

# RA2E2 グループ

## IO-Link ソリューション例

### 要旨

本アプリケーションノートでは、RA2E2 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 は IEC61131-9 に準拠したセンサ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

RA2E2、RH4Z2501(IO-Link Line Driver)

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	3
2. 関連ドキュメント	4
3. 動作確認環境	5
4. ハードウェア構成	5
4.1 IO-Link USB マスター	6
4.2 RA2E2 IO-Link Inductive Sensor Board	6
4.3 RA2E2 MCU の周辺モジュール	7
4.4 外部モジュール	8
4.4.1 コンパレータ	8
5. サンプルアプリケーション概要	9
5.1 ソフトウェアコンポーネント	10
5.1.1 Application	10
5.1.2 IO-Link Stack Library	10
5.1.3 FSP	10
5.2 センシングメカニズム	11
5.2.1 動作原理	11
5.3 全体処理フロー概要	13
5.4 スイッチング状態と LED 動作イメージ	15
5.5 IO-Link 通信仕様	16
5.5.1 ビットレート	16
5.5.2 SIO モード	16
5.5.3 プロセスデータ(PDIn)	16
5.5.4 パラメータ	16
5.6 使用する端子	17
5.7 プログラム構成	18
5.7.1 ファイル構成	18

5.7.2	関数一覧	20
5.7.2.1	hal_entry.c	20
5.7.2.2	START.c	20
5.7.2.3	EXCIT.c	21
5.7.2.4	MEASMNG.c	21
5.7.2.5	LED.c	21
5.7.2.6	main.c	22
5.7.2.7	IOLinkMain.c	22
6.	サンプルプロジェクト実行方法	23
6.1	サンプルプロジェクトのインポート	23
6.2	プロジェクトのビルド	25
6.3	デバッグ接続のセットアップ	27
6.4	MCU へのサンプルプロジェクトの書き込み	31
6.5	プログラムの開始	33
7.	PACTware DC の使用準備及び機能解説	34
7.1	.NET Framework 有効化	34
7.2	IO-Link Parameterization Tool インストール	38
7.3	IODD セットアップ	40
7.4	PACTware DC 操作方法	44
7.5	PACTware DC 機能説明	46
7.5.1	Identification	46
7.5.2	Parameter	47
7.5.2.1	パラメータをデフォルト値にリセット	47
7.5.2.2	Config.Mode 設定	49
7.5.2.3	Config.Logic 設定	50
7.5.2.4	Config.LED 設定	51
7.5.3	Process data	52
8.	使用例	53
8.1	default 設定での動作	54
8.2	設定変更での動作	55
	改訂記録	56

## 1. 概要

本例では RA2E2 IO-Link Inductive Sensor Solution Board(以降 Inductive Sensor Board)を IO-Link Device とし、市販の IO-Link USB Master を IO-Link Master とします。ホスト PC と IO-Link Master との通信は USB マスターメーカーが提供する IO-Link GUI ツールを使用します。IO-Link GUI ツールは Windows PC 上で動作するアプリケーションソフトウェアです。

Inductive Sensor Board では、ボード先端に搭載されている Sensing Coil で金属の接近を検出することができます。金属接近を検出すると Inductive Sensor Board 上の LED が点灯します。

また Inductive Sensor Board からは金属検出状況を IO-Link 通信により USB Master に通知します。ホスト PC の IO-Link GUI ツールでは検出状況を確認することができます。また IO-Link GUI ツールからは Inductive Sensor Board のモード変更などを操作することができます。

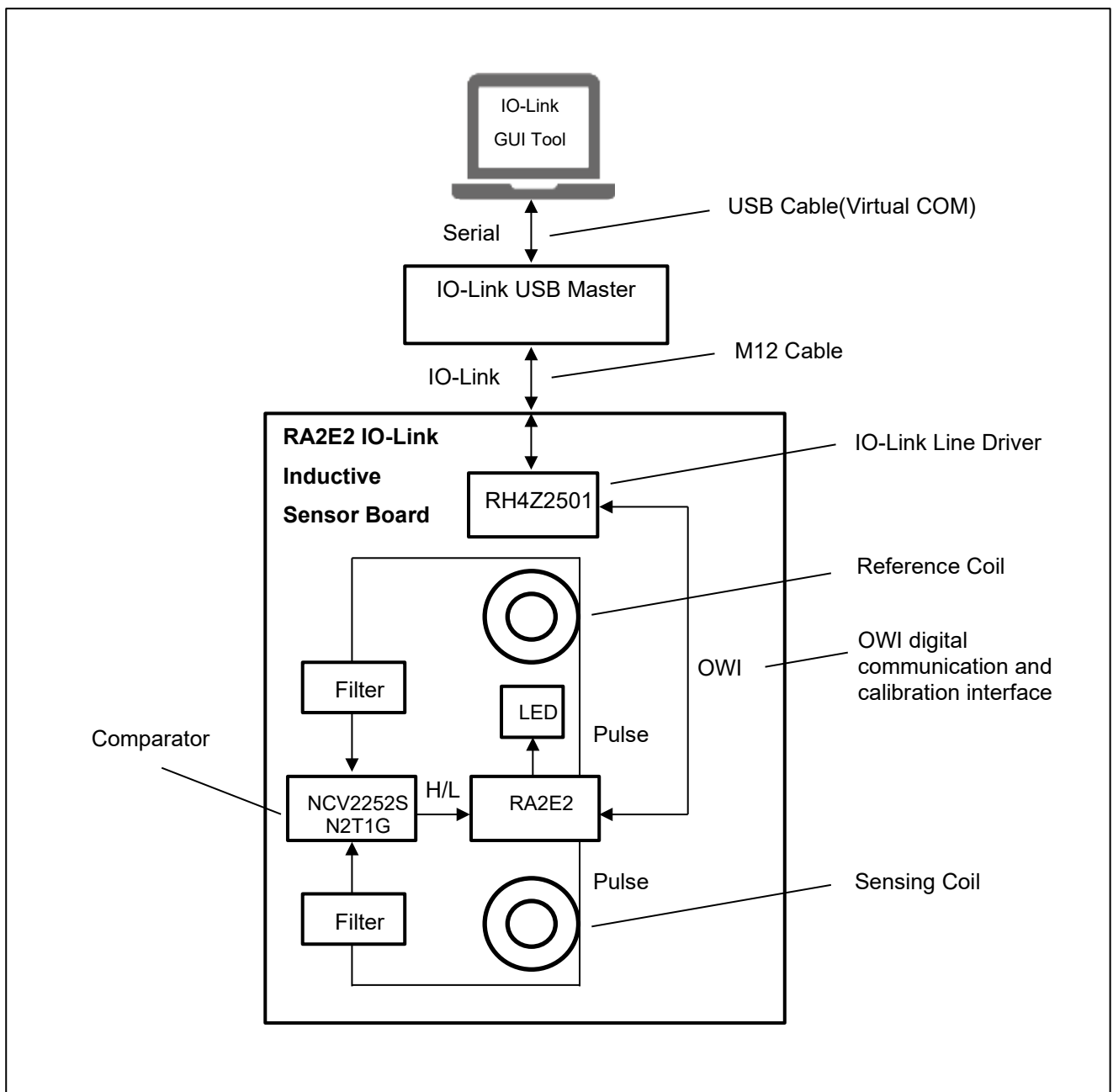


図 1-1 RA2E2 を用いた IO-Link Inductive Sensor システム

## 2. 関連ドキュメント

- R01UH0919JJ0120 Renesas RA2E2 グループユーザーズマニュアル ハードウェア編
- R01DS0387JJ0120 RA2E2 グループ Datasheet
- R11UM0155EU0310 Renesas Flexible Software Package (FSP) v4.6.0 User's Manual
- R20AN0657EU0101 Rev.1.01 EK-RA2E2 Example Project Bundle
- R20UT4686JJ0300 Rev.3.00 E2 エミュレータ, E2 エミュレータ Lite ユーザーズマニュアル別冊 (RA 接続時の注意事項)
- IO-Link Offline Parameterization Installation/User Instructions

### 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
Sensor Board	<a href="#">TS-IO-RA2E2-02</a> (RA2E2 IO-Link Inductive Sensor Solution Board)
MCU	R7FA2E2A72DNK(RA2E2 24pin)
IO-Link Line Driver	RH4Z2501
コンパレータ	NCV2252SN2T1G
IDE	Renesas e <sup>2</sup> studio 2023-07 (23.7.0)
Tool Chain	GCC ARM Embedded 12.2.1.20230214
FSP	v4.6.0
ライブラリ	TMG 社製 IO-Link スタック
エミュレータ	Renesas E2 Lite エミュレータ 用ユーザーインターフェースケーブル (20-10 pins) [RTE0T00020KCAC1000J]
IO-Link Master	Pepperl + Fuchs IO-Link-Master02-USB デバイスタイプマネージャ <sup>注</sup>
IO-Link GUI Tool	<a href="#">IO-Link Offline Parameterization Tool V1.00.006 / 2022-07-10</a>
IO-Link GUI Tool 用ホスト PC	Windows10 Professional

【注】 マスター検索されない場合はデバイスタイプマネージャを再インストールしてください。

### 4. ハードウェア構成

ハードウェアは IO-Link Master と IO-Link Device で構成されます。IO-Link Master は Pepperl + Fuchs IO-Link-Master02-USB、IO-Link Device は RA2E2 IO-Link Inductive Sensor Board で構成されます。

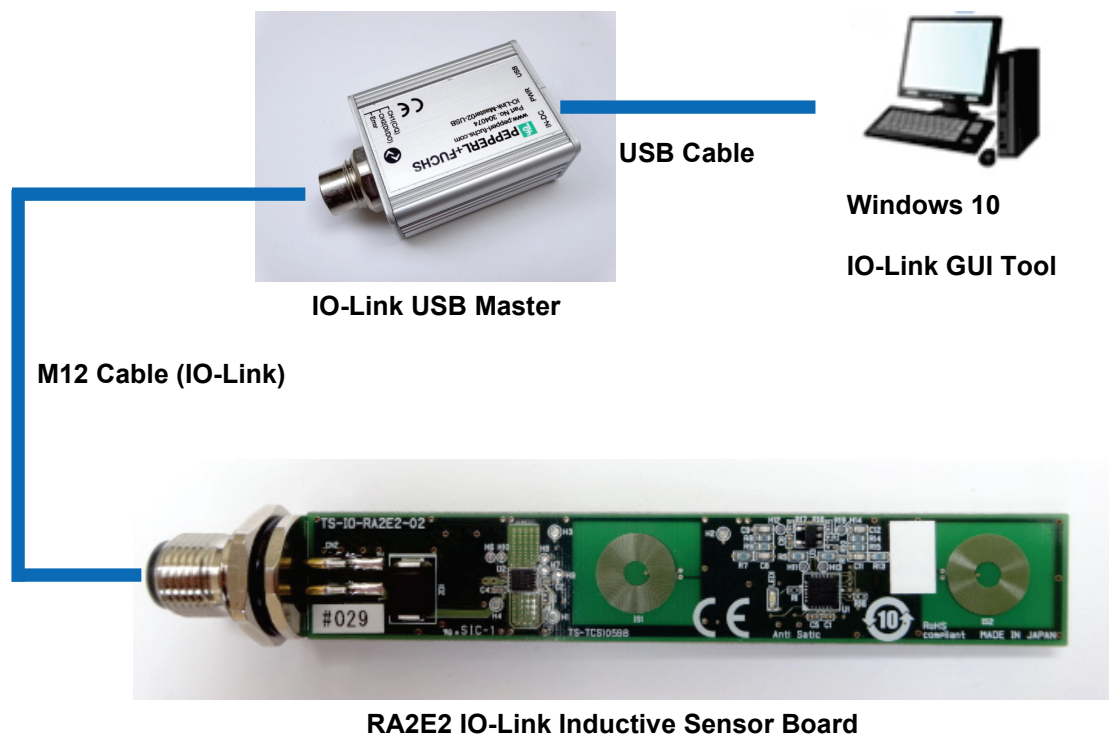


図 4-1 ハードウェア構成

### 4.1 IO-Link USB マスター

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



図 4-2 IO-Link USB Master

### 4.2 RA2E2 IO-Link Inductive Sensor Board

IO-Link Device として RA2E2 IO-Link Inductive Sensor Board を使用します。

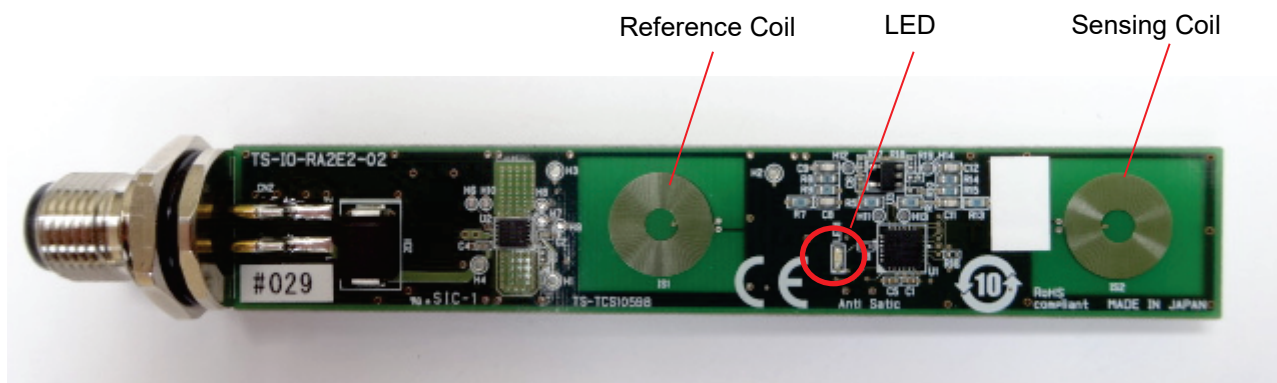


図 4-3 RA2E2 IO-Link Inductive Sensor Board

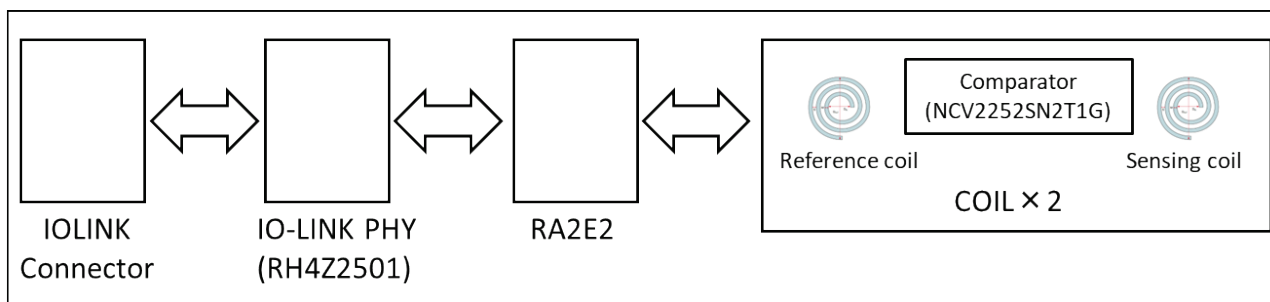


図 4-4 RA2E2 IO-Link Inductive Sensor Board ハードウェア構成

### 4.3 RA2E2 MCU の周辺モジュール

表 4-1 に、このアプリケーションプロジェクトで使用する RA2E2 MCU の主な周辺モジュールとその用法を示します。これらは図 4-5 RA2E2 MCU 機能概要に示されています。

表 4-1 周辺モジュール

モジュール	使用法
AGTW1	100ms の周期でタイマ割込みを発生させます。 割込みにより GPT5/6/8 を同時に起動しセンシング処理を行います。
GPT5	センシング処理開始から 2500ns 後に、サンプリングの為の割込みを発生させます。
GPT6	センシング処理開始により、センシングコイル励起用のこぎり波を発生させます。
GPT8	センシング処理開始により、リファレンスコイル励起用のこぎり波を発生させます。
DTC	GPT5 の割込みにより起動され、外付けコンパレータ結果取得に使用されます。

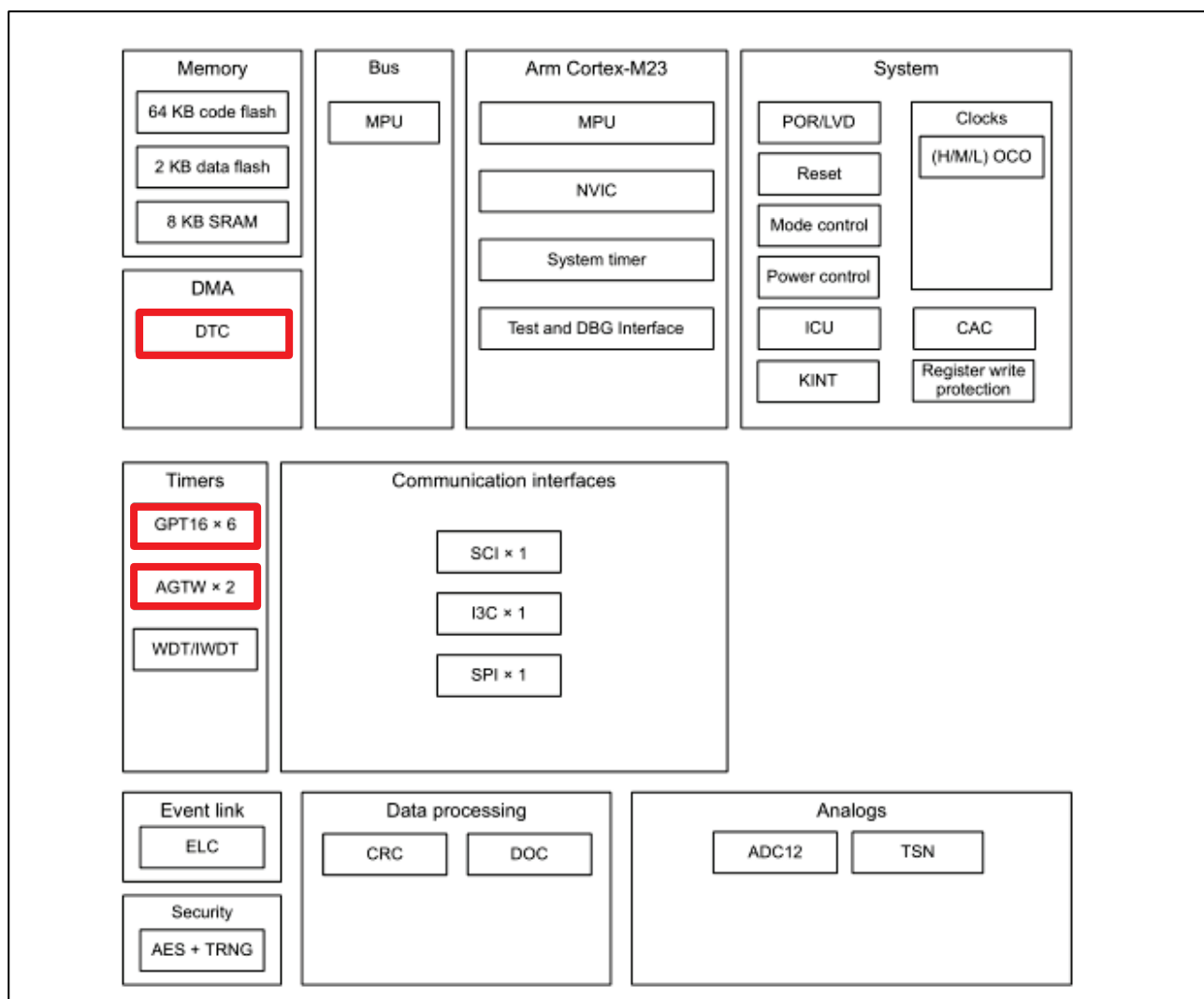


図 4-5 RA2E2 MCU 機能概要

## 4.4 外部モジュール

### 4.4.1 コンパレータ

ボードにはコンパレータ NCV2252SN2T1G が搭載されています。  
センシングコイルのフィルタリングされた出力とリファレンスコイルのフィルタリングされた出力とを比較し、センシングコイル上の金属の存在を判断するために使用されます。



## 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動作モードに関する情報を受け取ります。IO-Link デバイスはセンシング処理を定期的に（100ms に 1 回）実行し、IO-Link 通信で IO-Link マスターにセンシング情報（プロセスデータ）を送信します。

センシング処理では 3 つの GPT タイマを同時に起動します。2 つの GPT タイマはリファレンスコイルとセンシングコイルのパルス駆動を行います。3 つ目の GPT タイマがフィルターからの出力を比較する測定ポイント(2500ns)でタイムアウトし DTC でコンパレータ結果を取得します。タイムアウトによって起動された割込み処理でコンパレータ結果に基づき LED を点灯します。

スマートセンサ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。

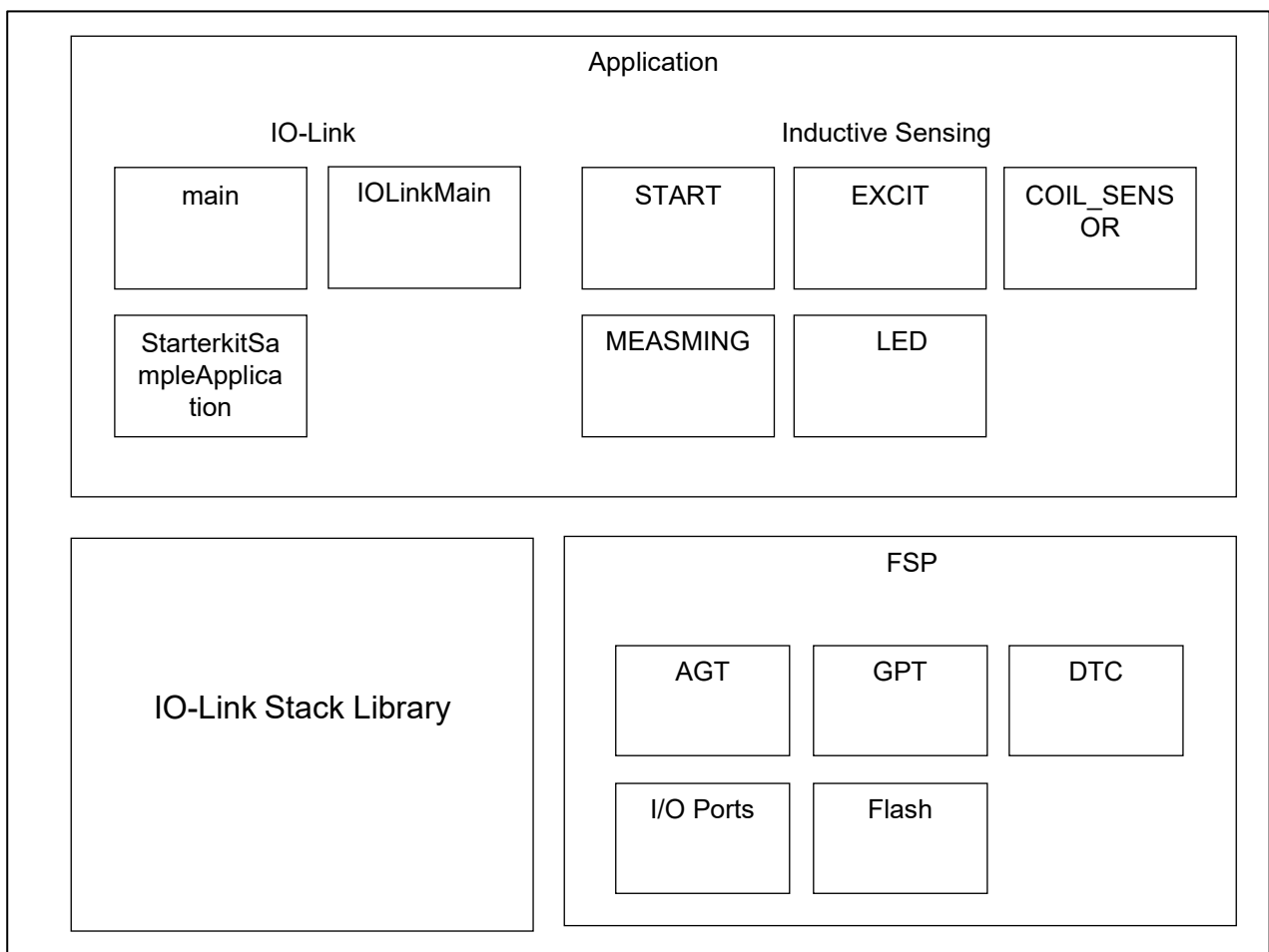


図 5-1 ソフトウェアコンポーネント

## 5.1 ソフトウェアコンポーネント

### 5.1.1 Application

#### IO-Link

main -このモジュールは、IO-Link の初期化と main-loop を担当します。

IOLinkMain -このモジュールは、定期的な IO-Link スタックの呼び出しと、ステータスの取得を担当します。

StarterkitSampleApplication -このモジュールは、TMG IO-Link スタックでのプロセス データとイベントに関する処理を担当します。

#### Inductive Sensing

START -このモジュールは、システム変数の初期化と周期タイマの起動を担当します。

COILSENSOR -このモジュールは、オンボードのコイルの処理を担当します。コイルをパルス駆動し測定を行います。センシングコイル上に金属を検出した際はボード上の LED が点灯します。

EXCIT -このモジュールには、タイマ、コンパレータ、および DTC の初期化が含まれています。また、周期的な測定トリガーマカニズムが含まれています。

MEASMNG -このモジュールは、測定を管理し、測定サイクルの次の励起を開始する役割を果たします。

LED -このモジュールは、LED 制御を担当します。点灯、点滅、消灯、反転の機能を提供します。

### 5.1.2 IO-Link Stack Library

このライブラリモジュールは、IO-Link に関する処理を担当します。

### 5.1.3 FSP

システムの MCU ハードウェアコンポーネントをアプリケーション部分と接続する役割を果たします。このレイヤーの主なコンテンツは MCU の周辺機能ドライバであり、e<sup>2</sup>studio IDE 環境上でスマートコンフィグレータを使用し、構成・関連ファイルを生成します。

## 5.2 センシングメカニズム

### 5.2.1 動作原理

動作原理は LC 回路の発振に基づいています。センシングコイル上の金属検出のためにリファレンスコイルが使用されます。リファレンスコイル上への金属配置は行いません。測定原理は LC 回路(リファレンスコイル LC 回路とセンシングコイル LC 回路)を励起することから始まります。

回路は、トランジスタドライバ回路の助けを借りて RA2E2 MCU によって生成されたパルスで励起されます。

コイルが励起された後それらは振動し始めます。次に振動がフィルタリングされ振動エンベロープのみが生成されます。プロセスの最後にボードに搭載されたコンパレータを使用してこれら 2 つのエンベロープを比較し、信号レベルの比較結果に基づいてセンシングコイル上で金属が検出されたかどうかを判断できます。

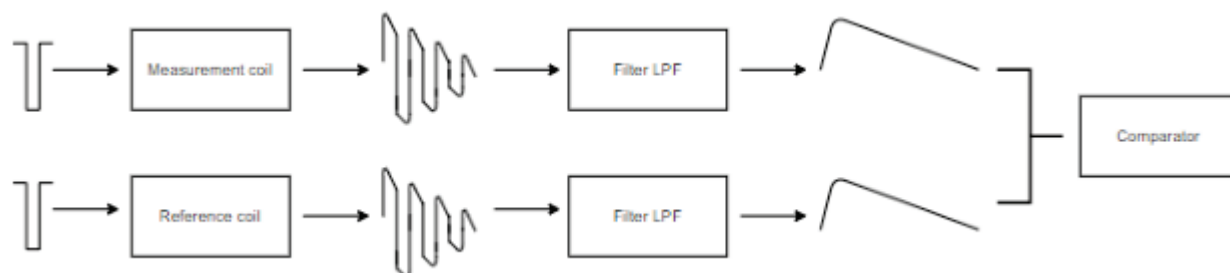


図 5-2 動作原理

タイマを使用したパルスの生成メカニズムについて説明します。

1 つはリファレンスコイル用、もう 1 つはセンシングコイル用の励起パルスを生成する為に 2 つのタイマ出力を使用します。測定ポイント用に出力無しのもう 1 つのタイマを使用して、フィルター出力を比較する測定ポイントを設定します。

パルス生成タイマはのこぎり波ワンショットパルスで構成する必要があり、測定ポイント用タイマはワンショットモードとして構成されます。のこぎり波ワンショットパルスモードは FSP のスマートコンフィグレータでは構成できないため周期モードを仮で選択し、ユーザアプリケーションコード内で、のこぎり波ワンショットパルスモードに書き換えます。図 5-3 はパルス生成メカニズムを示しています。

パルス生成タイマは、のこぎり波ワンショットパルスモードで構成され、パルス生成を開始および終了するカウントはコンペアキャプチャレジスタ GTCCRn で設定されます。タイマの周期は最大(0xFFFF)に設定されます。これは、これらのタイマが、測定ポイントタイマによって生成された割り込みで停止するためです。上記のすべてのタイマが同時に開始されます。パルスが生成された後、測定ポイントタイマがタイムアウトすると次のアクションが実行されます。

- DTC がトリガーされ、その時点でコンパレータの結果がハードウェアにより変数にコピーされます。
- 測定ポイントタイマ割り込みがトリガーされ、パルス生成タイマが停止する割り込みが発生します。

コンパレータの結果を用いたコイル上の金属部品の存在を判定する手順を以下に示します。

- 1つのコイルの測定が開始されます。
- そのコイルの測定ポイントタイマがタイムアウトするとコールバック関数が呼び出され、コンパレータの結果に基づいて測定データが処理されます。
- 実際の結果に基づいて、対応する表示 LED を制御します。

この測定サイクルは、AGTW1 タイマにより 100ms で周期的にトリガーされます。

検出しきい値は「Delay between pulses」で設定可能です。

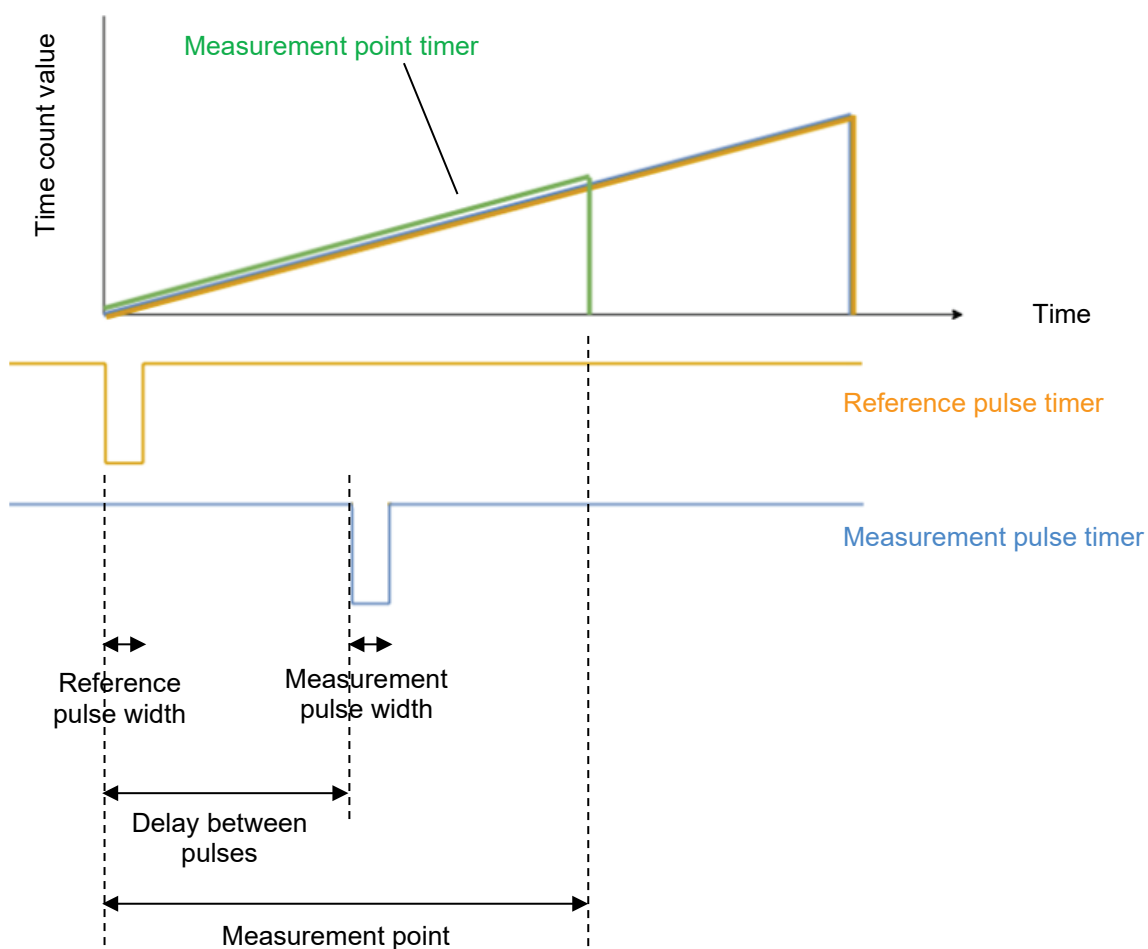


図 5-3 パルス発生メカニズム

5.3 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

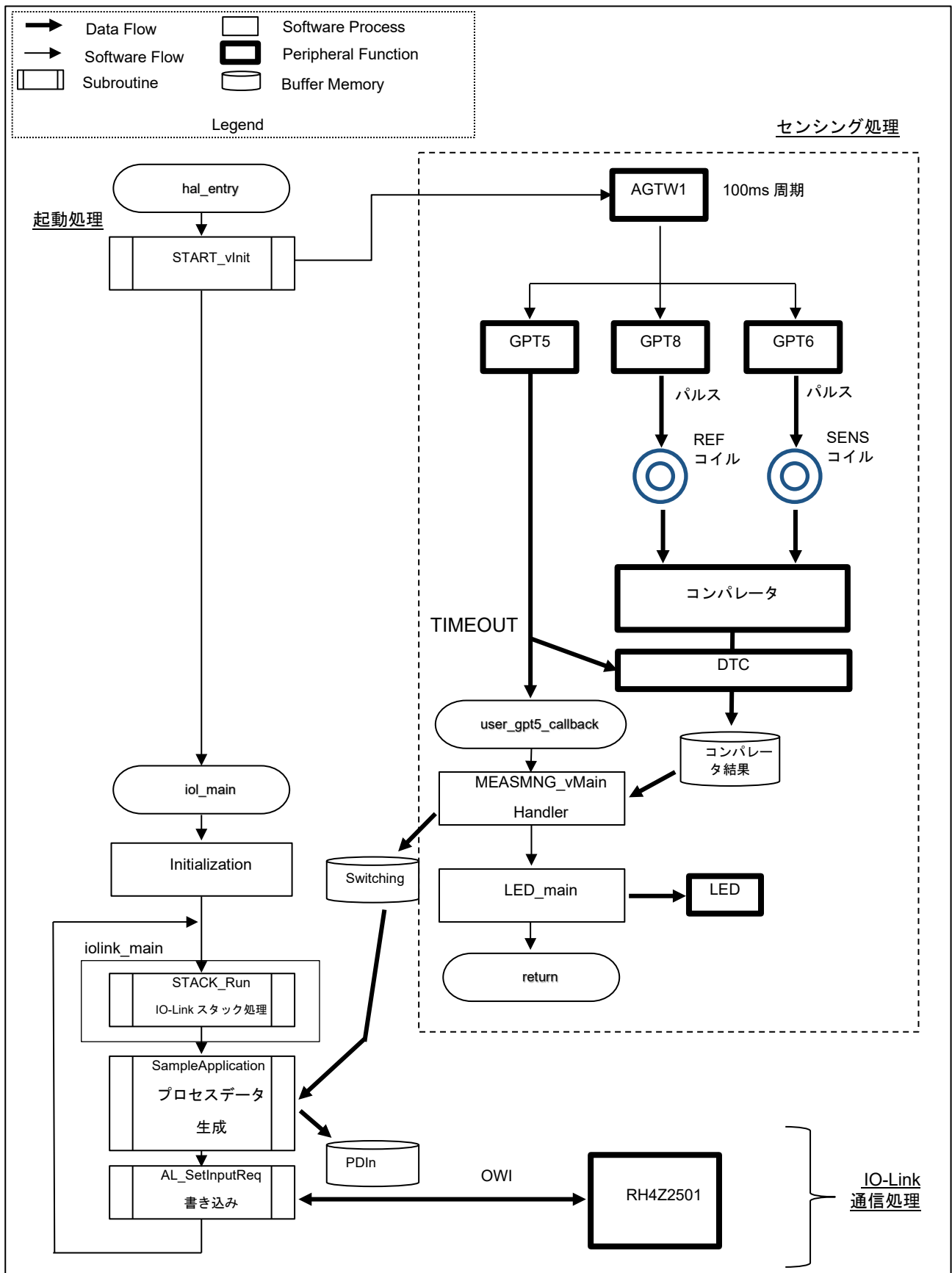


図 5-4 サンプルアプリケーション処理フロー

各処理の概略を記載します。

### 1. 起動処理

AGTW タイマ起動と IO-Link 処理の起動を行います。

#### A. AGTW 起動

AGTW タイマを 100ms 周期設定で起動します。

#### B. IO-Link 処理起動

iol\_main()を呼び出します。

### 2. センシング処理

#### A. AGTW タイマハンドラ

AGTW タイマにより 100ms 周期で起動されます。

REF コイルパルス用タイマ(GPT8)、SENS コイルパルス用タイマ(GPT6)、サンプリングポイント用タイマ(GTP5)を同時に開始します。

#### B. コイルパルス駆動

GPT8(REF コイル用)と GPT6(SENS コイル用)により、のこぎり波を発生させてコイルのパルス駆動を行います。

#### C. サンプリング処理

GPT5 開始から 2500ns でタイムアウトし、DTC が起動されます。

DTC により、REF コイルと SENS コイルのコンパレータ結果を取得します。

さらにユーザーハンドラ user\_gpt5\_callback()が起動されます。

#### D. user\_gpt5\_callback

コンパレータ結果から IO-Link 送信用のプロセスデータの生成と、LED 処理を行います。

### 3. 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 の 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 の 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。

### 5.4 スイッチング状態と LED 動作イメージ

センシングコイルで金属を検出するとスイッチング状態が ON になります。スイッチング状態に連動して LED も点灯します。Config.Logic=Low の時、スイッチング状態に対して LED の点灯論理が反転します。Config.LED=Blinking に設定すると、LED 点灯から点滅動作に切り替わります。

以下に動作イメージを記載します。

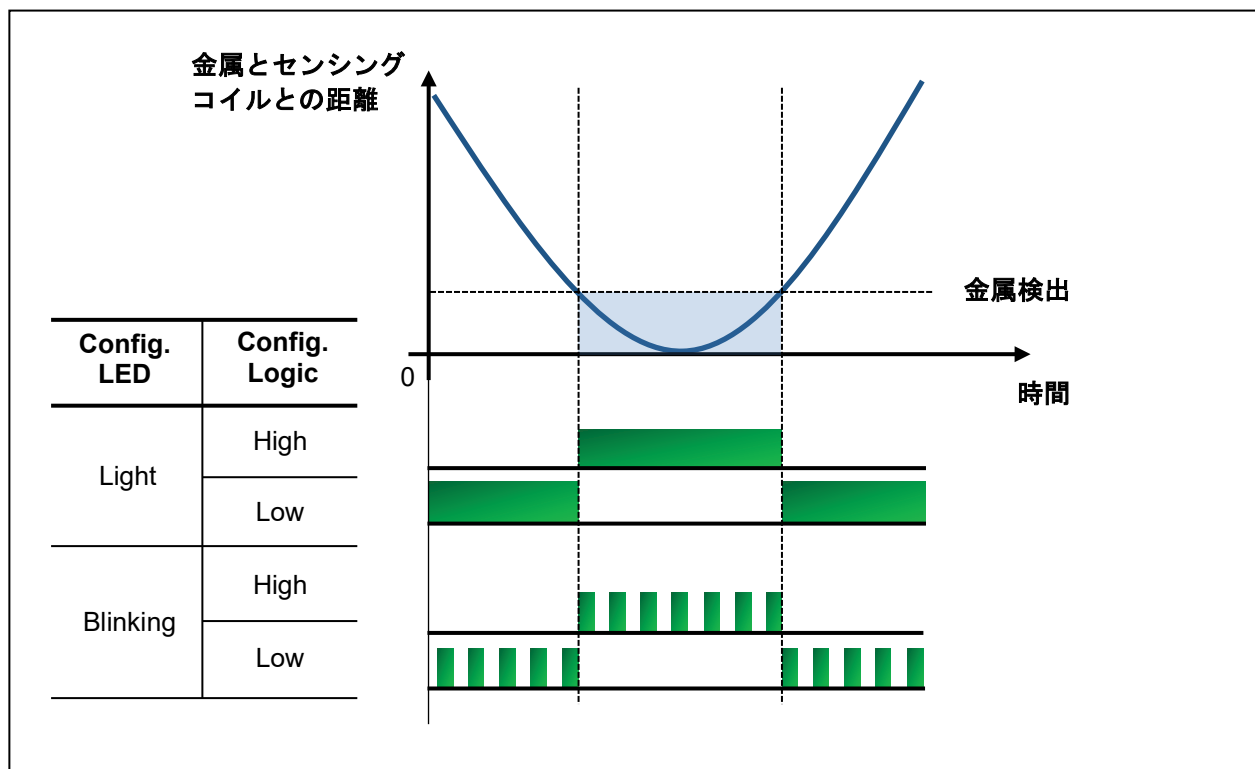


図 5-5 LED 動作イメージ

表 5-1 Switching 状態と LED 動作状況

Config.Logic \ 金属 検出/非検出	金属 検出/非検出	Switching 状態 <sup>注1</sup>	LED 動作 <sup>注1</sup>
High	非検出	OFF	OFF
	検出	ON	ON <sup>注2</sup>
Low	非検出	OFF	ON <sup>注2</sup>
	検出	ON	OFF

【注】 1.Config.Mode=Deactivated の時は Switching 状態と LED は常に OFF です。  
 2.Config.LED=Blinking の時、LED は点滅動作をします。

## 5.5 IO-Link 通信仕様

IO-Link の通信仕様について記載します。

### 5.5.1 ビットレート

ビットレートは COM3(230.4 kbps)です。

### 5.5.2 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。

### 5.5.3 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 5-2 に記載します。

**表 5-2 プロセスデータ (PDIn)**

プロセスデータ (PDIn) データ長 : 1[byte]			
PDIn[n]	Bit	格納データ	Details
PDIn[0]	1-6 0	- Switching Signal	(不使用) Switching Signal 状態を格納

### 5.5.4 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 5-3 に記載します。

**表 5-3 設定パラメータ一覧**

名称 (型)	ビット 数	値域 (初期値)	R/W	単位	概要
Switching Signal Channel1					
Config Mode (unsigned char)	8	0, 1 (1)	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
Config Logic (unsigned char)	8	0, 1 (0)	RW	-	LED 点灯論理設定 0 : High active 1 : Low active
Config LED (unsigned char)	8	0, 1 (0)	RW	-	LED 点灯状態 0 : Light, 1 : Blinking



5.6 使用する端子

本アプリケーションで使用する端子を以下に示します。

表 5-4 使用端子一覧

端子名	入出力	用途
P015	出力	Inductive Sensor Board の LED1 の制御
P101(GTIOC8A)	出力	Reference Coil 回路ライブパルス信号
P102	入力	外部コンパレータ結果入力
P111(GTIOC6A)	出力	Sensing Coil 回路ライブパルス信号
P400(IRQ0)	入力	IO-LINK ウェイクアップ信号
P401(GTIOC9B)	入出力	RH4Z2501 レジスタアクセス OWI 信号
P914(RXD9)	出力	C/Q トランシーバ制御 UART 受信信号
P205(TXD9)	入力	C/Q トランシーバ制御 UART 送信信号
P103	入力	C/Q トランシーバ制御 UART 送信イネーブル信号

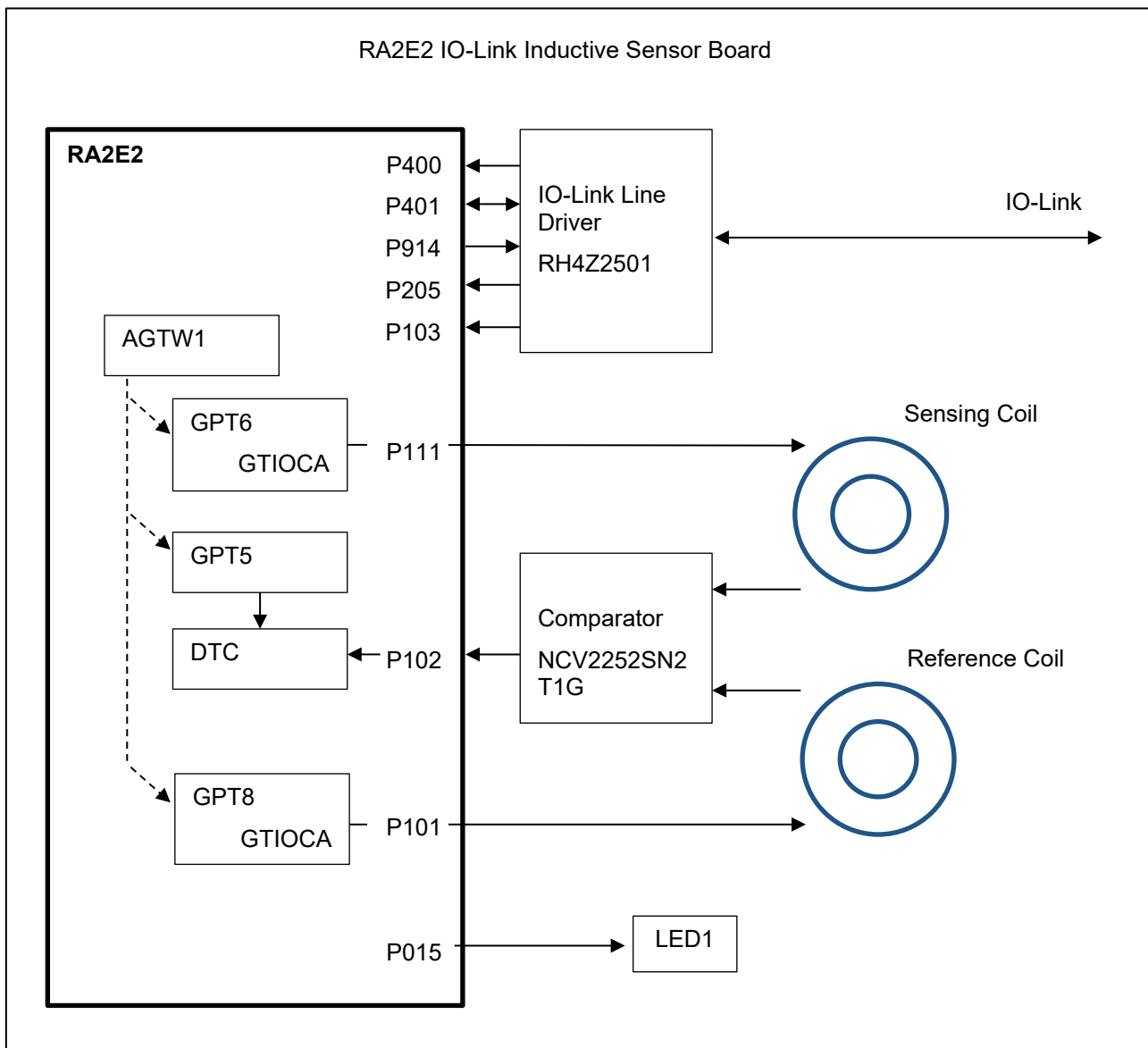


図 5-6 使用する周辺機能と端子

## 5.7 プログラム構成

### 5.7.1 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 Manuals フォルダ内に配置されています。ファイル構成を表 5-5 に記載します。

表 5-5 ファイル構成

フォルダ名、ファイル名	説明
ROOT	
└─IO-Link	IO-Link フォルダ(TBD)
├─Application	アプリケーションフォルダ
main.c	アプリケーション main
StarterkitSampleApplication.c	アプリケーションプログラム
StarterkitSampleApplication.h	アプリケーションヘッダ
Starterkit_Config.h	Starterkit_Config.h
├─BSP	BSP フォルダ
BSPStack.h	User-implemented Hardware Abstraction definitions
IOLD_Config.h	system specific Defines
SystemInit.h	Hardware settings definitions
├─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-RA2E2-Starterkit-20230313-IODD1.1.xml	IODD1.1 ファイル
TMG-RA2E2-Starterkit-con-pic.png	M12 コネクタ 4Pin 画像ファイル
TMG-RA2E2-Starterkit-icon.png	ボードアイコン画像ファイル
TMG-RA2E2-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_RA2E2_lib.a	IO-Link Library for the starterkit solution
ParameterManager.h	Interface to the application framework
Profile_common.h	common Declarations for Profiles
├─Manuals	TMG IO-Link Device Stack マニュアルフォルダ
Dokumentation.chm	Dokumentation.chm
TMG IO-Link Device Stack Users Manual.pdf	IO-Link Device Stack User's manual Stack Revision 1.1.147
TMG IO-Link StackExtensions Users Manual.pdf	IO-Link Device Stack Extensions User's manual Stack Revision 1.1.218

TMG StackExtensions Sample.pdf	User Manual Sample Application for TMG IO-Link Device Stack and Stack Extensions
TMG_IO-Link_DeviceApplication_ReleaseNotes.txt	TMG_IO-Link_DeviceApplication_ReleaseNotes
TMG_IO-Link_DeviceStackExtensions_ReleaseNotes.txt	TMG_IO-Link_DeviceStackExtensions_ReleaseNotes
TMG_IO-Link_DeviceStack_ReleaseNotes.txt	TMG_IO-Link_DeviceStack_ReleaseNotes
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
src	アプリケーションソースフォルダ
hal_entry.c	hal_entry プログラム
SystemConfig.h	SystemConfig ヘッダ
COIL_SENSORS	センサコイル用フォルダ
COILSENSORS.c	センサコイル操作関数
COILSENSORS.h	センサコイル操作ヘッダ
EXCIT	コイルパルス発生フォルダ
EXCIT.c	コイルパルス発生用関数
EXCIT.h	コイルパルス発生用ヘッダ
LED	LED 処理
LED.c	LED 操作関数
LED.h	LED 処理用ヘッダ
MEASMNG	計測用フォルダ
MEASMNG.c	計測用関数
MEASMNG.h	計測用関数ヘッダ
START	スタートフォルダ
START.c	スタート関数
START.h	スタート関数ヘッダ

## 5.7.2 関数一覧

以下では、「5.3 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。

### 5.7.2.1 hal\_entry.c

[関数名] hal\_entry

---

概要	起動処理
ヘッダ	なし
宣言	void hal_entry(void)
説明	TMG 社から提供された hal_entry()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。アプリケーションで使用する周辺機能や IO-Link stack ライブラリ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

### 5.7.2.2 START.c

[関数名] START\_vlnit

---

概要	起動処理
ヘッダ	START.h
宣言	void START_vlnit()
説明	MEASMNG、EXCIT、COILSENSOR、LED の各初期化処理実行後に、AGT タイマ起動処理を実行します。
引数	なし
リターン値	なし
備考	なし

## 5.7.2.3 EXCIT.c

[関数名] user\_gpt5\_callback

---

概要	GPT5 タイマハンドラ	
ヘッダ	EXCIT.h	
宣言	void user_gpt5_callback(timer_callback_args_t *p_args)	
説明	計測ポイント用ハンドラです。2500ns に設定された GPT5 タイマのタイムアウトにより起動され、DTC で取得されたコンパレータ結果から Switching Data を生成し、その結果を LED 処理に引き渡します。	
引数	p_args	引数ポインタ
リターン値	なし	
備考	なし	

## 5.7.2.4 MEASMNG.c

[関数名] MEASMNG\_vMainHandler

---

概要	金属検出処理	
ヘッダ	MEASMNG.h	
宣言	void MEASMNG_vMainHandler()	
説明	GPT6/8 を停止後、DTC で取得されたコンパレータ結果から金属検出を判定します。	
引数	なし	
リターン値	なし	
備考	なし	

## 5.7.2.5 LED.c

[関数名] LED\_main

---

概要	LED 制御	
ヘッダ	LED.h	
宣言	void LED_Control(bool switching)	
説明	LED 表示処理を行います。 引数としてコンパレータ結果の Switching 状態を受け取ります。 Switching 状態に応じて LED を点灯/消灯します。 Config.Mode が Deactivated 時は Switching 及び LED を OFF とします。 Config.Logic が Low active(1)の場合は LED を反転させます。	
引数	Switching	true : ON      false : OFF
リターン値	なし	
備考	なし	

## 5.7.2.6 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	IO-Link stack ライブラリの 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

## 5.7.2.7 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

## 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. e<sup>2</sup> 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。

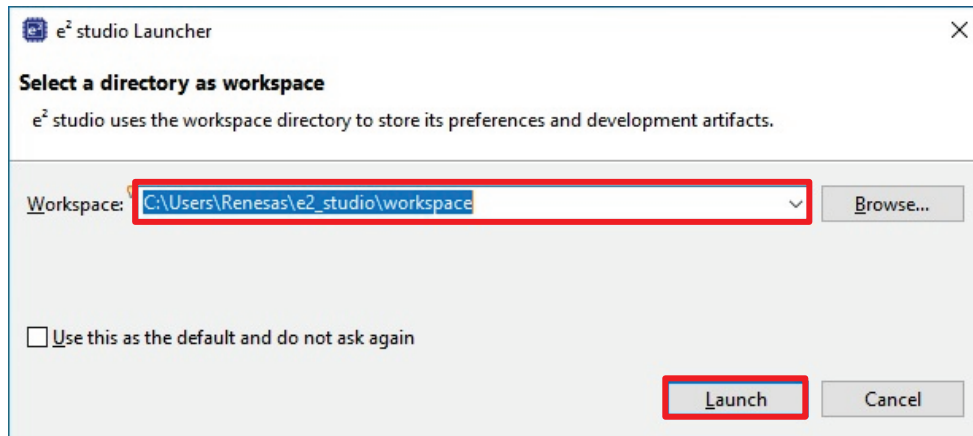


図 6-1 新規ワークスペースの使用開始

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

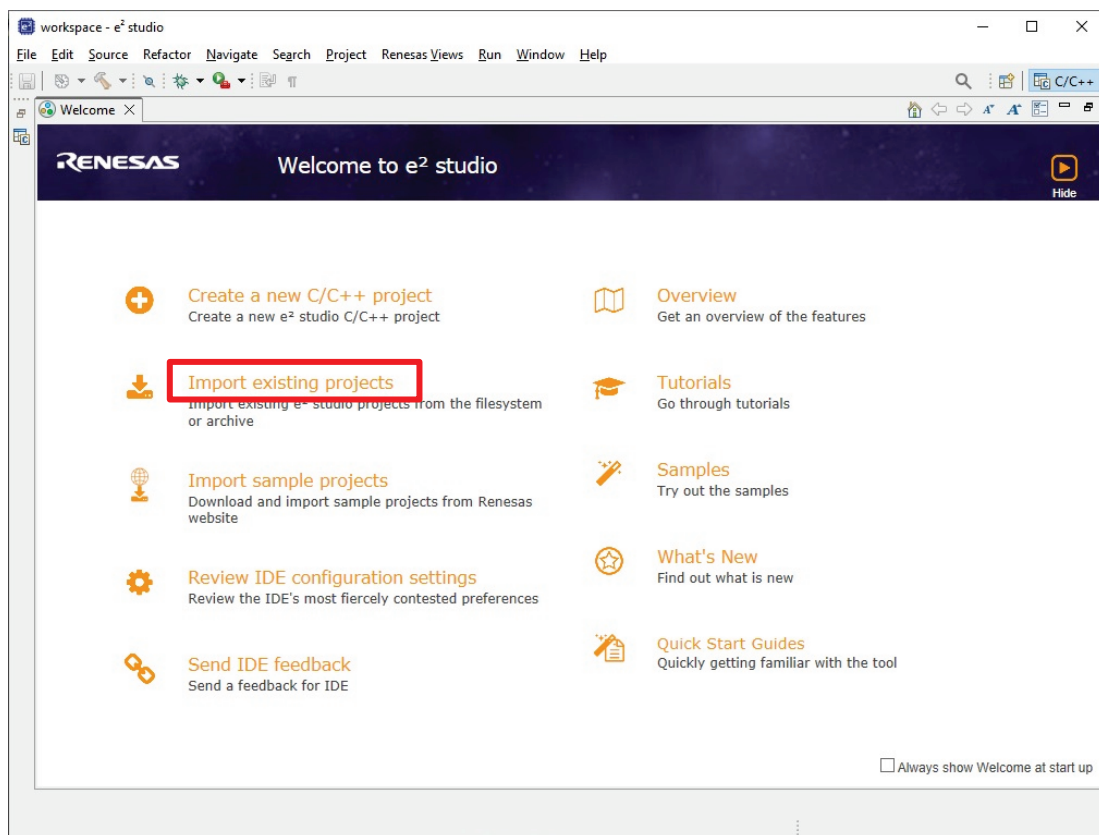


図 6-2 Welcome to e<sup>2</sup>studio

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。

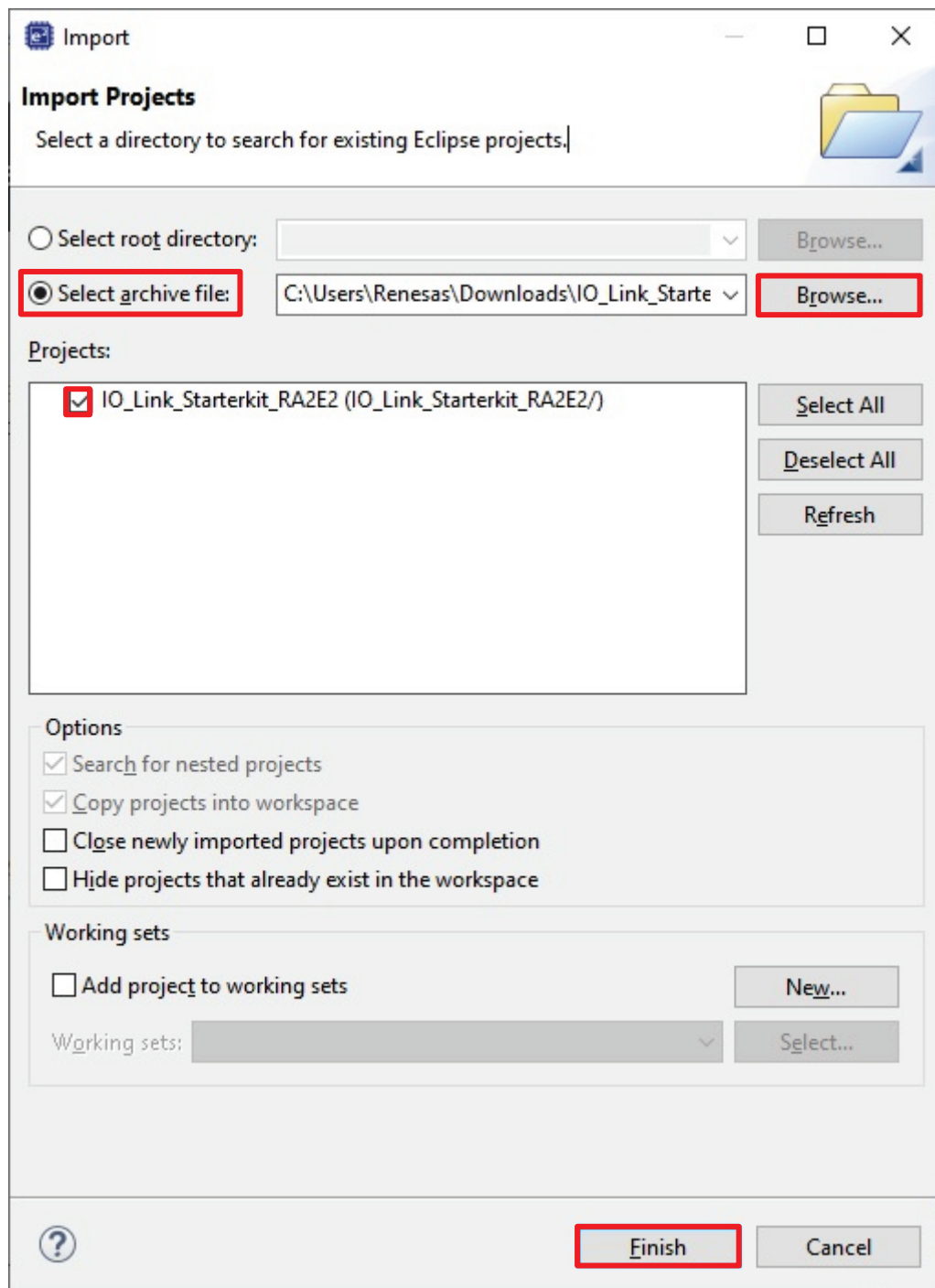


図 6-3 サンプルプロジェクトのインポート



## 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スマートコンフィグレータを開きます。

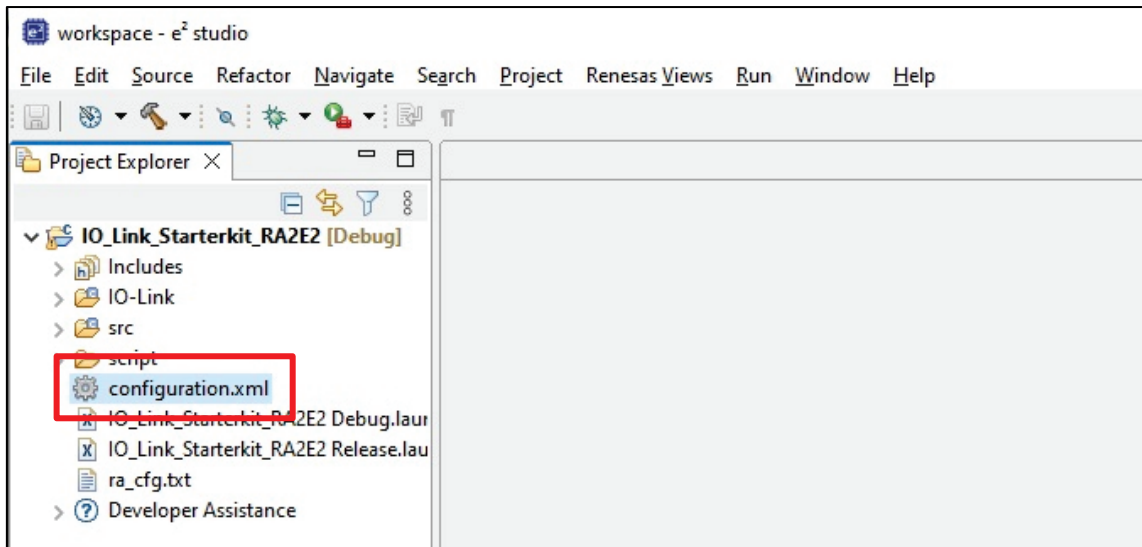


図 6-4 スマートコンフィグレータの起動

2. Generate Project Content をクリックしてください。スマートコンフィギュレータが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。

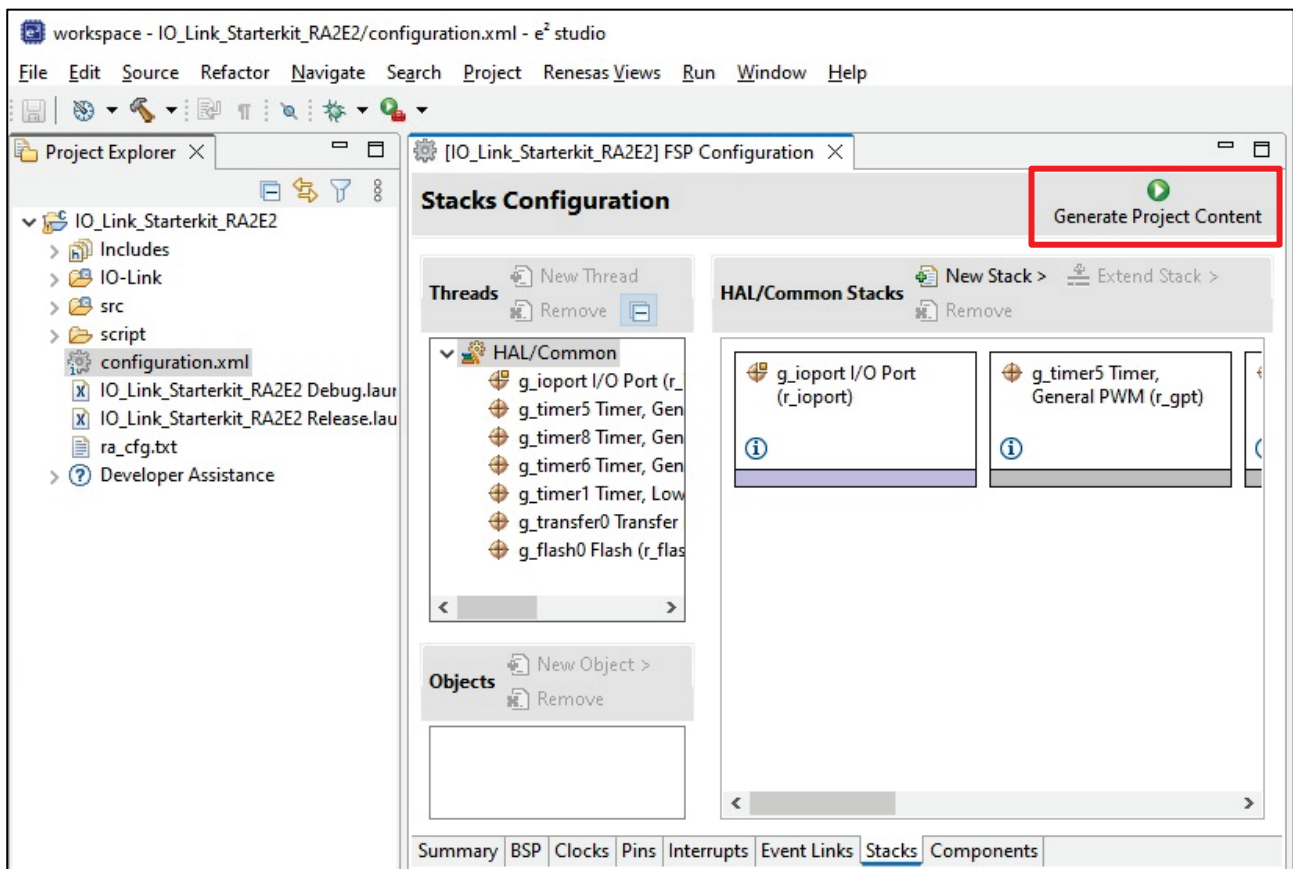


図 6-5 プロジェクトコンテンツの生成

3. Project Explorer でプロジェクトを右クリックし、“Build Project”を選択してプロジェクトをビルドしてください。

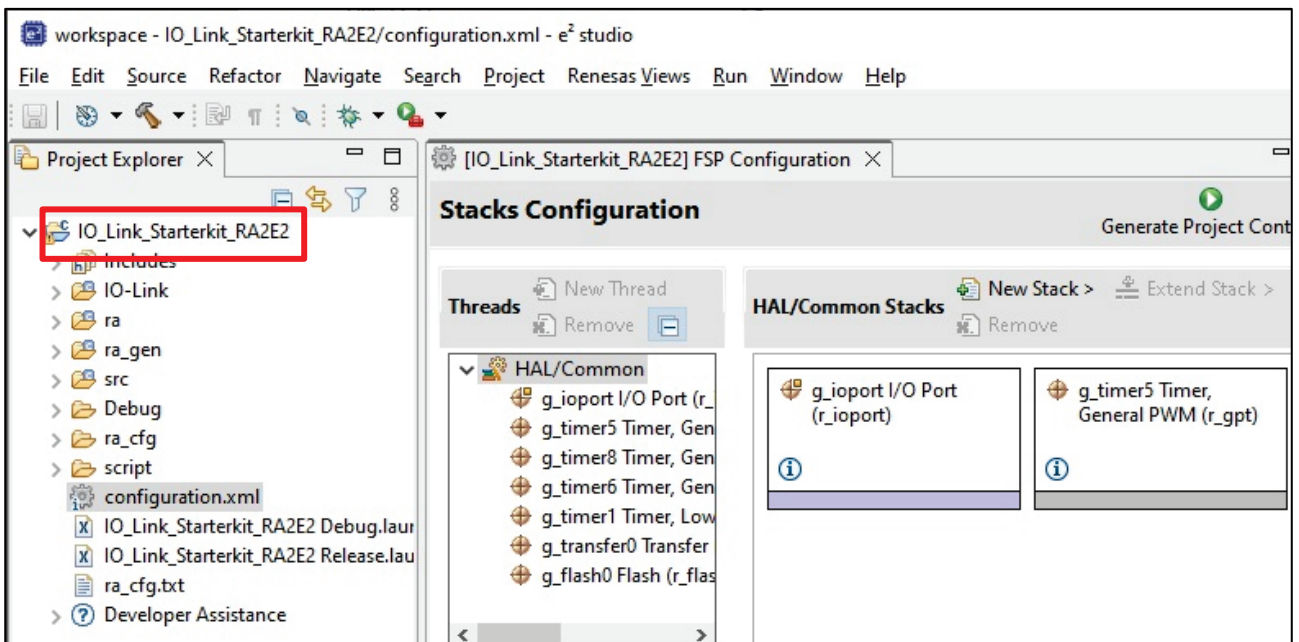


図 6-6 プロジェクトのビルド

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。

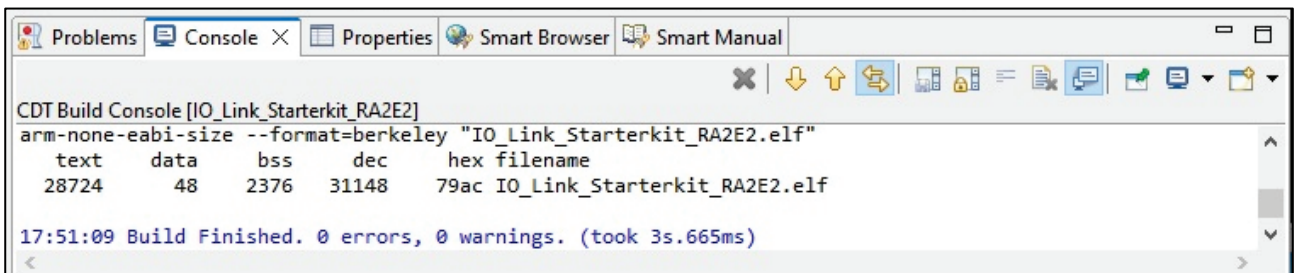


図 6-7 ビルド成功時のアウトプット

### 6.3 デバッグ接続のセットアップ

1. M12 ケーブルで USB マスターと Inductive Sensor Board を接続します。



図 6-8 USB マスターと Inductive Sensor Board 接続

【注】 USB マスターが電源に接続された状態で Inductive Sensor Board を接続すると、Inductive Sensor Board が故障する可能性があります。USB マスターが電源に接続されていない状態で接続してください。

以下の手順で接続するようにしてください。

- ①USB マスターと Inductive Sensor Board 接続
  - ②USB マスターと PC の接続
2. ユーザーインターフェースケーブル(20-10pin) [RTE0T00020KCAC1000J]を E2 エミュレータ Lite と Inductive Sensor Board に接続します。

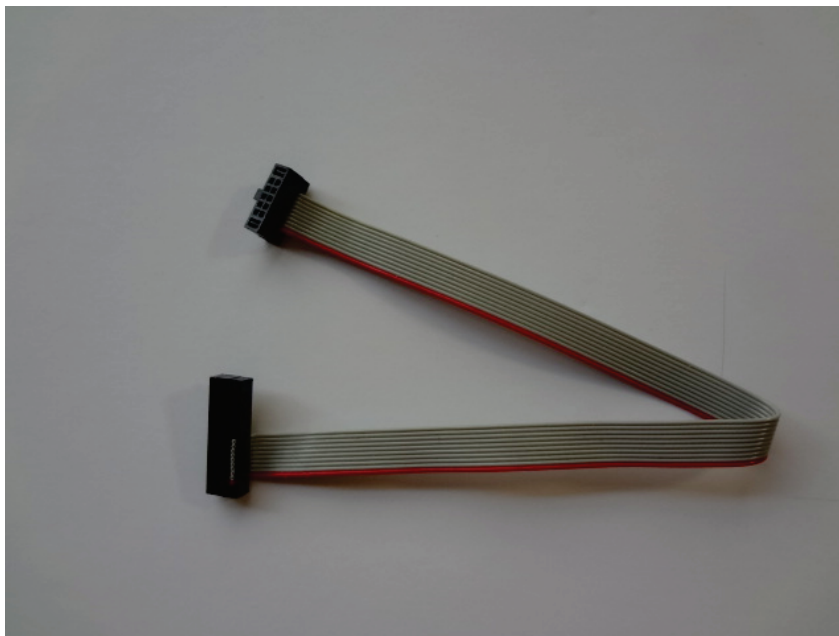


図 6-9 ユーザーインターフェースケーブル(20-10pin)

Inductive Sensor Board 側の 10pin コネクタは、凹凸の位置を合わせて接続します。

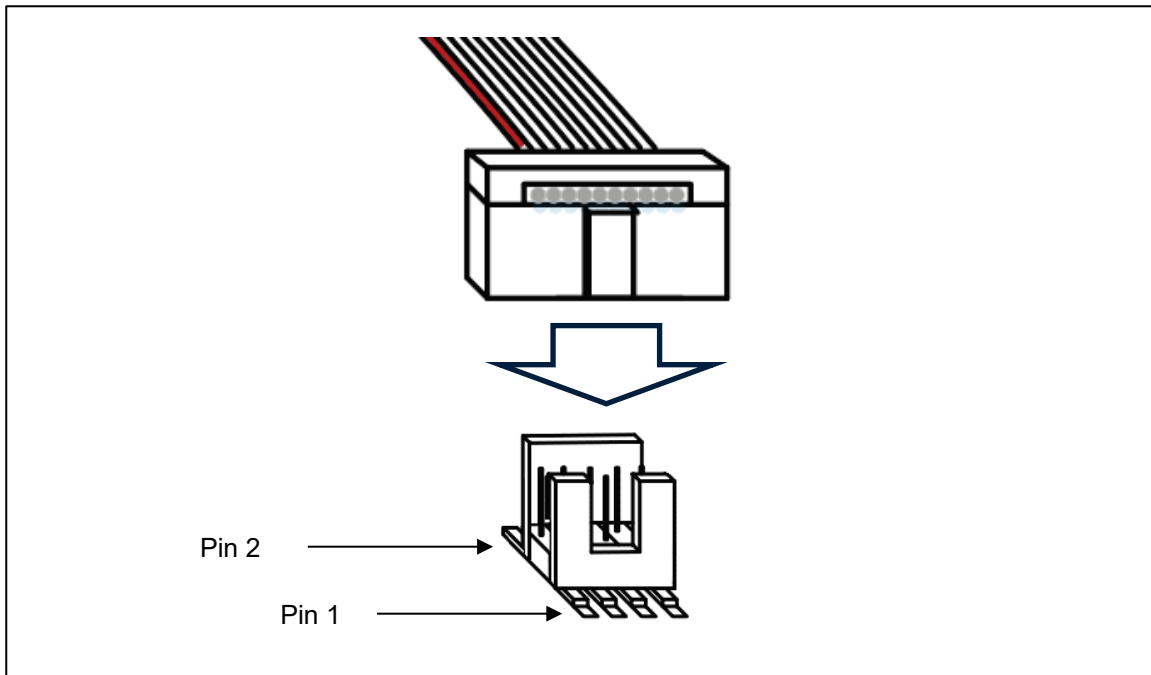


図 6-10 10pin コネクタの接続方向

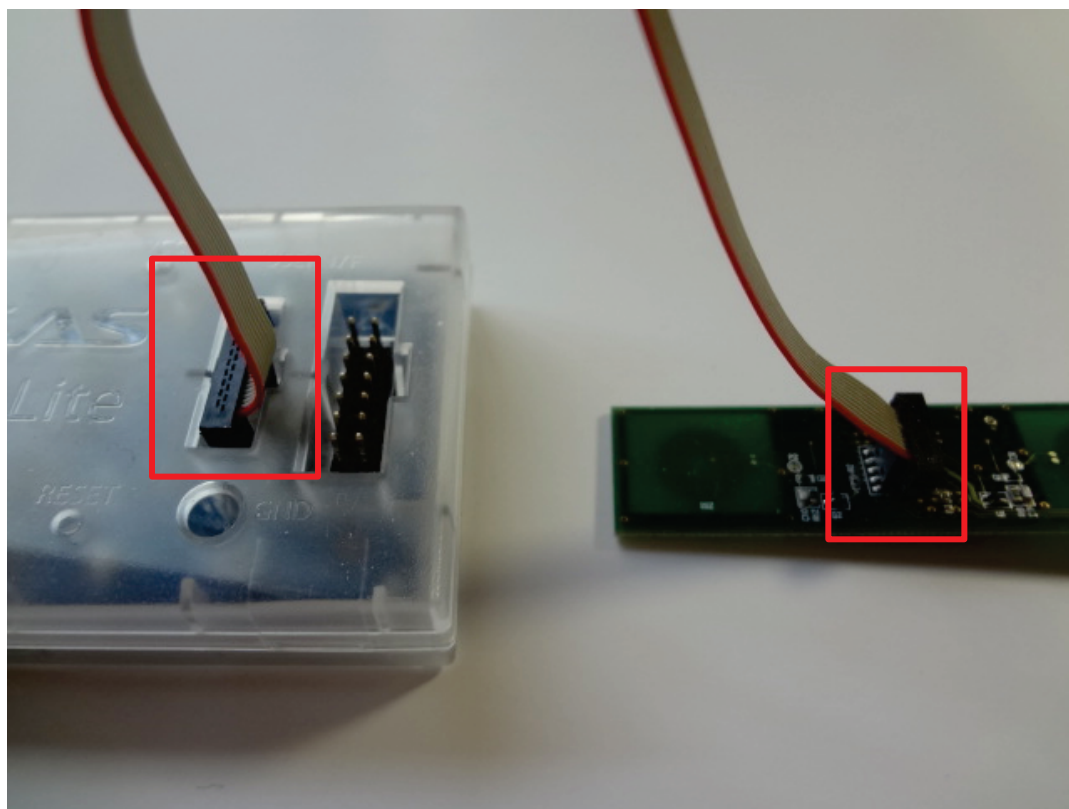


図 6-11 ユーザーインターフェースケーブル接続

3. PC と E2 エミュレータ Lite を USB ケーブルで接続します。E2 エミュレータ Lite の ACT LED が点滅します。

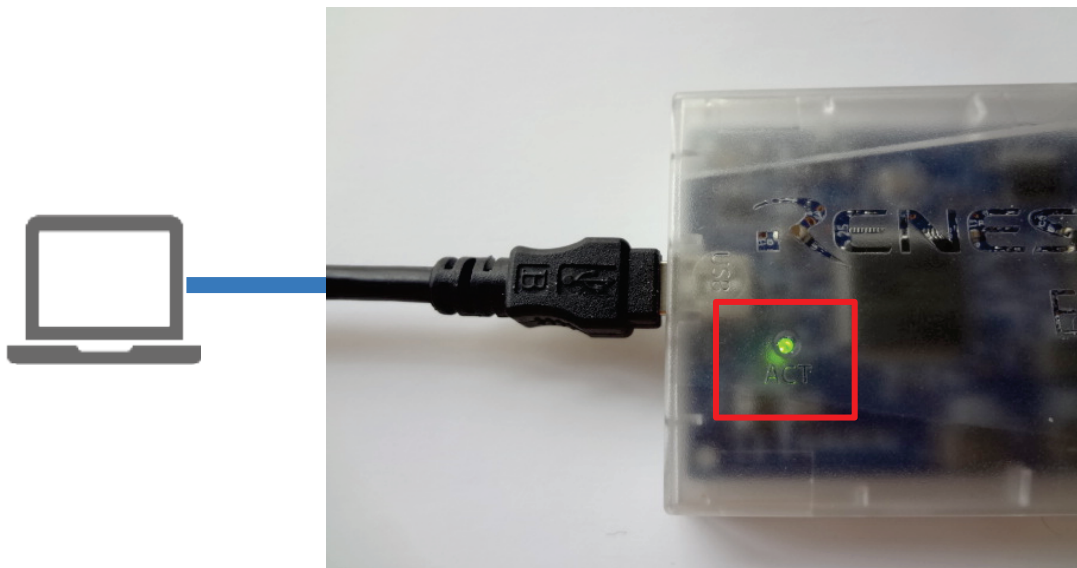


図 6-12 E2 エミュレータ Lite の PC 接続

4. PC と USB マスターを USB ケーブルで接続します。USB マスターに電源が供給され PWR LED がオレンジに点灯します。

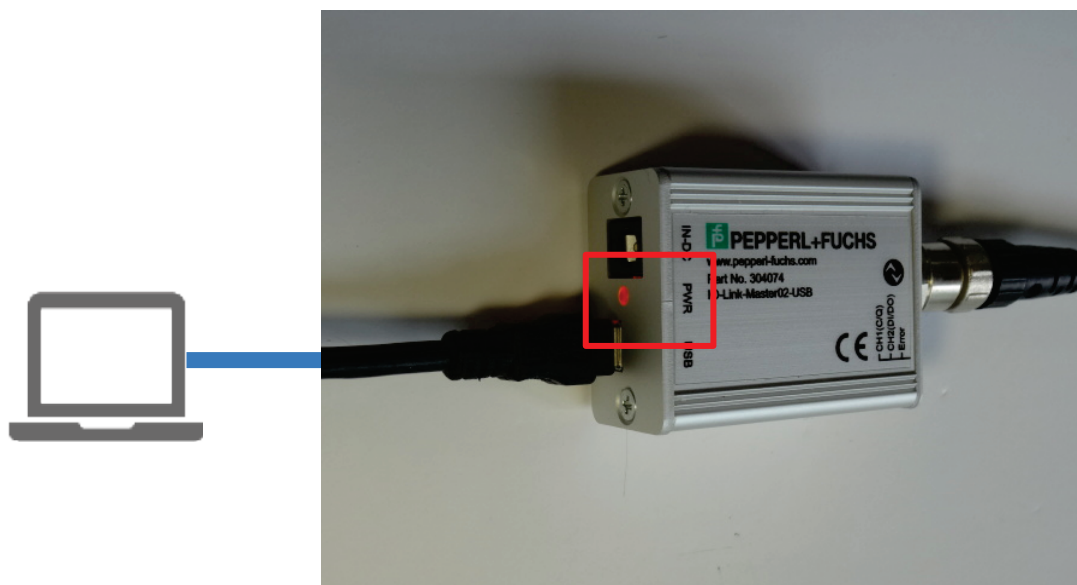


図 6-13 USB マスターの PC 接続

E2 エミュレータ Lite の VCC と RESET LED が点灯します。

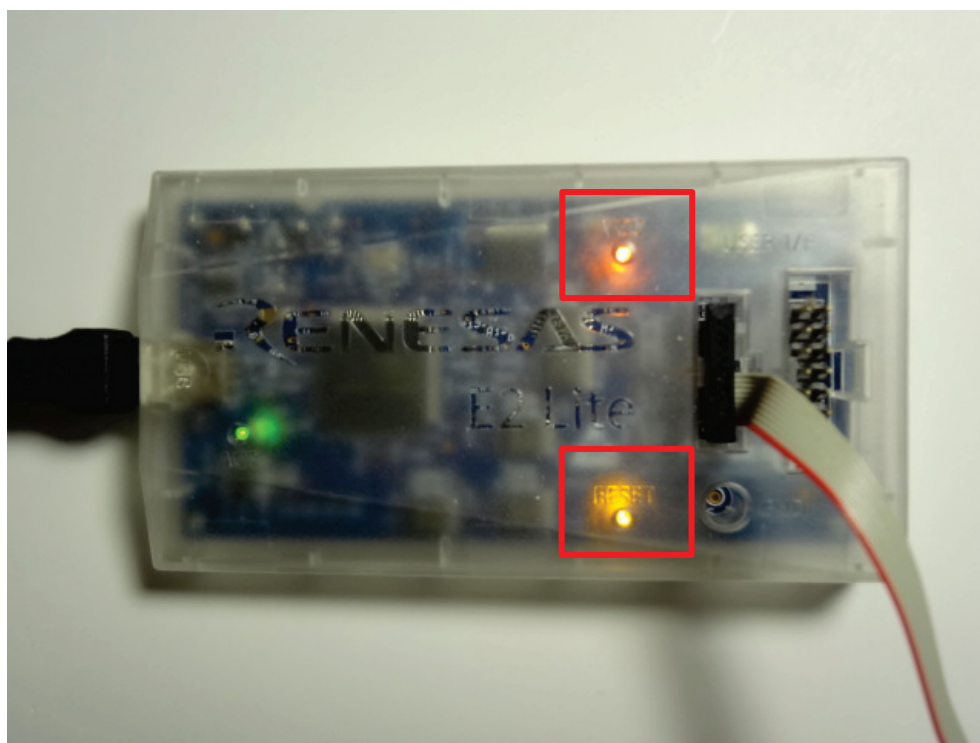


図 6-14 VCC/RESET LED の点灯

## 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。

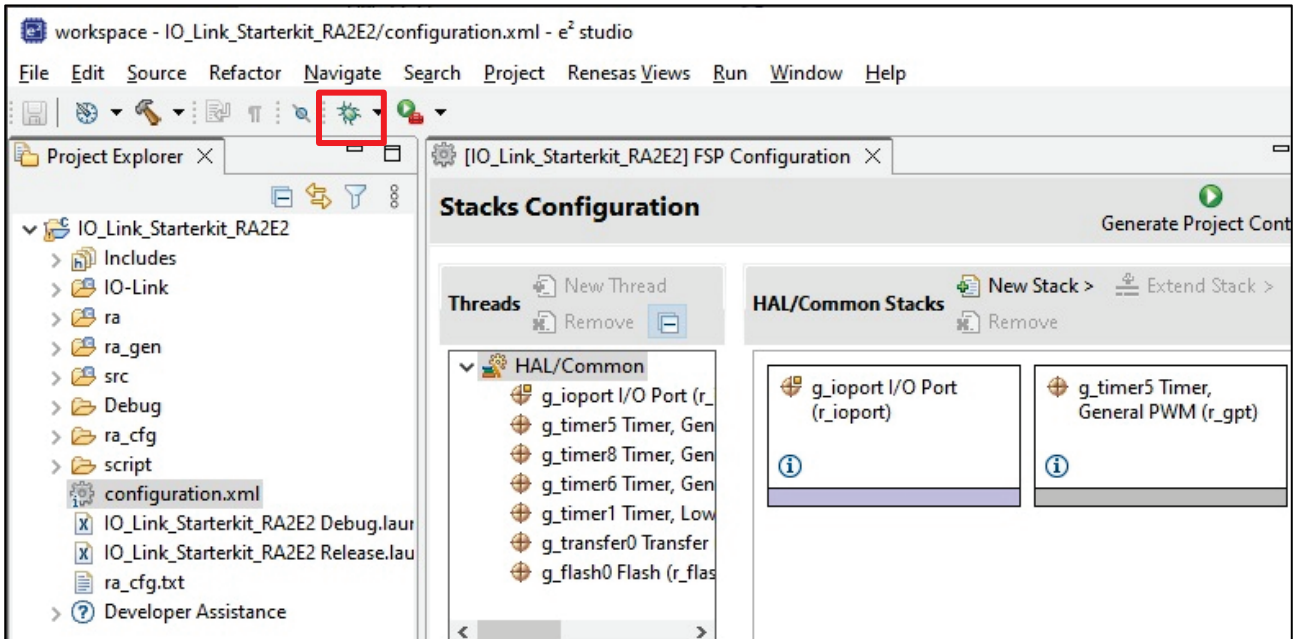


図 6-15 Launch in 'Debug' mode

2. 以下のダイアログが表示された場合、「IO\_Link\_Starterkit\_RA2E2\_Debug」を選択して、OK をクリックします。

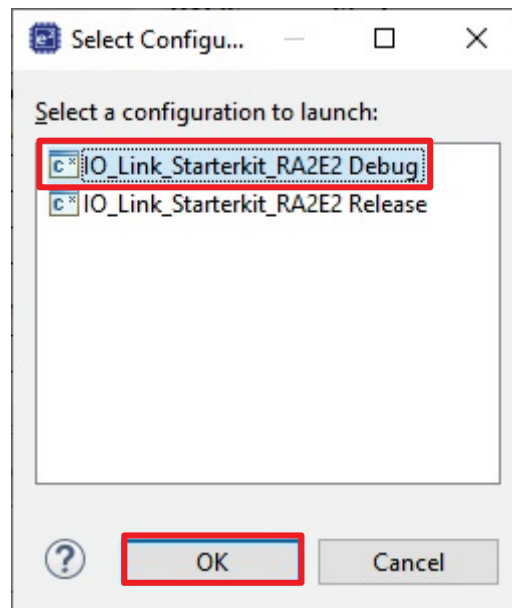


図 6-16 Select a configuration to launch

3. '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。

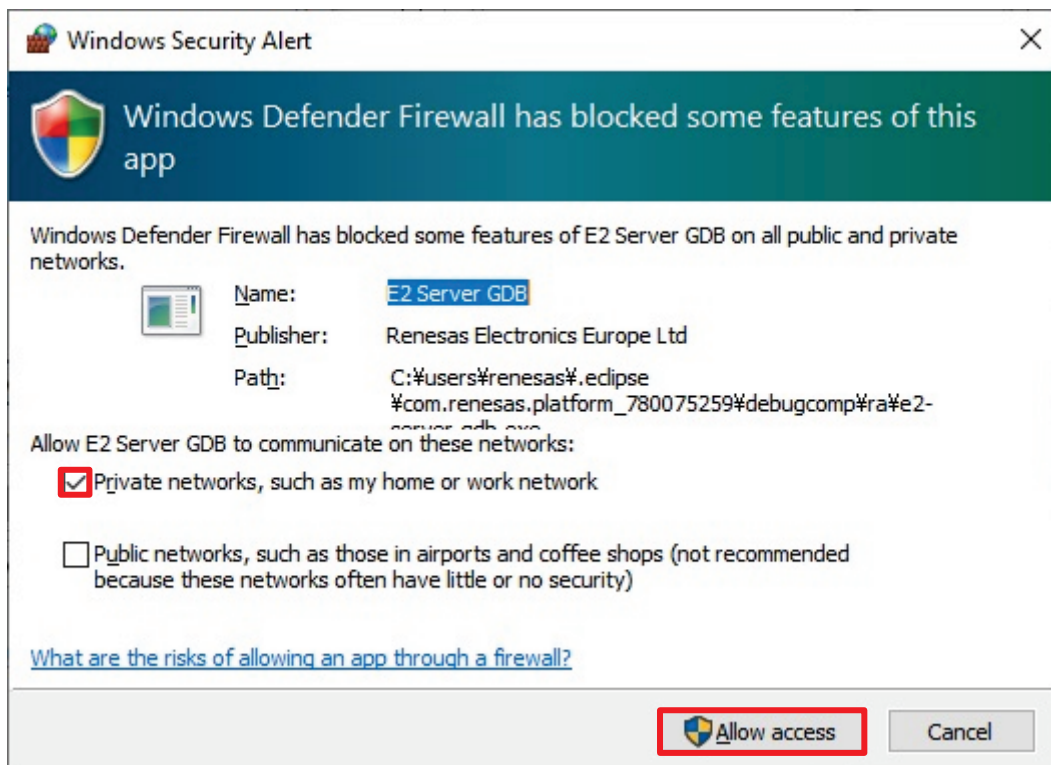


図 6-17 ファイアウォール警告

4. 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。

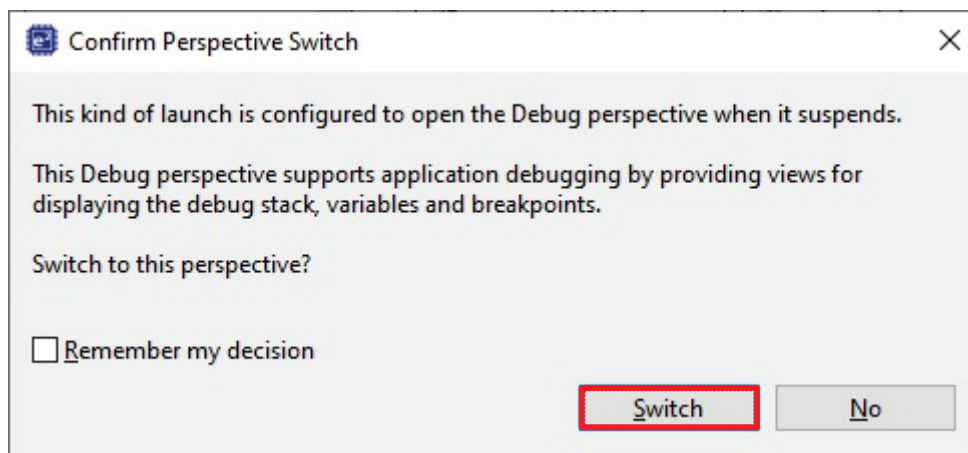


図 6-18 Confirm Perspective Switch



5. MCU へアプリケーションのフラッシュイメージデータの書き込みが実施され、画面が切り替わります。

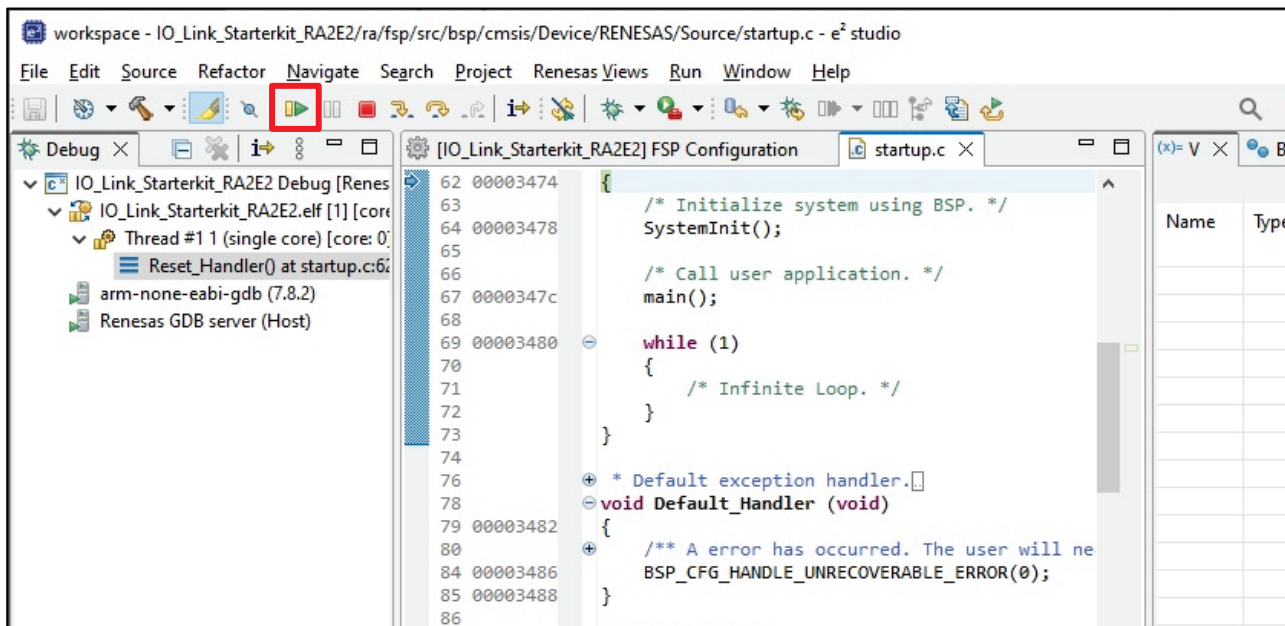


図 6-19 MCU への書き込み完了

## 6.5 プログラムの開始

1. 図 6-19 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。

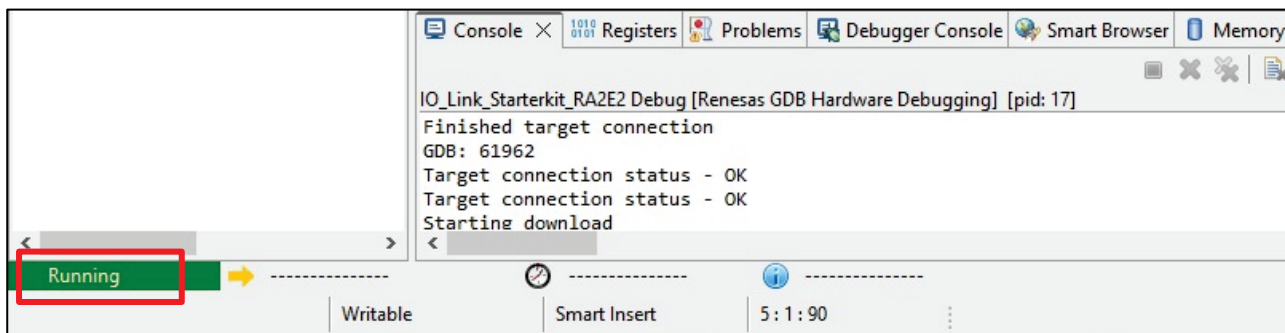


図 6-20 プログラムの開始

## 7. PACTware DC の使用準備及び機能解説

本章では、Pepperl + Fuchs PACTware の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。

【注】 必要に応じて以下の資料を参照してください。  
[IO-Link Offline Parameterization Installation/User Instructions](#)

### 7.1 .NET Framework 有効化

PACTware インストール前に.NET Framework の有効化を行います。

1. コントロールパネルの「プログラム」をクリックします。

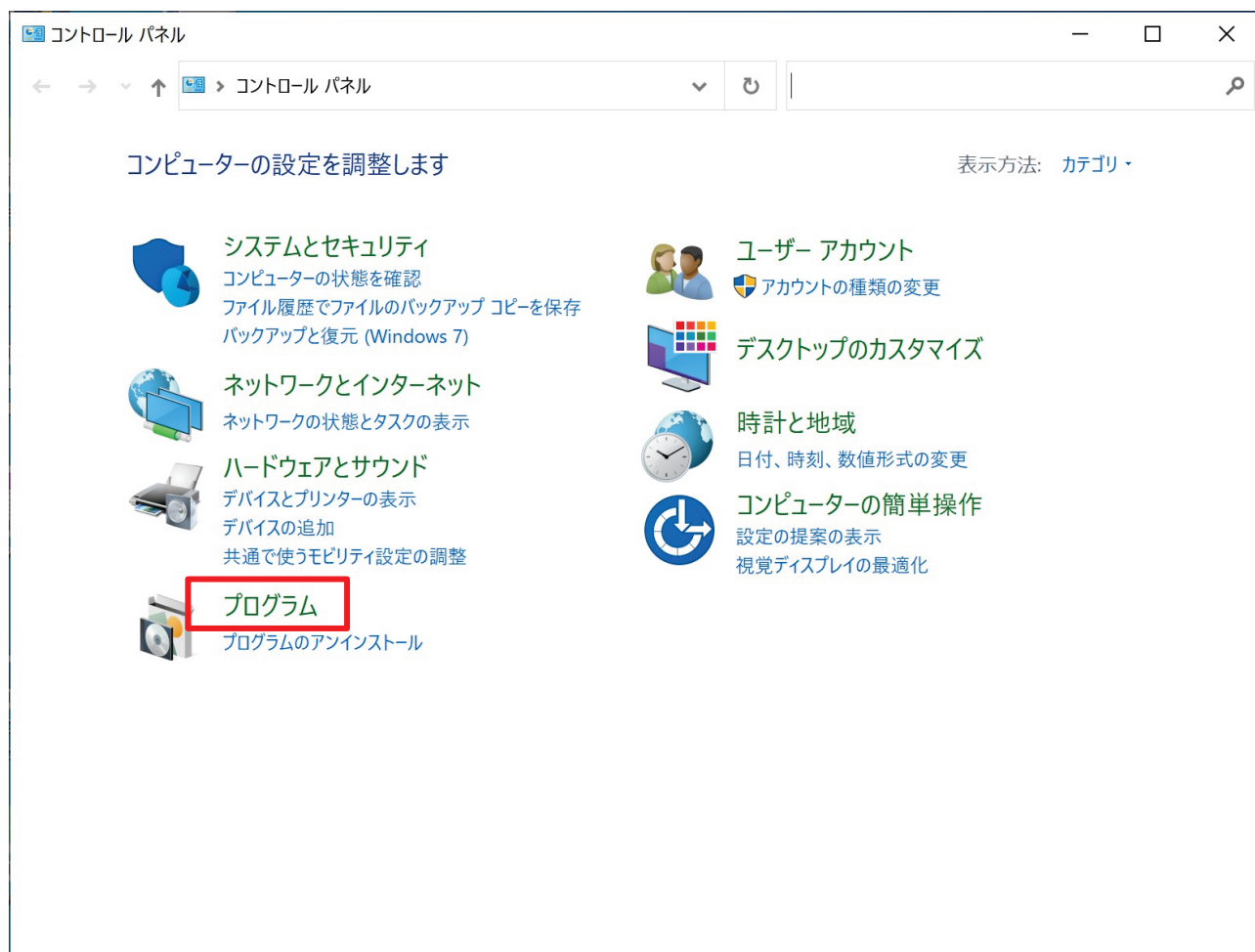


図 7-1 コントロールパネル - プログラム

2. 「Windows の機能の有効化または無効化」をクリックします。



図 7-2 Windows の機能の有効化または無効化

3. 「.NET Framework 3.5」と「.NET Framework 4.8」のチェックボックスを ON にして「OK」をクリックします。

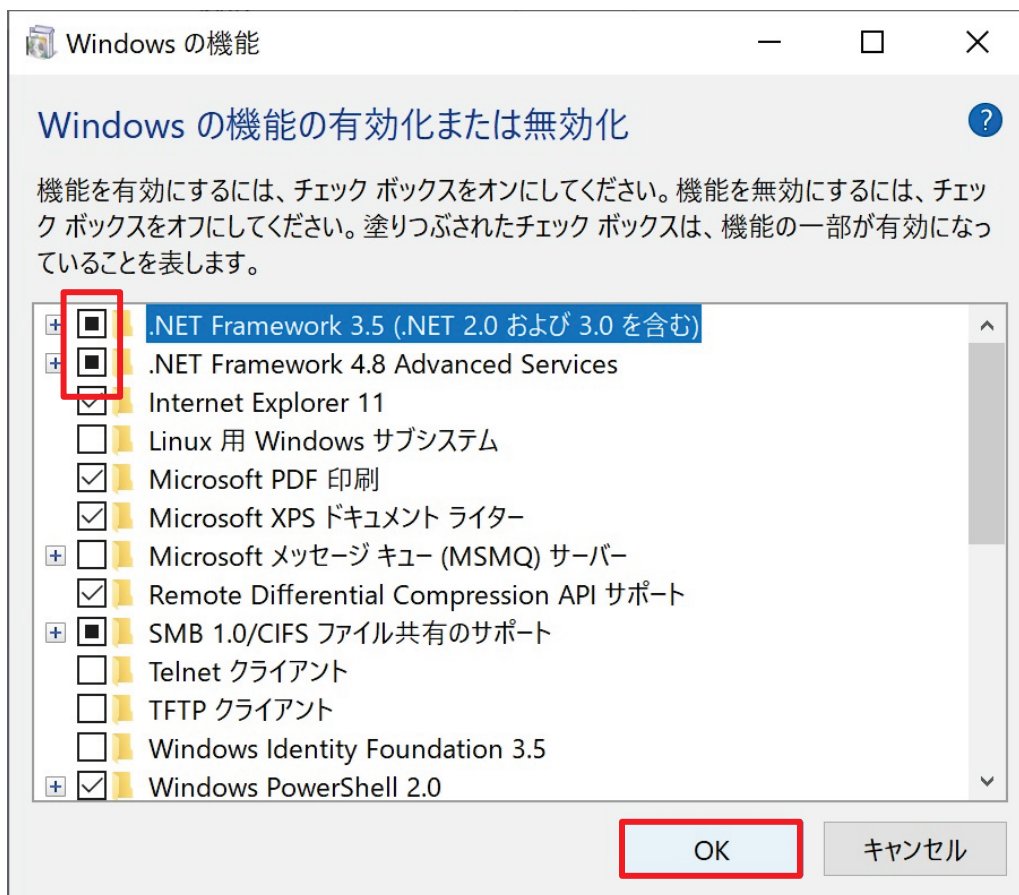


図 7-3 .NET Framework

- 【注】 System Requirements:
- ・.Net 3.5 SP1 and .Net 4.0 (usually supported in Windows 10)

4. 「Windows Update でファイルを自動ダウンロードする」を選択します。

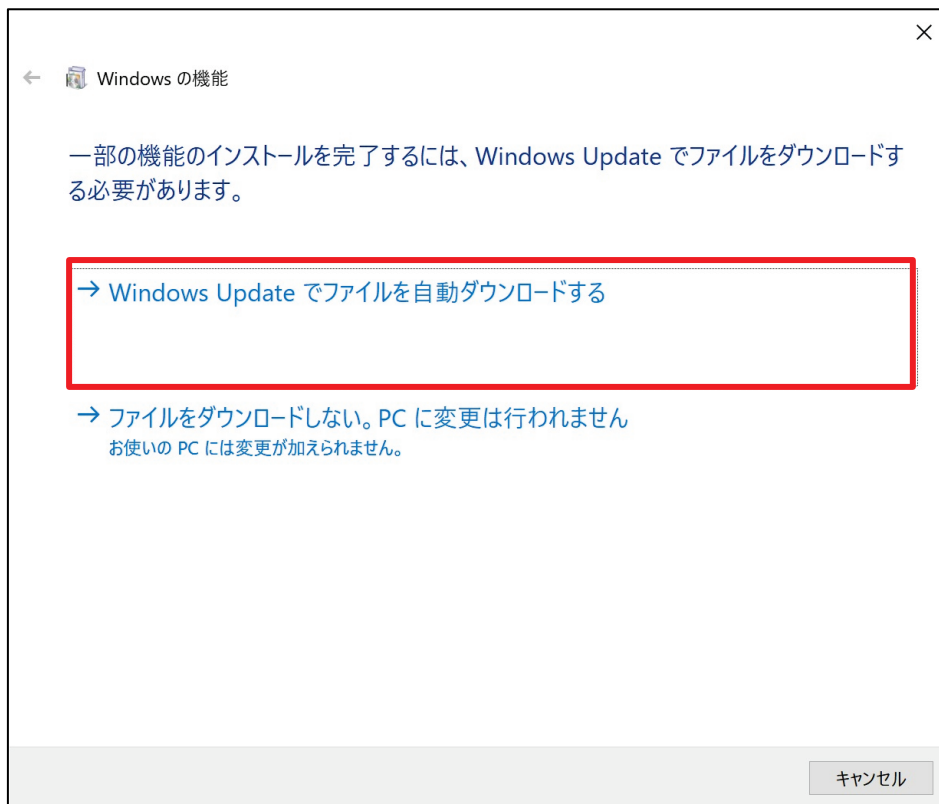


図 7-4 Windows の機能

5. 変更が完了したら「閉じる」をクリックします。



図 7-5 Windows の機能 - 変更完了

## 7.2 IO-Link Parameterization Tool インストール

1. [IO-Link Offline Parameterization Tool V1.00.006](#) をダウンロードして解凍します。
2. IO-Link\_ParamTool\_Setup.exe を実行します。
3. 「この不明な発行元からのアプリがデバイスに変更を加えることを許可しますか？」が表示された場合は「はい」をクリックします。
4. Select Language が表示されたら「OK」をクリックします。

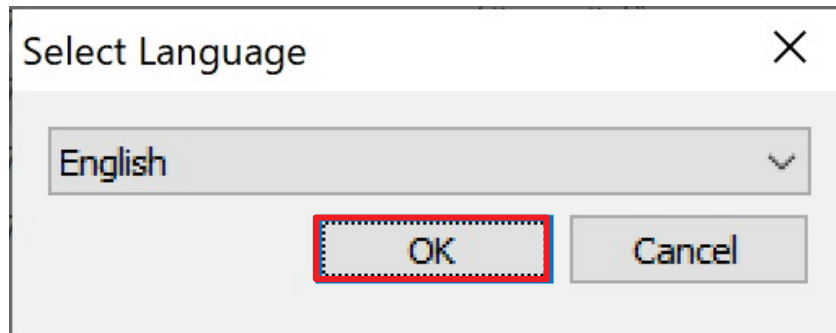


図 7-6 Select Language

5. 「I have read and accepted the license agreement.」のリンクをクリックして内容を確認します。チェックボックスをチェックして「Install」をクリックします。

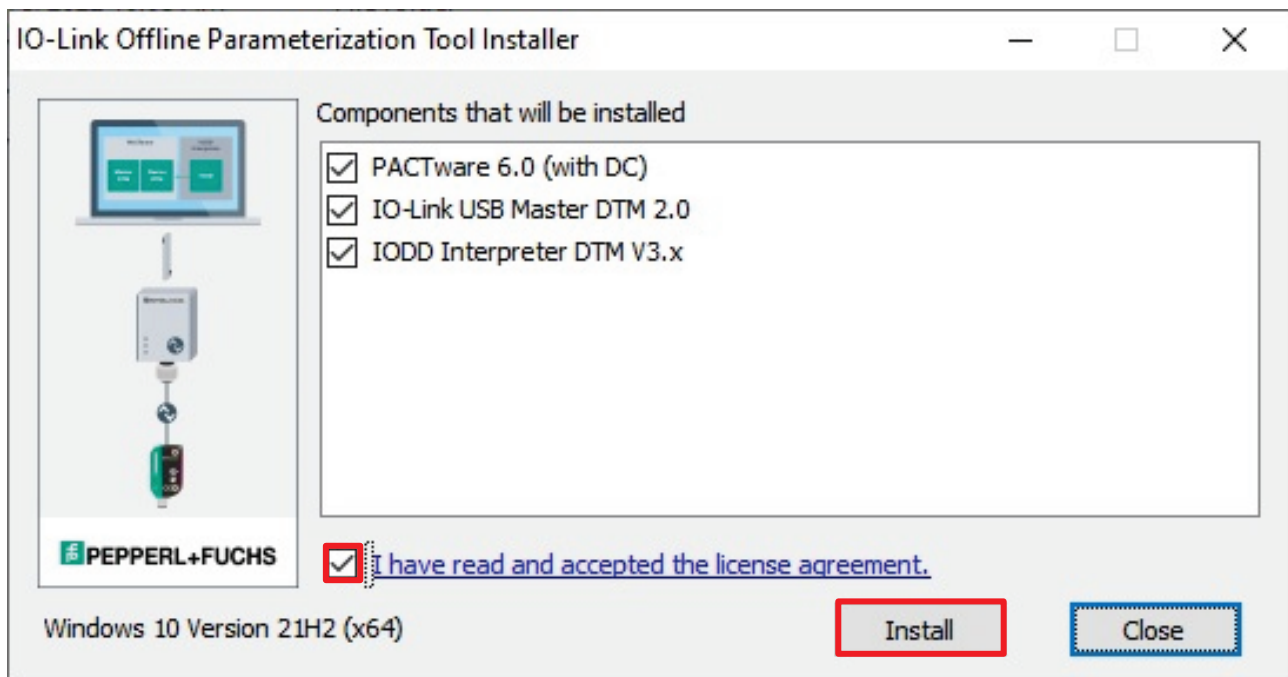


図 7-7 Components that will be installed

6. インストールが成功したら「OK」をクリックします。

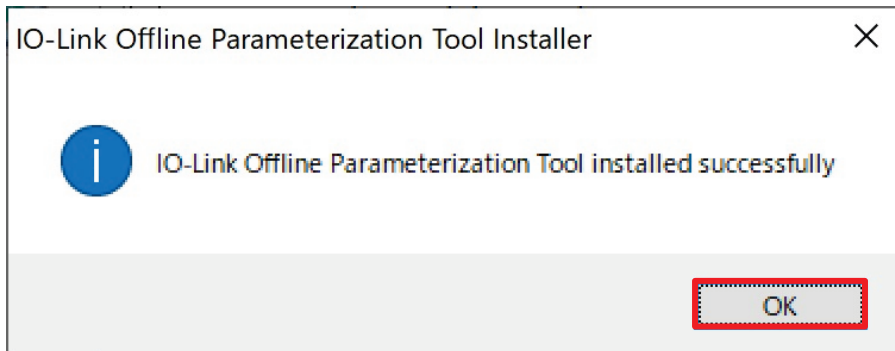


図 7-8 Installed successfully

### 7.3 IODD セットアップ

1. 初めにサンプルプロジェクトの zip ファイルを解凍しておきます。

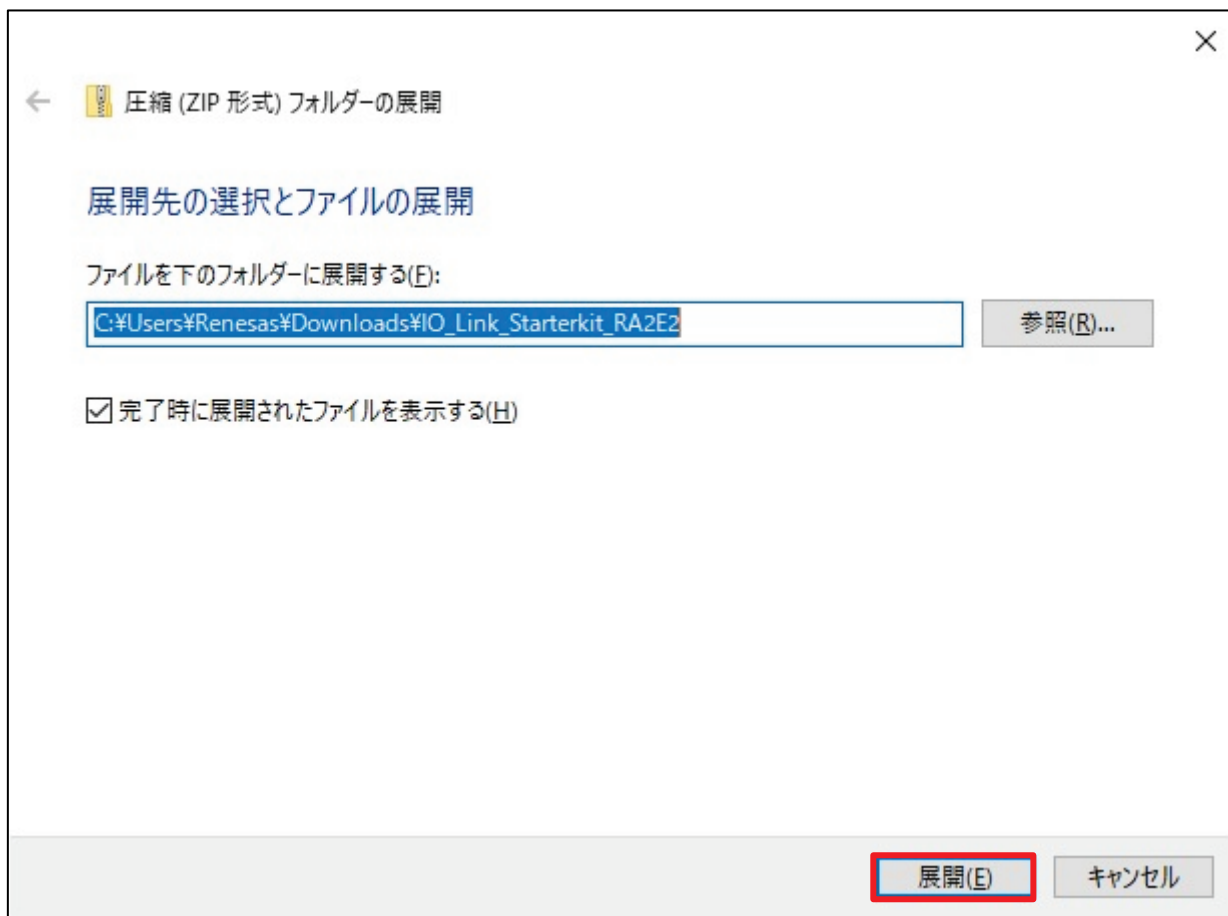


図 7-9 サンプルプロジェクト zip ファイルの解凍



2. IODD DTM Configurator を起動します。



図 7-10 IODD DTM Configurator

3. 「Add IODD」 をクリックします。

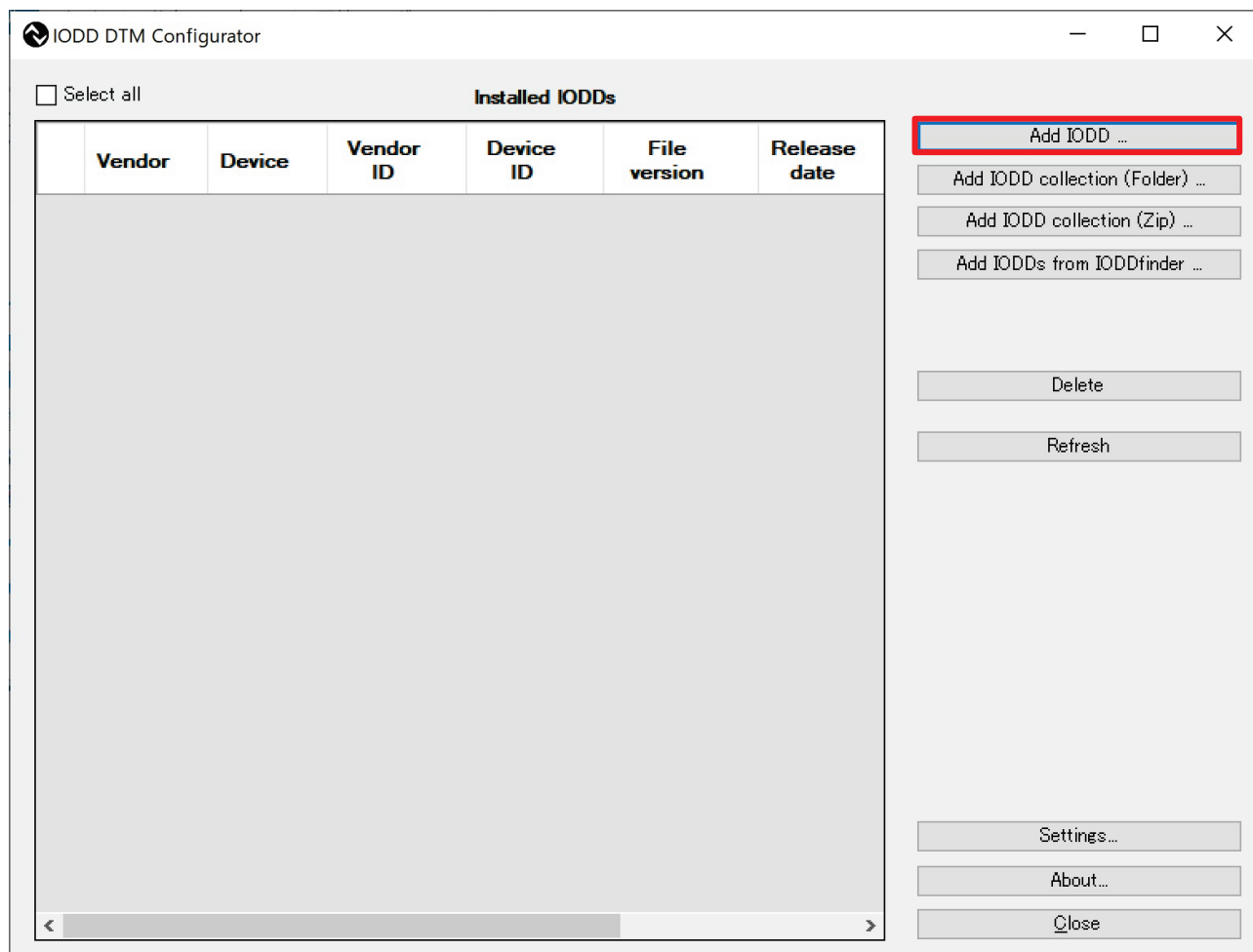


図 7-11 IODD DTM Configurator

4. 解凍した zip ファイルの IO-Link\IODD フォルダから IODD ファイルを選択し、「開く」をクリックします。

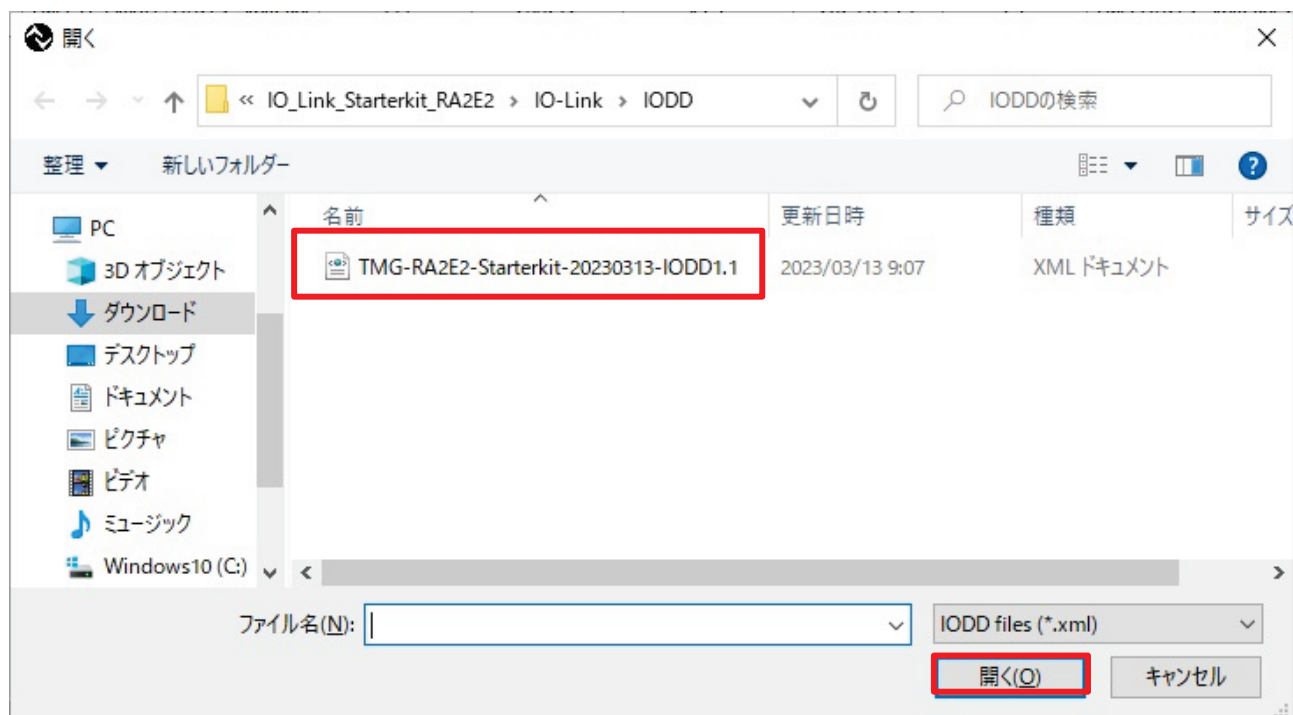


図 7-12 IODD ファイルの選択

以下が表示された場合、FDT frame アプリケーションが起動していない事を確認して「OK」をクリックします。

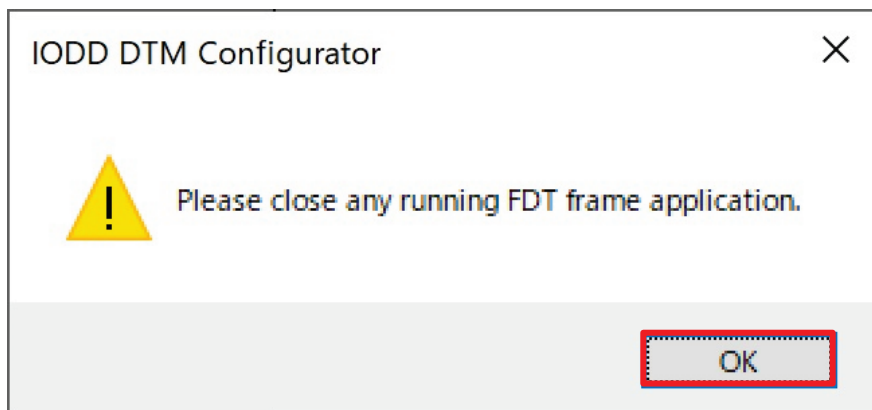


図 7-13 close FDT frame application

5. IODD が追加されたことを確認し、「Close」をクリックします。

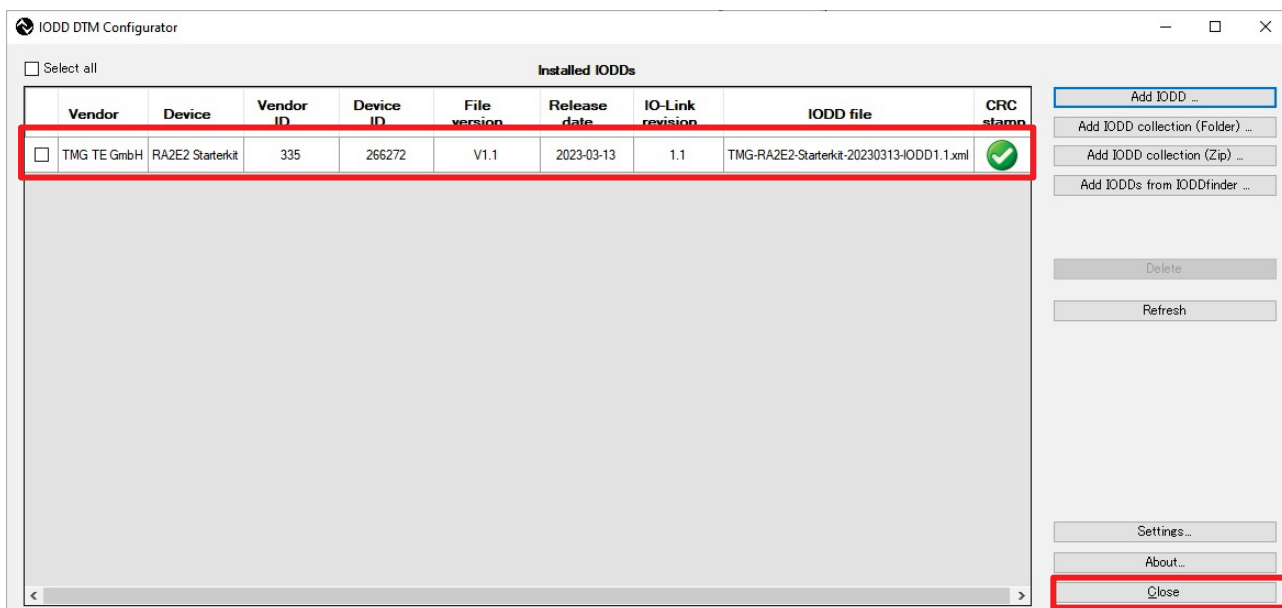


図 7-14 Installed IODDs

6. 以下が表示されたら「OK」をクリックします。

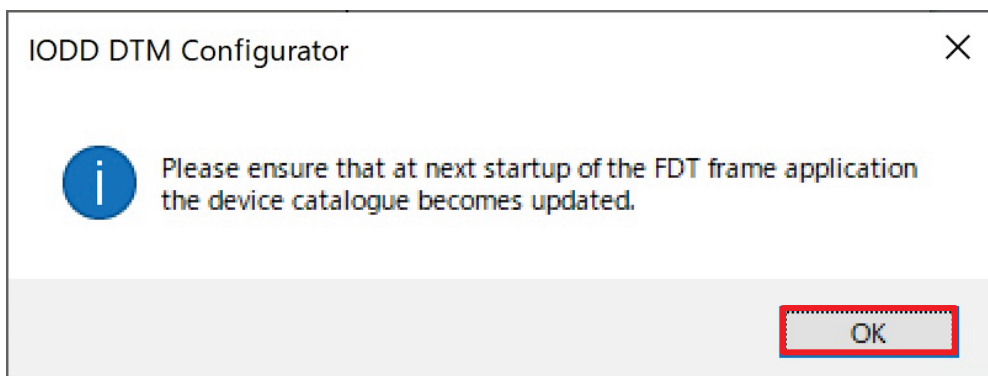


図 7-15 information

## 7.4 PACTware DC 操作方法

PACTware DC の操作方法について解説します。

1. PACTware DC 6.0 を起動します。



図 7-16 PACTware DC 6.0

2. 次のメッセージが表示された場合は「Yes」をクリックします。

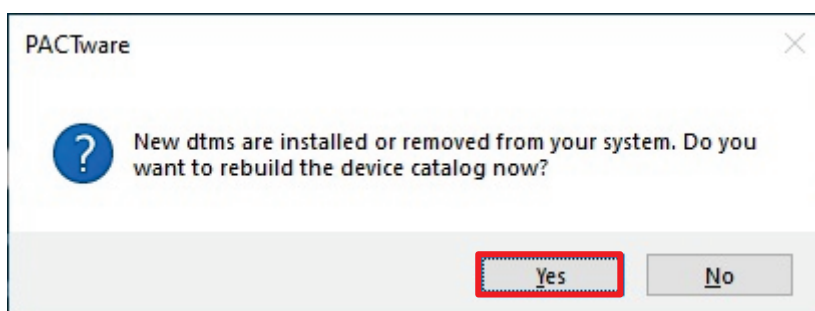


図 7-17 rebuild the device catalog

3. PACTware DC が起動されたら「Search new device」をクリックします。

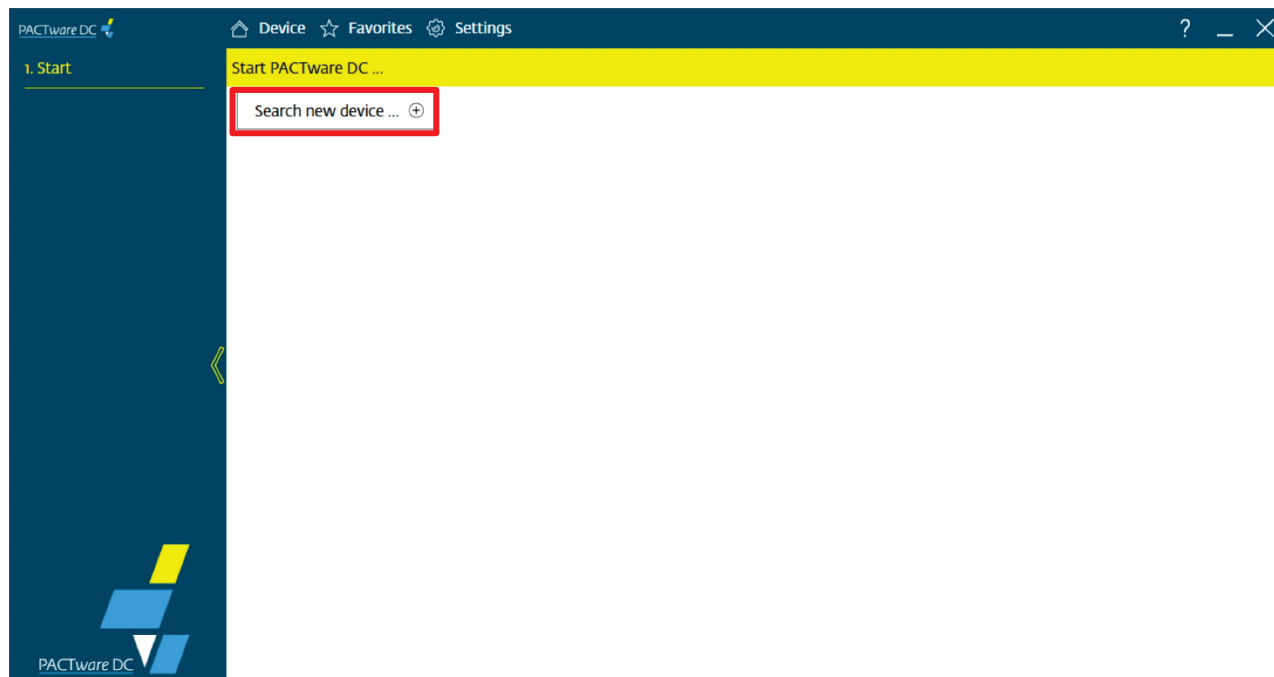


図 7-18 PACTware DC 初回起動時の画面

4. 検索された USB マスターが表示されます。「IO-Link USB Master 2.0」をクリックします。

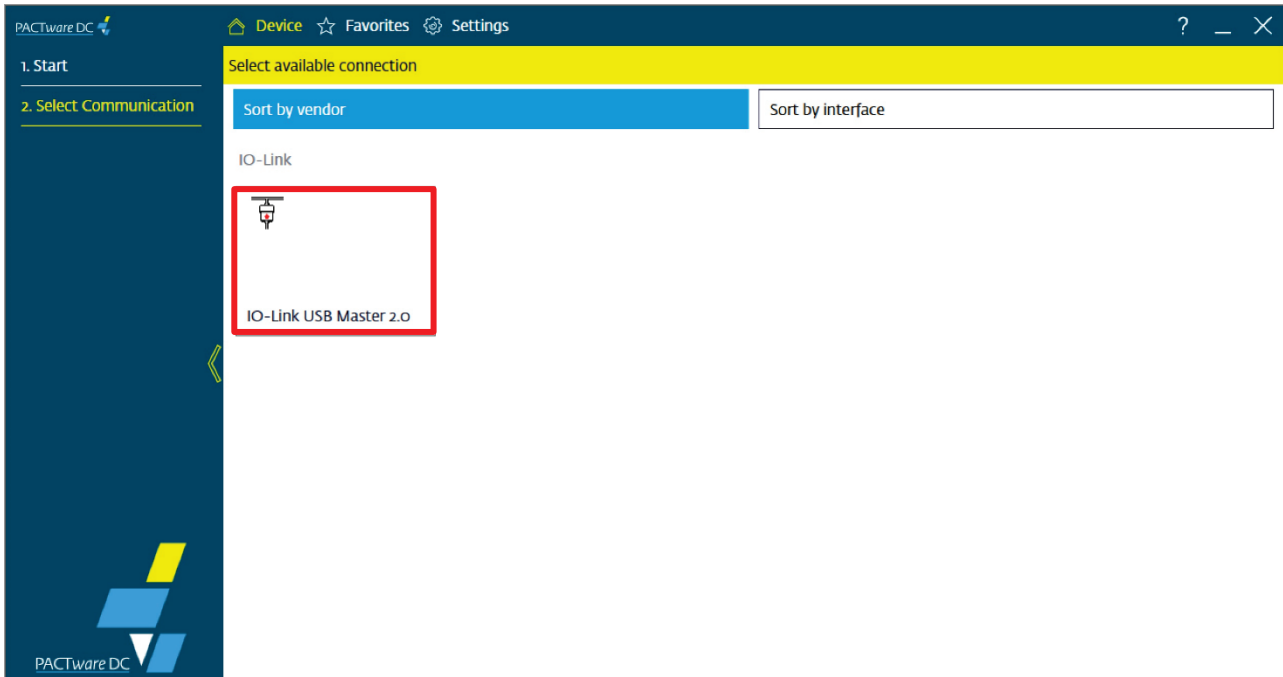


図 7-19 IO-Link USB Master 2.0

5. IO-Link マスターと通信が行われ画面が切り替わります。「Read from device (Upload)?」ダイアログが表示されたら「Yes」をクリックします。

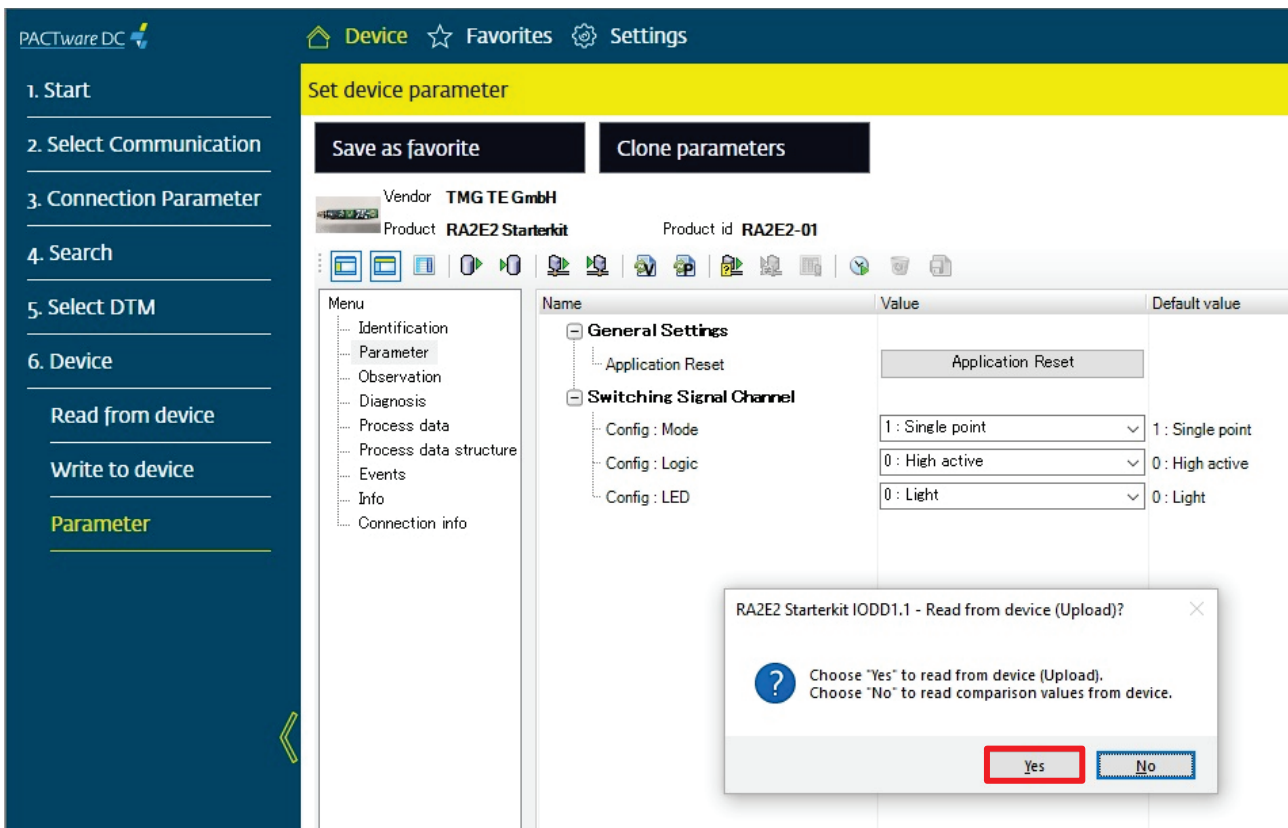


図 7-20 Read from device (Upload)?

## 7.5 PACTware DC 機能説明

Inductive Sensor Board を使用する為に必要な機能の説明を行います。

### 7.5.1 Identification

Identification メニューでは、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

Menu	Name	Value	Default value
Identification	Vendor Name	TMG TE GmbH	TMG TE GmbH
Parameter	Vendor Text	www.tmgte.com	www.tmgte.com
Observation	Product Name	RA2E2 Starterkit	RA2E2 Starterkit
Diagnosis	Product Text	IO-Link Starterkit with RA2E2 Proces	IO-Link Starterkitwith RA2...
Process data	Product ID	RA2E2-01	RA2E2-01
Process data structure	Serial Number	123456789	
Events	Hardware Revision	1.0	
Info	Firmware Revision	1.0	
Connection info	Application-specific Tag	***	---
	Function Tag	***	---
	Location Tag	***	---

図 7-21 Identification

## 7.5.2 Parameter

Parameter メニューには、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。

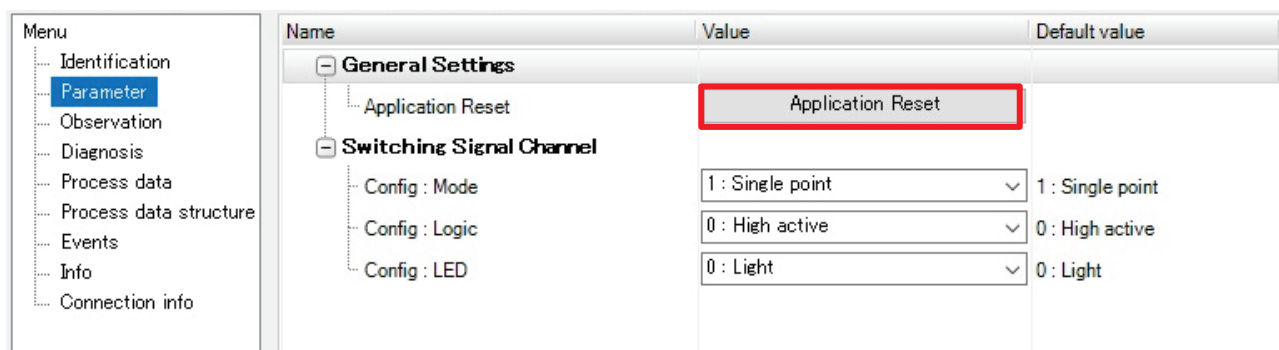


図 7-22 Parameter

### 7.5.2.1 パラメータをデフォルト値にリセット

図 7-22 の「Application Reset」ボタンをクリックすることで、デバイスの設定をデフォルト値に戻すことができます。

1. 「Application Reset」をクリックします。
2. 次のメッセージが表示されたら「OK」をクリックします。

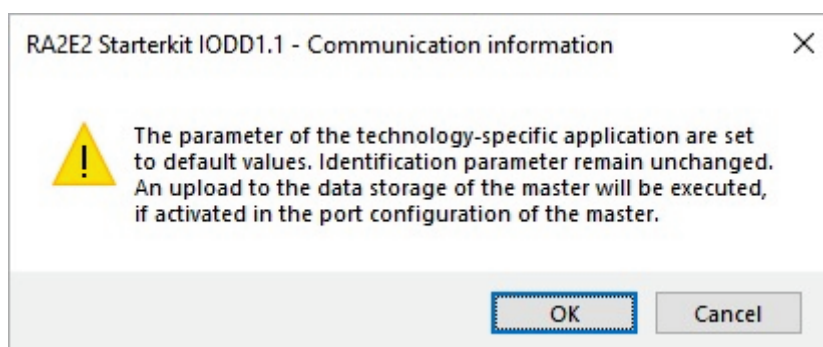


図 7-23 確認メッセージ

3. Application Reset が実施され次のメッセージが表示されます。確認して「OK」をクリックします。



図 7-24 実施後 Communication information

**【注】** この時点ではリセットされたデフォルト値は PACTware DC に読み込まれていません。次の操作でデバイスからの読み込みを実施する必要があります。

4. 「Read from device(Upload)」 ボタンをクリックします。

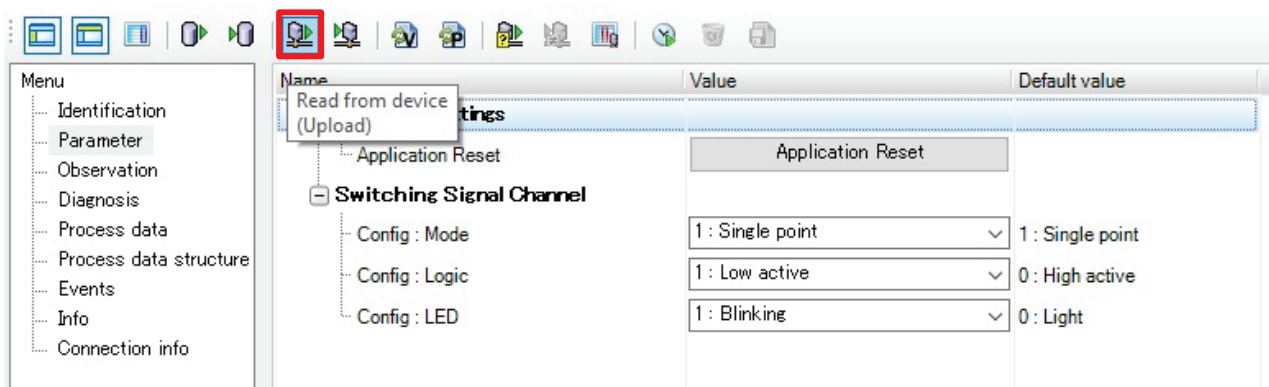


図 7-25 Read from device(Upload)

5. デバイスからの読み込みが実施され、表示内容がデフォルト値に更新されます。

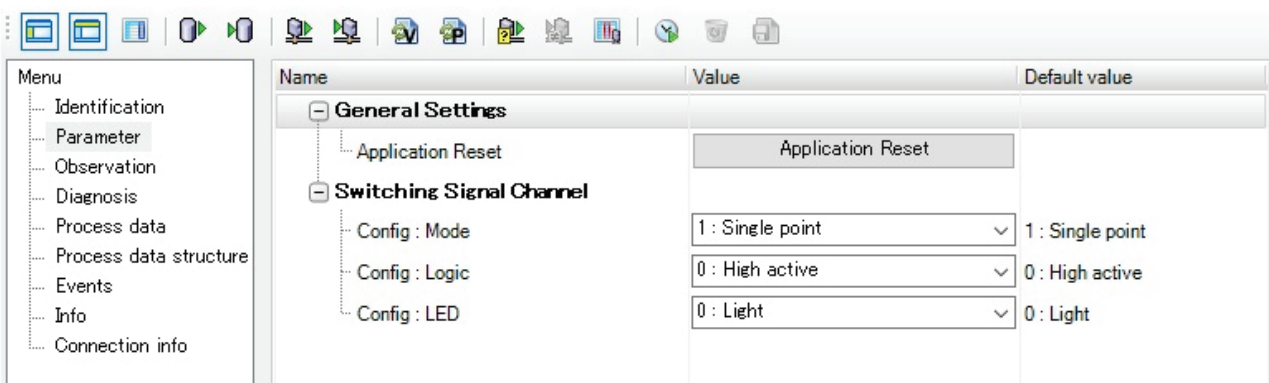


図 7-26 デフォルト値



## 7.5.2.2 Config.Mode 設定

初期値として「1:Single point」が設定されています。基本的に設定を変更する必要はありません。スイッチング機能を無効化する場合は、「0:Deactivated」に変更します。

1. Config : Mode のリストで 「1:Single point」 または 「0:Deactivated」 を選択します。
2. 「Write to device(Download)」 ボタンをクリックしてデバイスに設定します。

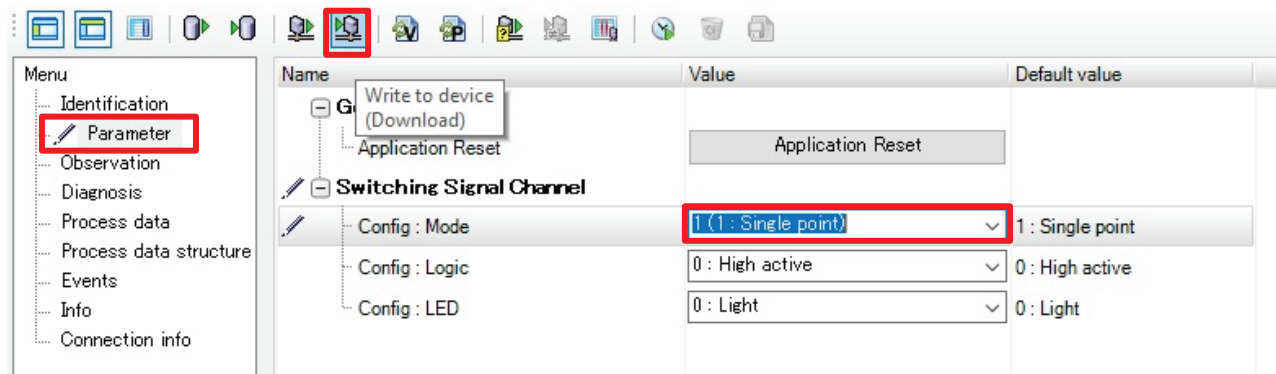


図 7-27 Config.Mode 設定

## 7.5.2.3 Config.Logic 設定

LED 点灯の論理を設定します。初期値として「0:High active」が設定されています。「1:Low active」に設定すると LED 点灯の論理が反転します。

Config.Logic については「5.4 スwitching状態と LED 動作イメージ」を参照して下さい

1. Config : Logic のリストで 「0:High active」 または 「1:Low active」 を選択します。
2. 「Write to device(Download)」 ボタンをクリックしてデバイスに設定します。

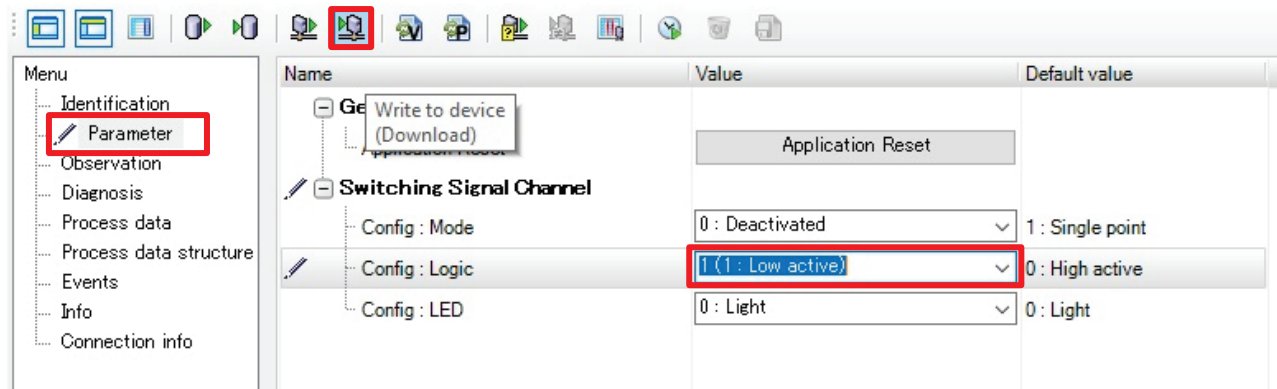


図 7-28 Config.Logic 設定

## 7.5.2.4 Config.LED 設定

LED の点灯/点滅状態を設定します。初期値として「0:Light」が設定されています。「1:Blinking」を設定すると LED は点滅モードとなります。

Config.LED については「5.4 スwitching状態と LED 動作イメージ」を参照して下さい

1. Config : LED のリストで 「0:Light」または 「1:Blinking」 を選択します。
2. 「Write to device(Download)」 ボタンをクリックしてデバイスに設定します。

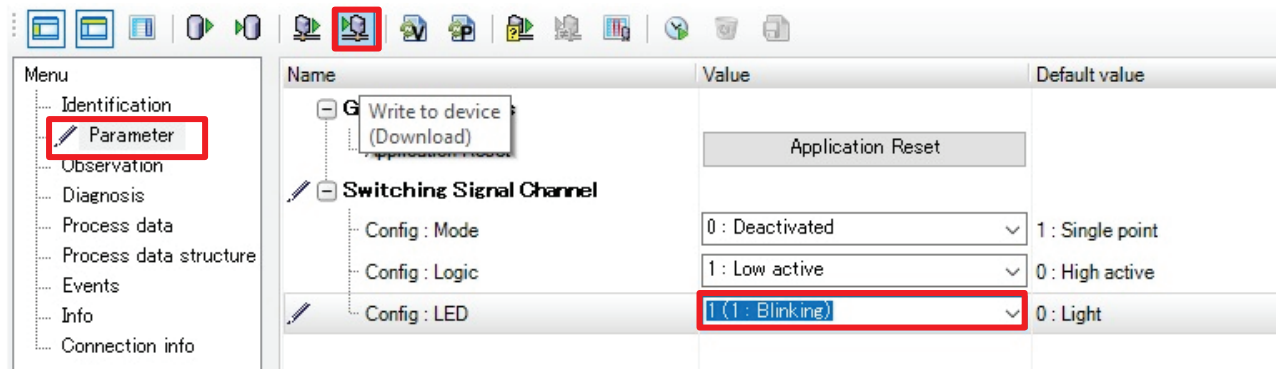


図 7-29 Config.LED 設定

### 7.5.3 Process data

Process data メニューには、PD Input の情報が表示されます。

#### PD Input

- Switching Signal  
スイッチング状態 (Low or High)

1. “P”と表示されている「Enable or disable cyclic read from device for process data」ボタンをクリックし、デバイスからの周期的なデータ取得を有効にします。
2. PD Input: Switching Signal に、金属非検出時は Low、検出時は High で表示されます。

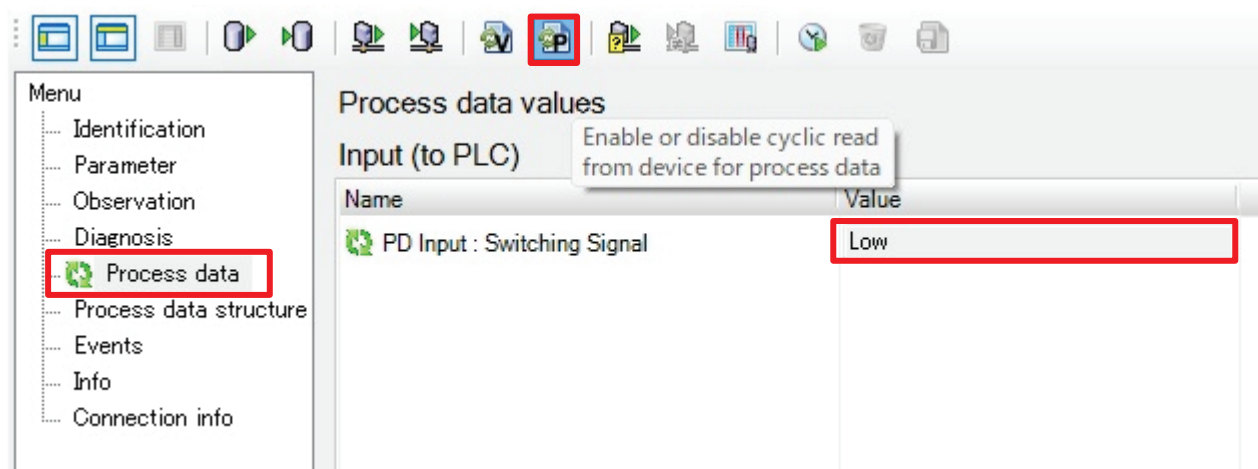


図 7-30 Process data

## 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 GUI Tool の「7.5.3 Process data」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 default 設定での動作

本章では、default 設定での動作例を示します。

設定フィールド	設定値	概要
Config.Mode	Single point	動作有効
Config.Logic	High active	金属検出時 LED ON
Config.LED	Light	LED 点灯

- 「7.5.2.1 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。
- Sensing Coil に金属が接近していない状態では、LED は消灯しています。

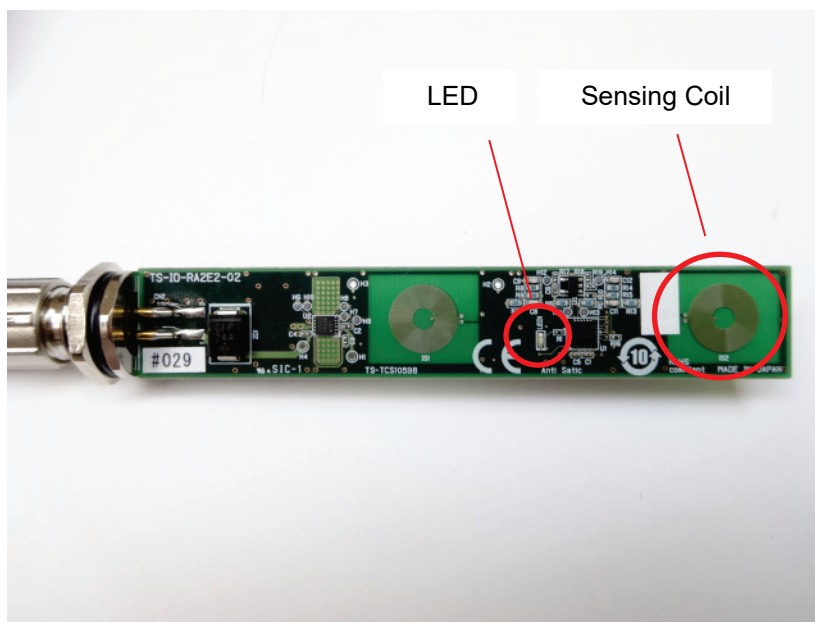


図 8-1 金属 - 非検出時

- Sensing Coil 上空の 5mm 付近に金属を近づけます。金属接近を検出すると LED が点灯します。

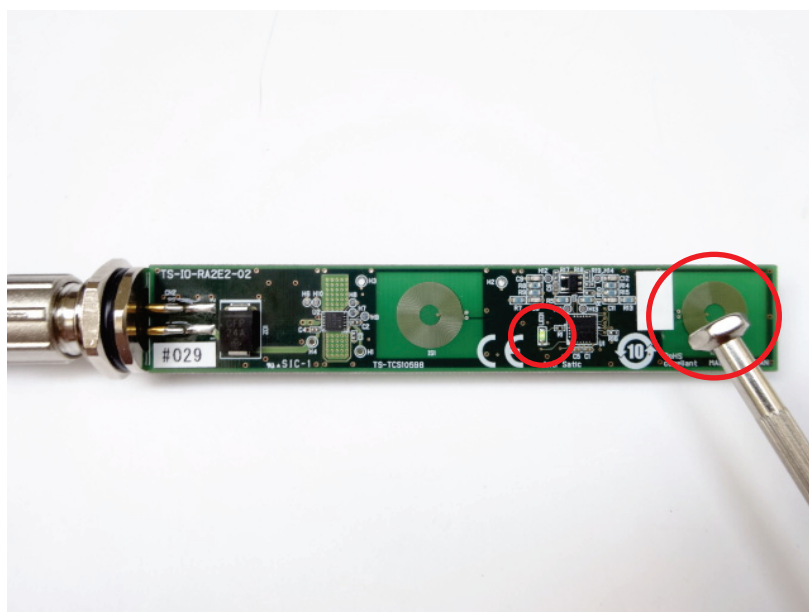


図 8-2 金属 - 検出時

## 8.2 設定変更での動作

設定を変更した場合の動作例を示します。以下の設定で行います。

設定フィールド	設定値	概要
Config.Mode	Single point	動作有効
Config.Logic	Low active	金属検出時 LED OFF
Config.LED	Blinking	LED 点滅

- 「7.5.2.3 Config.Logic 設定」と「7.5.2.4 Config.LED 設定」に従って、設定値を変更します。
- Sensing Coil に金属が接近していない状態で、LED は点滅しています。

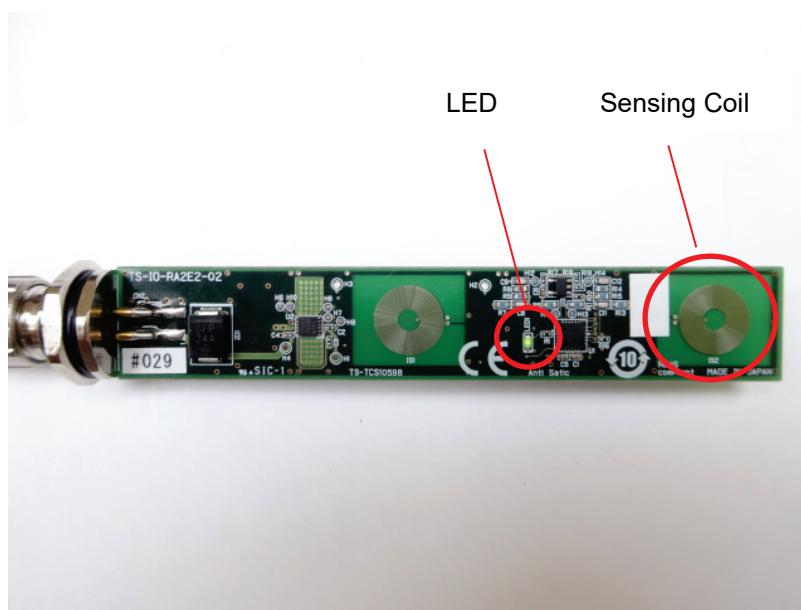


図 8-3 金属 - 非検出時

- Sensing Coil 上空の 5mm 付近に金属を近づけます。金属接近を検出すると LED が消灯します。

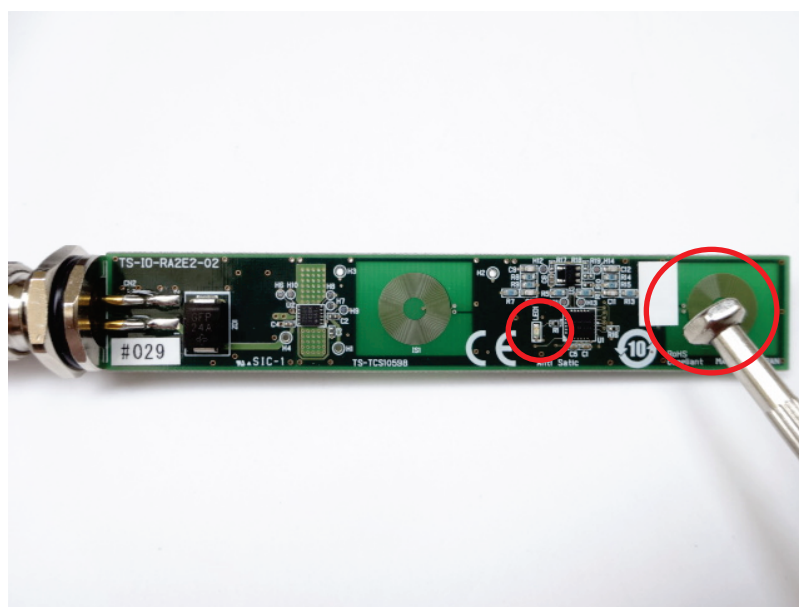


図 8-4 金属 - 検出時

## 改訂記録

Ver.	発行日	改訂内容	
		ページ	ポイント
1.00	Sep.28.23	-	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。