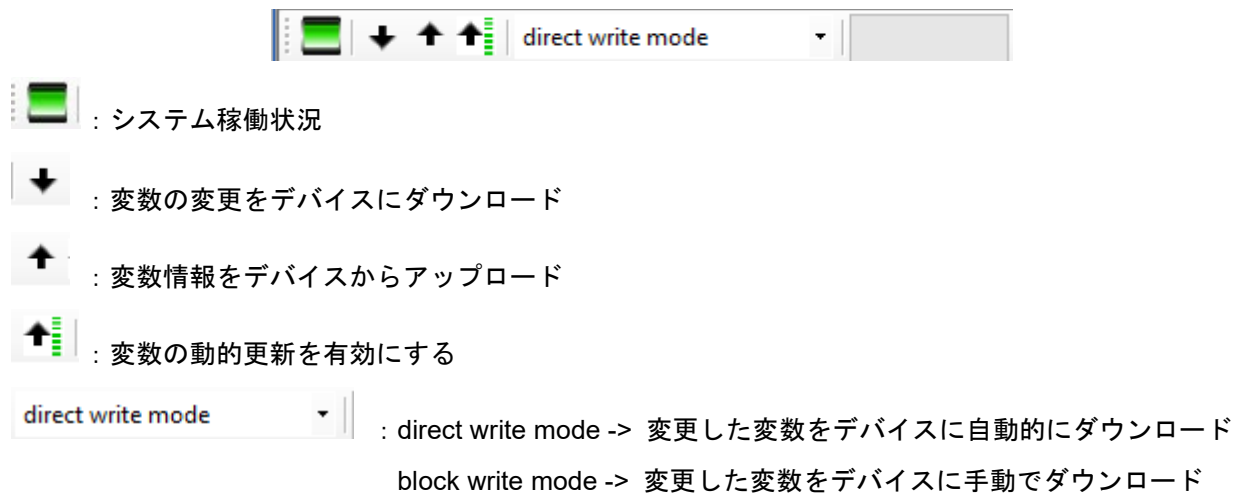


図 5-12 IO-Link Device Tool V5.1 – PE（デバイス操作ツールバー）

## 5.7.1 Common タブ

図 5-11 に示すように、Common タブには以下のようなデバイスに関する一般的な説明情報が表示されません。

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 : RX23e-A Starterkit

デバイスの説明 : IO-Link Starterkit with RX23e-A Processor

デバイス ID : 0x00041000

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 : 5000[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

## 5.7.2 Process Data タブ

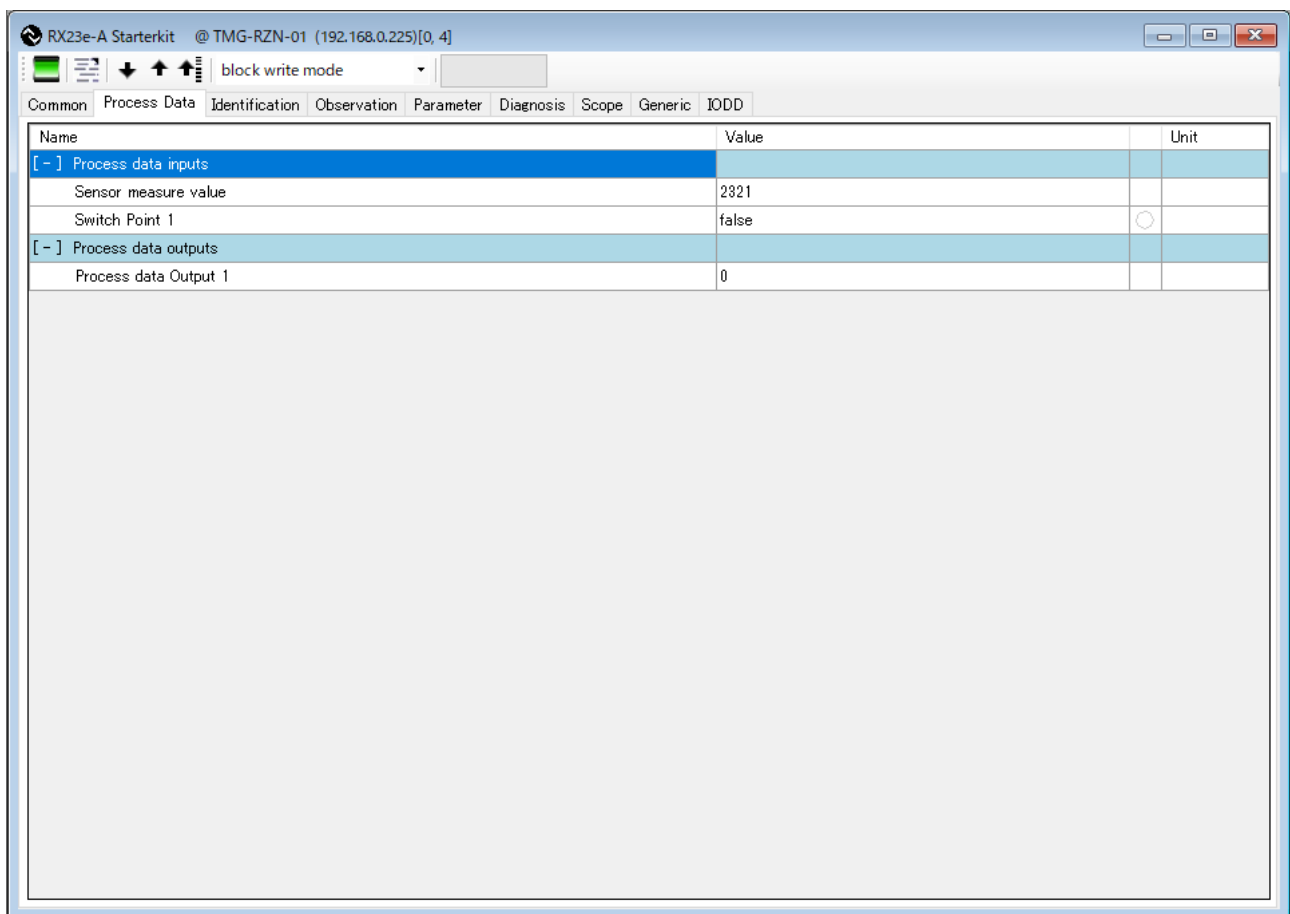
Process Data タブには、Process data inputs と Process data outputs が表示されます。

## 1. Process data inputs

- Sensor measure value  
計測温度[0.01°C]
- Switch Point 1  
スイッチング状態 (true or false)

## 2. Process data outputs

非対応です。



The screenshot shows the 'Process Data' tab in the IO-Link Device Tool V5.1 - PE. The window title is 'RX23e-A Starterkit @ TMG-RZN-01 (192.168.0.225)[0, 4]'. The 'block write mode' is selected. The 'Process Data' tab is active, and the table below displays the following data:

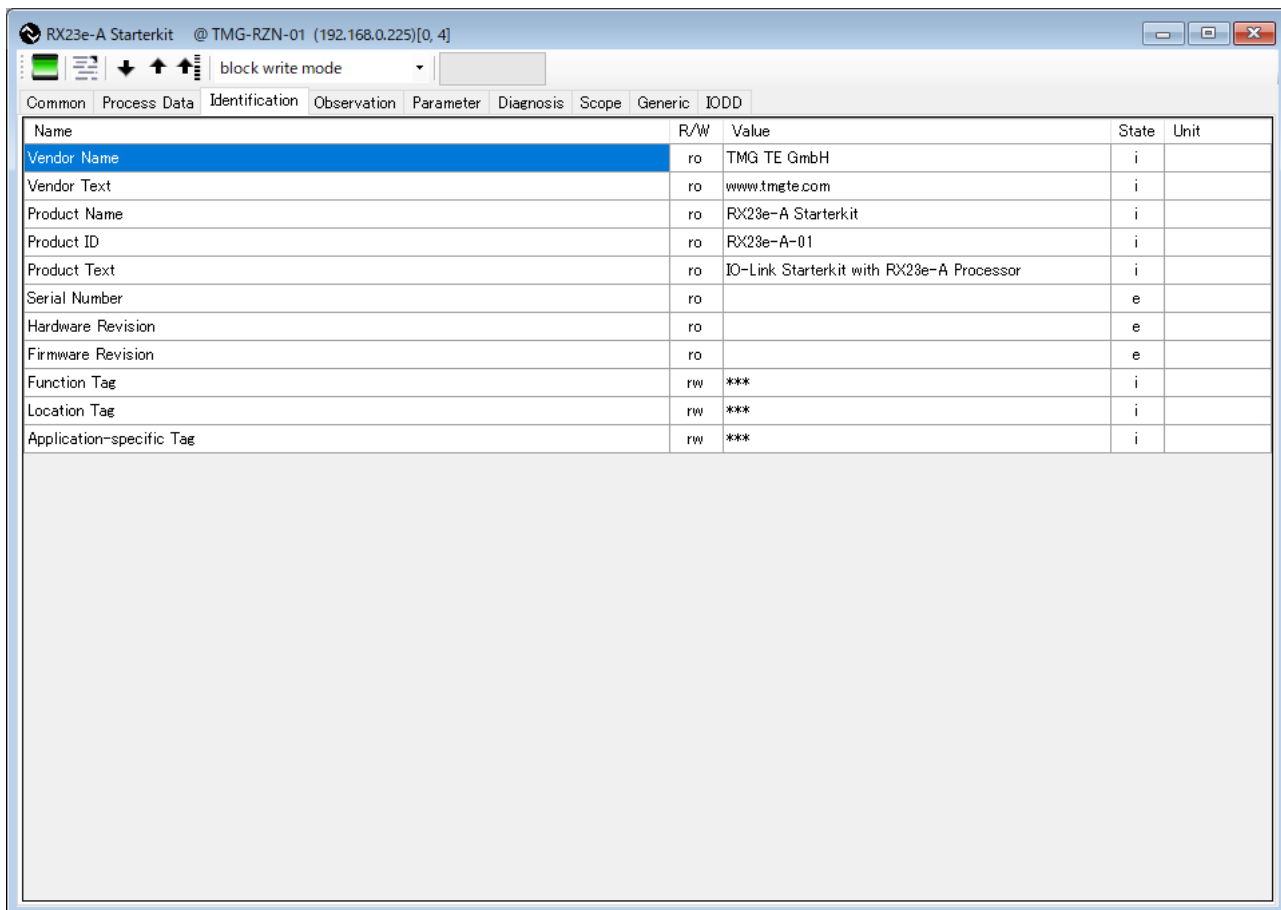
Name	Value	Unit
[ - ] Process data inputs		
Sensor measure value	2321	
Switch Point 1	false	<input type="radio"/>
[ - ] Process data outputs		
Process data Output 1	0	

図 5-13 IO-Link Device Tool V5.1 – PE (Process Data タブ)

## 5.7.3 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表示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 'RX23e-A Starterkit @ TMG-RZN-01 (192.168.0.225)[0, 4]'. The interface includes a toolbar with 'block write mode' and a menu bar with 'Common', 'Process Data', 'Identification', 'Observation', 'Parameter', 'Diagnosis', 'Scope', 'Generic', and 'IODD'. The main area displays a table with the following data:

Name	R/W	Value	State	Unit
Vendor Name	ro	TMG TE GmbH	i	
Vendor Text	ro	www.tmgte.com	i	
Product Name	ro	RX23e-A Starterkit	i	
Product ID	ro	RX23e-A-01	i	
Product Text	ro	IO-Link Starterkit with RX23e-A Processor	i	
Serial Number	ro		e	
Hardware Revision	ro		e	
Firmware Revision	ro		e	
Function Tag	rw	***	i	
Location Tag	rw	***	i	
Application-specific Tag	rw	***	i	

図 5-14 IO-Link Device Tool V5.1 – PE (Identification タブ)



## 5.7.4 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温度の情報を見ることができます。

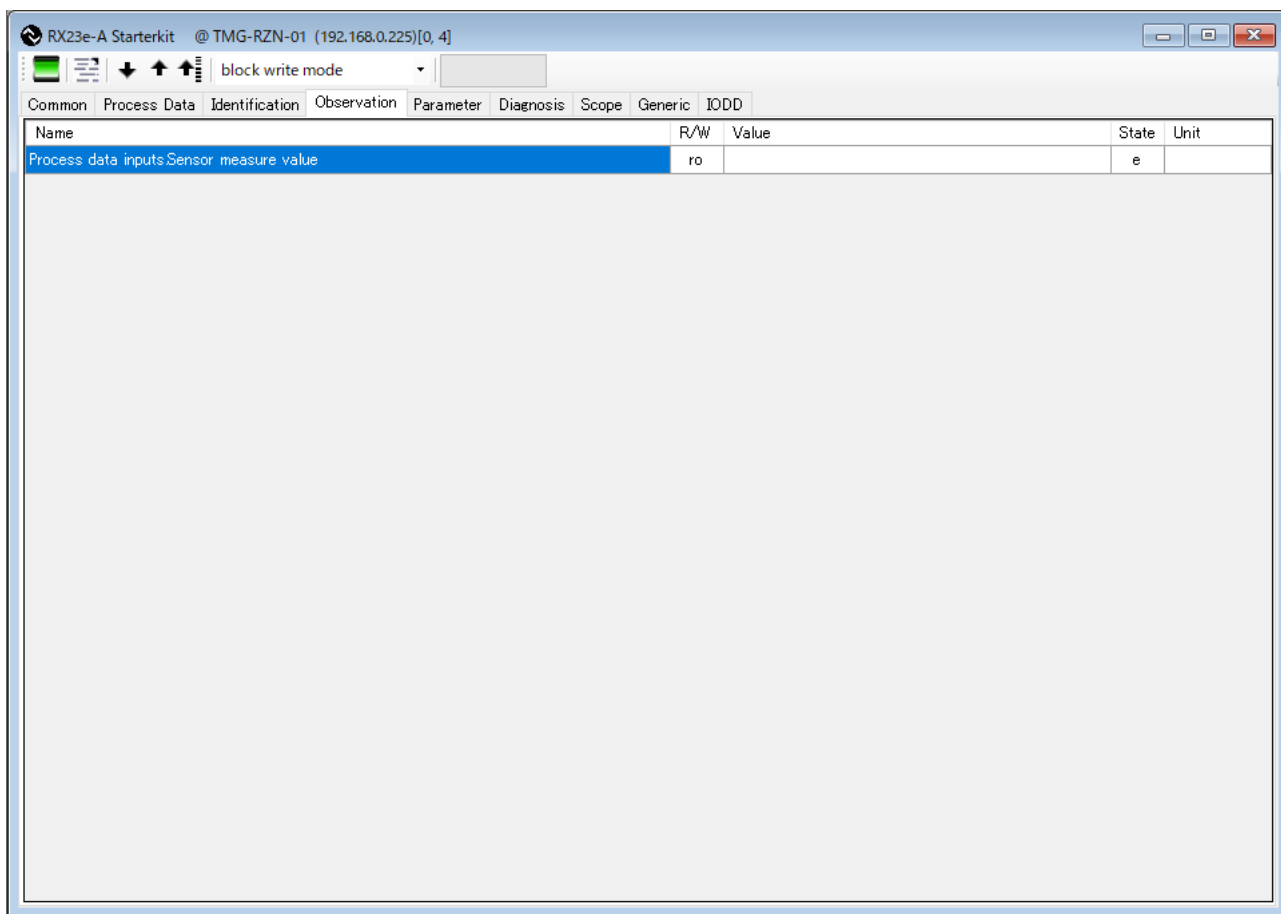


図 5-15 IO-Link Device Tool V5.1 – PE (Observation タブ)

## 5.7.5 Parameter タブ

Parameter タブには、デバイスのパラメータ設定が表示され、ユーザーはデバイスから設定値を読み取ることができます。また、デバイスに新しい設定を書き込むことができます。パラメータの意味、機能については表 4-3 と表 4-4 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「5.7.7 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
[-] Switching Signal Channel 1				
SSC1 Param.SP1	rw	25.00	i	° C
SSC1 Param.SP2	rw	20.00	i	° C
SSC1 Config.Logic	rw	High active - Not Inverted	i	
SSC1 Config.Mode	rw	Two point	i	
SSC1 Config.Hysteresis	rw	Standard	i	
[-] Teach-In Single Value				
System Command	wo	Teach SP1		
System Command	wo	Teach SP2		
TeachIn - ResultState	ro	Idle	i	

図 5-16 IO-Link Device Tool V5.1 – PE (Parameter タブ)

## 5.7.6 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。

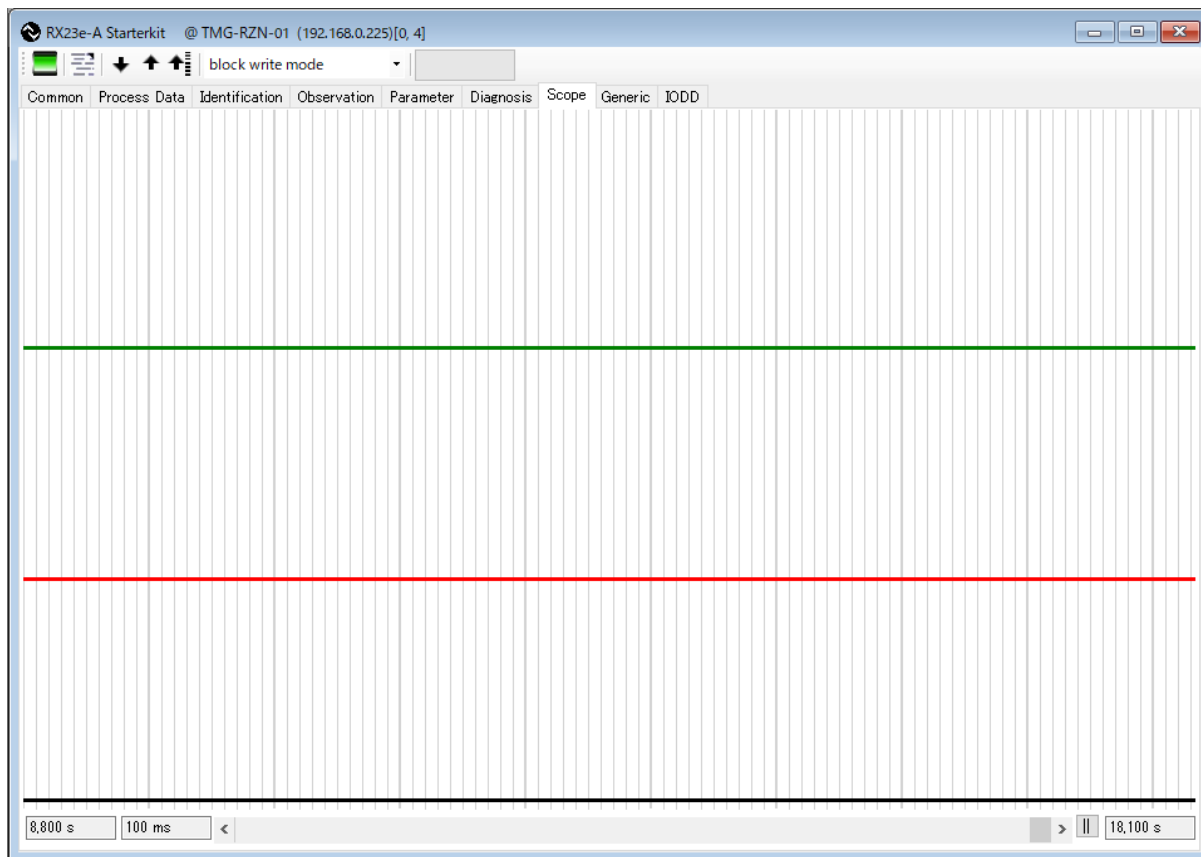


図 5-17 IO-Link Device Tool V5.1 – PE (Scope タブ)

ユーザーは Scope エリアで右クリックすることで、Scope 構成の設定を行うことができます。以下のウィンドウが表示され、設定を編集することができます。

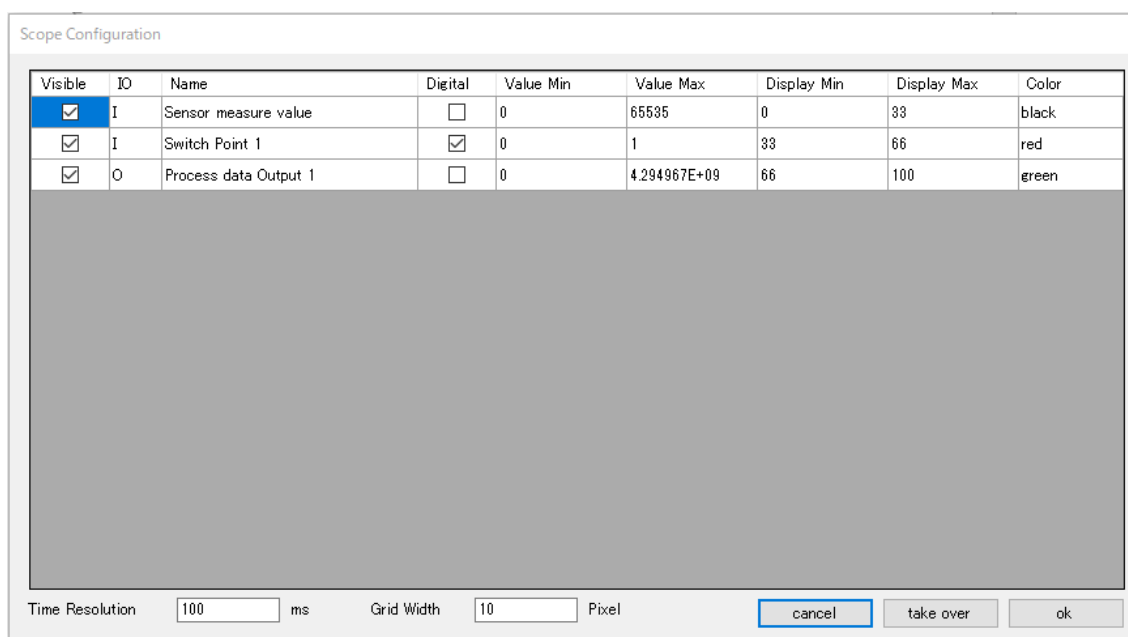


図 5-18 IO-Link Device Tool V5.1 – PE (Scope タブ、Scale/Parameter 設定)

### 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 列のデフォルト値に設定されます。これらの値は IODD ファイルに記録されています。詳細については図 5-16 を参照してください。

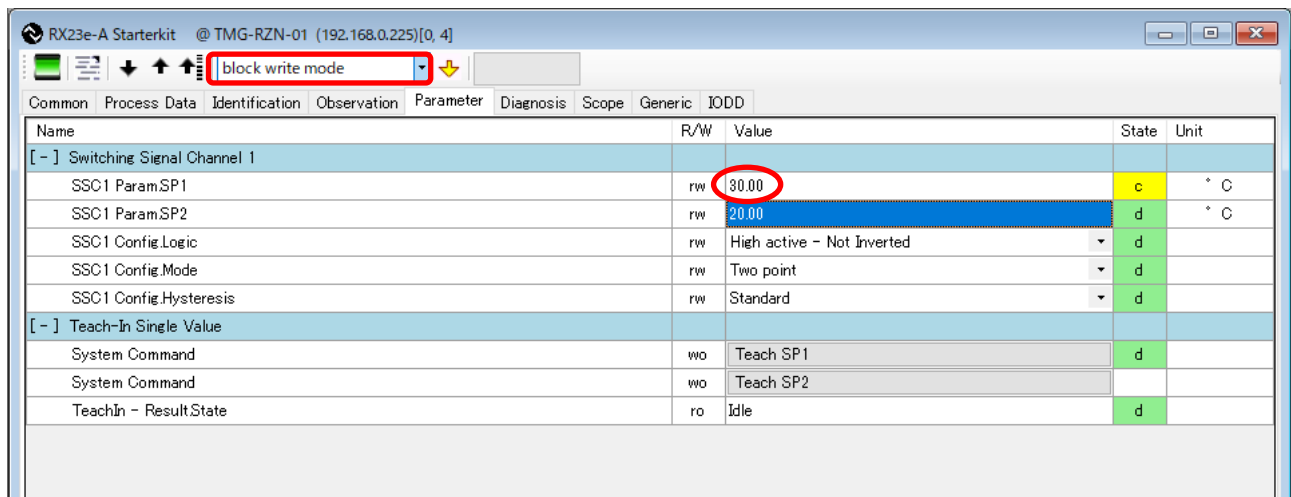
SetPoint (SP1, SP2 など) をデバイスに設定するときは、「Device Access Locks」が false に設定されていることを確認してください。true の場合には、デバイスへのパラメータ設定が失敗します。SetPoint パラメータは「Teach Values」パラメータを使用して設定するか、「Standard Command」を使用して設定することができます。

ユーザーは IO-Link デバイスツールにより、書き込みモードを選択してパラメータの設定を行うことができます。

#### 5.7.6.1 Teach Values (block write mode) でのパラメータ設定

このモードでは、パラメータの変更を行ってから「変更した全ての値をデバイスに書き込む」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「SSC1 Param SP1」または「SSC1 Param SP2」の「Value」フィールドをクリックします。
3. 数値を入力し、Enter を押下します。「Status」の黄色の網掛けは、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. 図 5-19 に示すように、パラメータを変更する度に「変更した全ての値をデバイスに書き込む」ボタンをクリックする必要があります。
5. 「Status」の緑色の網掛けは、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行われたことを示します (図 5-20 を参照してください)。



Name	R/W	Value	State	Unit
[-] Switching Signal Channel 1				
SSC1 Param.SP1	rw	30.00	c	° C
SSC1 Param.SP2	rw	20.00	d	° C
SSC1 Config.Logic	rw	High active - Not Inverted	d	
SSC1 Config.Mode	rw	Two point	d	
SSC1 Config.Hysteresis	rw	Standard	d	
[-] Teach-In Single Value				
System Command	wo	Teach SP1	d	
System Command	wo	Teach SP2		
TeachIn - ResultState	ro	Idle	d	

図 5-19 IO-Link Device Tool V5.1 – PE (block write mode でのパラメータ設定)

## 5.7.6.2 Teach Values (direct write mode) でのパラメータ設定

このモードでは、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「SSC1 Param SP1」または「SSC1 Param SP2」の「Value」フィールドをクリックします。
3. 数値を入力し、Enter 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。

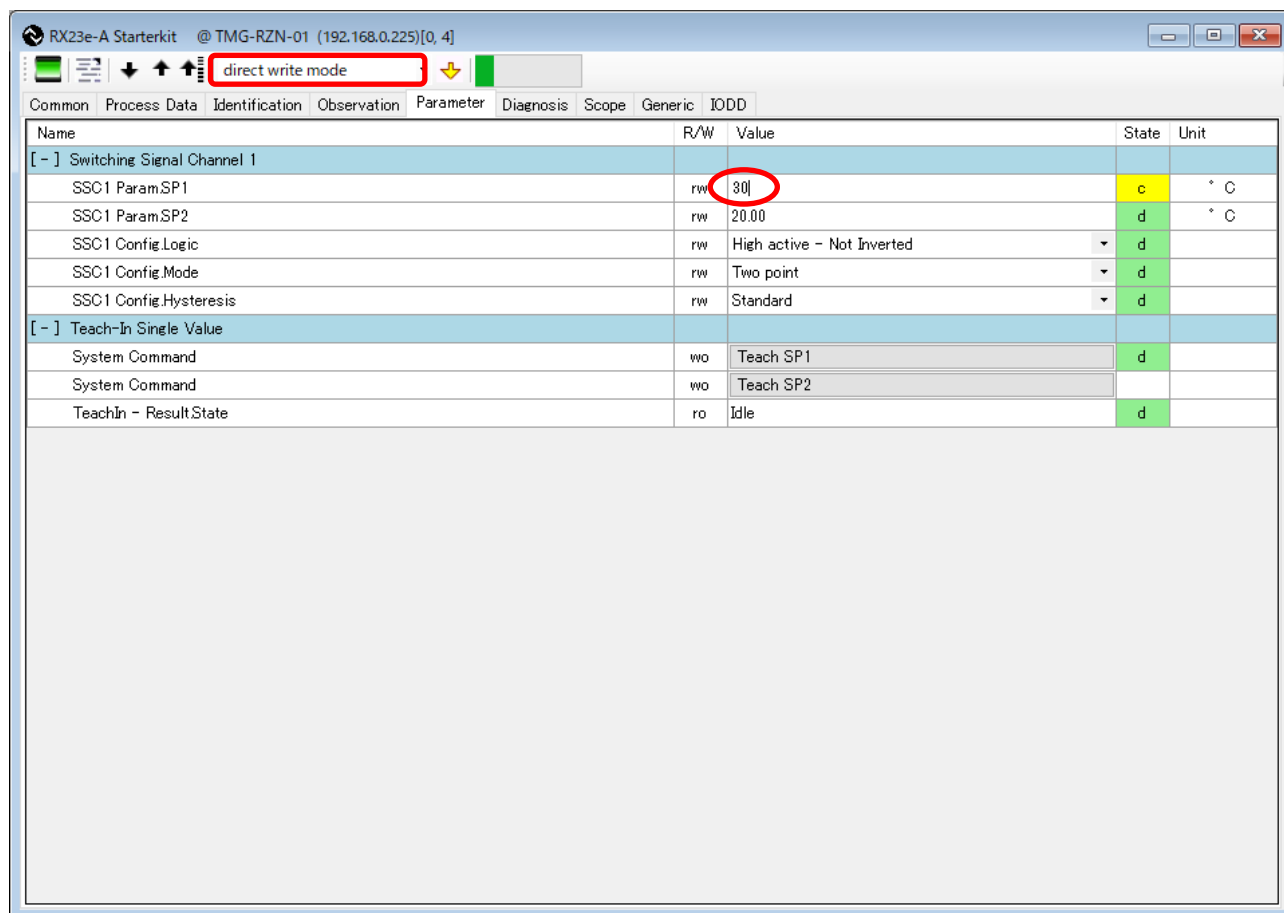


図 5-20 IO-Link Device Tool V5.1 – PE (direct write mode でのパラメータ設定)

## 5.7.6.3 Standard Command でのパラメータ設定

Standard Command を使用して SetPoint を設定します。

1. デバイスの測定対象の温度を、SetPoint として設定したい温度にします。
2. 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。

計測された温度が「SSC1 Param SP1」または「SSC1 Param SP2」に自動的に設定され、Teach-In の結果も「SP1 Success」または「SP2 Success」となります。

The screenshot shows the IO-Link Device Tool V5.1 - PE interface. The 'Parameter' tab is active, displaying a table of parameters. The 'Value' column for 'SSC1 Param.SP1' is circled in red and contains the value '23.27'. A red arrow points from the text 'Press Teach SP1' to the 'Teach SP1' button in the 'Teach-In Single Value' section. Another red arrow points from the text 'Successful Teach-in' to the 'SP1 Success' result state in the 'TeachIn - ResultState' section.

Name	R/W	Value	State	Unit
[-] Switching Signal Channel 1				
SSC1 Param.SP1	rw	23.27	d	° C
SSC1 Param.SP2	rw	20.00	d	° C
SSC1 Config.Logic	rw	High active - Not Inverted	d	
SSC1 Config.Mode	rw	Two point	d	
SSC1 Config.Hysteresis	rw	Standard	d	
[-] Teach-In Single Value				
System Command	wo	Teach SP1	d	
System Command	wo	Teach SP2		
TeachIn - ResultState	ro	SP1 Success	d	

図 5-21 IO-Link Device Tool V5.1 - PE (Standard Command でのパラメータ設定)

#### 5.7.6.4 パラメータの読み取り

ユーザーは「変数情報をデバイスからアップロード」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。「変数情報をデバイスからアップロード」ボタンについては図 5-12 を参照してください。

#### 5.7.7 工場出荷設定への復元

デバイスの設定を工場出荷時の設定に戻します。

1. 「Diagnosis」タブに移動します。
2. 「Restore Factory Setting」ボタンをクリックします。

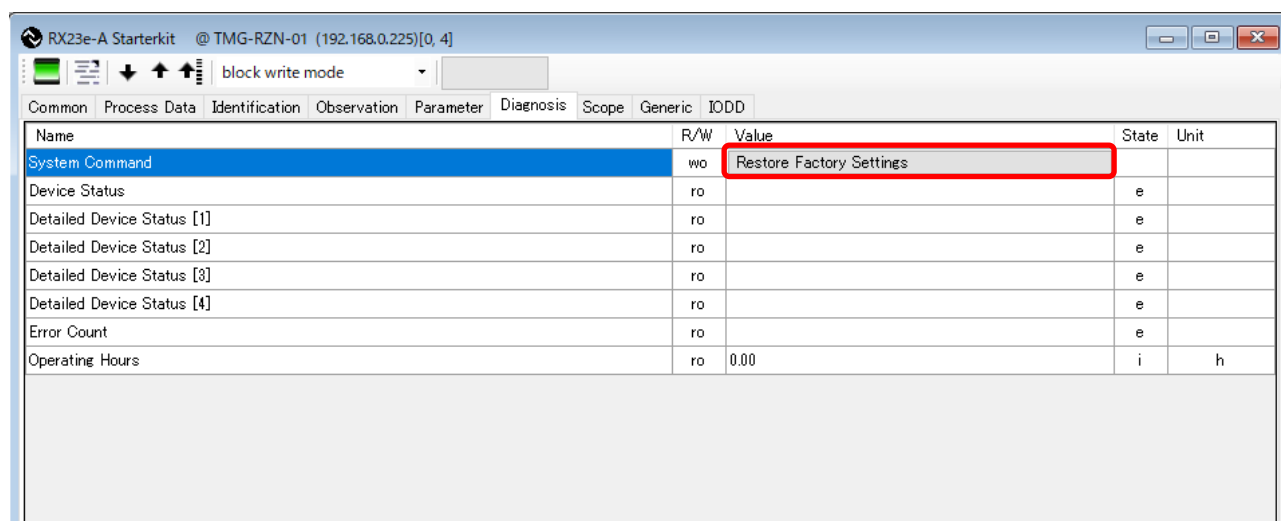


図 5-22 IO-Link Device Tool V5.1 – PE（工場出荷設定への復元）

#### 5.7.8 Generic タブ

このタブは、IODD ファイルなしで IO-Link デバイスを操作する際に使用します。データは RAW データとして表示され、アドレスはインデックスとサブインデックスを介して設定されます。IO-Link ポートのメッセージボックスには、手動で処理された読み取り／書き込み要求の結果が表示されます。

#### 5.7.9 IODD タブ

このタブには IODD 情報が表示されます。

## 改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2021.1.15	-	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（豊洲フォレストシア）

[www.renesas.com](http://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/](http://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。