

---

# RA ファミリ、RX ファミリ、RL78 ファミリ

## RRH46410 サンプルソフトウェアマニュアル

---

### 要旨

本アプリケーションノートでは、RA ファミリ、RX ファミリ、RL78 ファミリで動作する RRH46410 ガスセンサのサンプルソフトウェアについて説明します。

### 動作確認デバイス

RA2E1 グループ

RA0E1 グループ

RX140 グループ

RL78/G23 グループ

### 動作確認センサボード

Digital Gas Sensor Module for Indoor Air Quality Applications (QCIOT-RRH46410POCZ)

本アプリケーションノート記載の設定例は、上記のセンサボードを使用する場合の例です。

そのため、組み込み対象回路に合わせて、以下の設定を見直す必要があります。

- 割り込み信号回路：「6.5 割り込み信号回路に関する注意」を参照してください。
- RESET 信号回路：「6.6 RESET 信号回路に関する注意」を参照してください。

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、十分評価してください。

## 目次

1. 概要	4
1.1 用語／略語	4
2. 動作確認環境	5
2.1 RA 動作確認環境	5
2.2 RX 動作確認環境	8
2.3 RL78 動作確認環境	9
3. RRH46410 センサ仕様	10
3.1 センサ仕様概要	10
3.2 センサの機能と方式	10
3.2.1 出力データの変換 (ファームウェア / API / アルゴリズム)	10
4. サンプルソフトウェア仕様	11
4.1 サンプルソフトウェア構成	11
4.2 Sensor Control Module API 関数仕様	11
4.2.1 Sensor Control Module API 関数一覧	11
4.2.2 API 使用ガイド	12
4.3 サンプルソフトウェア Non-OS 版メイン処理フロー	14
4.4 サンプルソフトウェア OS 版メイン処理フロー	16
5. Configuration 設定	18
5.1 RRH46410 Sensor Control Module 設定	18
5.1.1 RA ファミリ	18
5.1.2 RX ファミリ	19
5.1.3 RL78 ファミリ	20
5.2 I2C Communication Middleware (COMMS_I2C)設定	21
5.2.1 RA ファミリ	21
5.2.2 RX ファミリ	22
5.2.3 RL78 ファミリ	23
5.3 I2C ドライバ設定	24
5.3.1 RA ファミリ	24
5.3.2 RX ファミリ	27
5.3.3 RL78 ファミリ	30
5.4 IRQ ドライバ設定	31
5.4.1 RA ファミリ	31
5.4.2 RX ファミリ	32
5.4.3 RL78 ファミリ	32
6. デバイス変更ガイド	33
6.1 サンプルプロジェクトのインポート	33
6.2 RA サンプルプロジェクト	35
6.2.1 FSP Configurator の設定変更	35
6.2.2 サンプルソースの変更	45
6.2.3 IRQ を使用しない場合の変更	45

6.2.4 ツールチェイン設定変更 .....	45
6.3 RX サンプルプロジェクト .....	46
6.3.1 Smart Configurator 設定の変更 .....	46
6.3.2 サンプルソースの変更 .....	55
6.3.3 ツールチェイン設定変更 .....	55
6.3.4 GCC ビルド時の注意 .....	55
6.3.5 IAR 社統合開発環境「IAR Embedded Workbench」利用時 .....	55
6.4 RL78 サンプルプロジェクト .....	56
6.4.1 Smart Configurator 設定の変更 .....	56
6.4.2 生成コードの変更 .....	66
6.4.3 サンプルソースの変更 .....	67
6.4.4 ツールチェイン設定変更 .....	68
6.4.5 LLVM ビルド時の注意 .....	70
6.5 割り込み信号回路に関する注意 .....	71
6.6 RESET 信号回路に関する注意 .....	72
6.7 Renesas Pmod Sensor Board のデジ・チェーン接続時のプルアップ抵抗処理 .....	72
7. ガスデータの確認方法 .....	73
改訂記録 .....	75

## 商標

FreeRTOS™ は Amazon Web Services, Inc. の商標です。

Pmod™ は Digilent Inc. の商標です。

## 1. 概要

本ソフトウェアは、RRH46410 ガスセンサによるデータの取得および、演算を行うためのサンプルプログラムです。MCUに内蔵されているI2Cを用いFSP/FITやコードジェネレータのI2Cドライバとの組み合わせによってRRH46410の測定、データ取得、および測定結果の演算を行います。

### 1.1 用語／略語

用語とその略語を以下に示します。

表 1-1 用語／略語一覧

用語	略語
RRH46410 Sensor Control Module	Sensor Control Module RA MCU の場合、“rm_rrh46410” RX MCU の場合、“r_rrh46410_rx” RL78 MCU の場合、“r_rrh46410”
I2C Communication Middleware	COMMS_I2C RA MCU の場合、“rm_comms_i2c” RX MCU の場合、“r_comms_i2c_rx” RL78 MCU の場合、“r_comms_i2c”
I2C ドライバ	RA MCU の場合、“r_iic_master”, “r_sci_i2c”, “r_iica_master” RX MCU の場合、“r_riic_rx”, “r_sci_iic_rx” RL78 MCU の場合、“r_iica_master”
IRQ ドライバ	RA MCU の場合、“r_icu” RX MCU の場合、“r_irq_rx” RL78 MCU の場合、“割り込みコントローラ”
Serial Communications Interface	RA MCU の場合、“SCI”もしくは“SCI I/F” RX MCU の場合、“SCI”もしくは“SCI I/F”
Serial Array Unit	RA MCU の場合、“SAU”もしくは“SAU I/F” RL78 MCU の場合、“SAU”もしくは“SAU I/F”
I2C Bus Interface	RA MCU の場合、“IIC”もしくは“IIC I/F” RX MCU の場合、“RIIC”もしくは“RIIC I/F”
I2C Bus Interface (IICA)	RA MCU の場合、“IICA”もしくは“IICA I/F”
Serial Interface IICA	RL78 MCU の場合、“IICA”もしくは“IICA I/F”
I2C Bus Interface、 I2C Bus Interface (IICA)、 Serial Interface (IICA)の総称	“I2C I/F”
割り込みコントローラの総称	“ICU I/F” (Interrupt Controller Unit)
汎用 I/O ポート	“GPIO”もしくは“GPIO I/F”
Renesas Pmod Type 6A Sensor Pmod Board の 1 ピン(#1)	“IRQ#” (割り込み発生時に L 出力)

## 2. 動作確認環境

### 2.1 RA 動作確認環境

本ソフトウェアの RA 動作確認環境を以下に示します。

#### (1) Evaluation Kit for RA2E1 (EK-RA2E1)

表 2-1 EK-RA2E1 動作確認環境

項目	内容
デモボード	RTK7EKA2E1S00001BE (EK-RA2E1)
使用マイコン	RA2E1 (R7FA2E1A92DFM:64pin)
動作周波数	48MHz
動作電圧	5V
統合開発環境	Renesas Electronics e <sup>2</sup> studio 2024-07
C コンパイラ	GNU ARM Embedded 13.2.1.arm-13-7
オプション 設定	コンパイラのデフォルト設定に対して、以下の設定を追加： ISO C99 (-std = c99)、最適化レベル：デフォルト設定 (-O2)
FSP	v5.4.0
RTOS	FreeRTOS v10.6.1
エミュレータ	On board (J-LINK)
変換ボード	Interposer Board for Pmod Type2/3 to 6A (US082-INTERPEVZ)
センサボード	Digital Gas Sensor Module for Indoor Air Quality Applications (QCIOT-RRH46410POCZ)

表 2-2 RA2E1 使用メモリ量（最大 ROM サイズの動作モード設定時）

動作モード	領域	サイズ (Non-OS) [Bytes]	サイズ (FreeRTOS) [Bytes]
IAQ 2nd Gen	ROM	2,372	4,240 (注 1)
	RAM	288	500

注 メモリサイズはサンプルコードと RRH46410 Sensor Control Module、COMMS\_I2C を対象に計算しています。FreeRTOS の場合は、スレッドのメモリサイズは計算に含めていません。

注 1 Relax 機能による 1,554 バイト増加分を含みます。

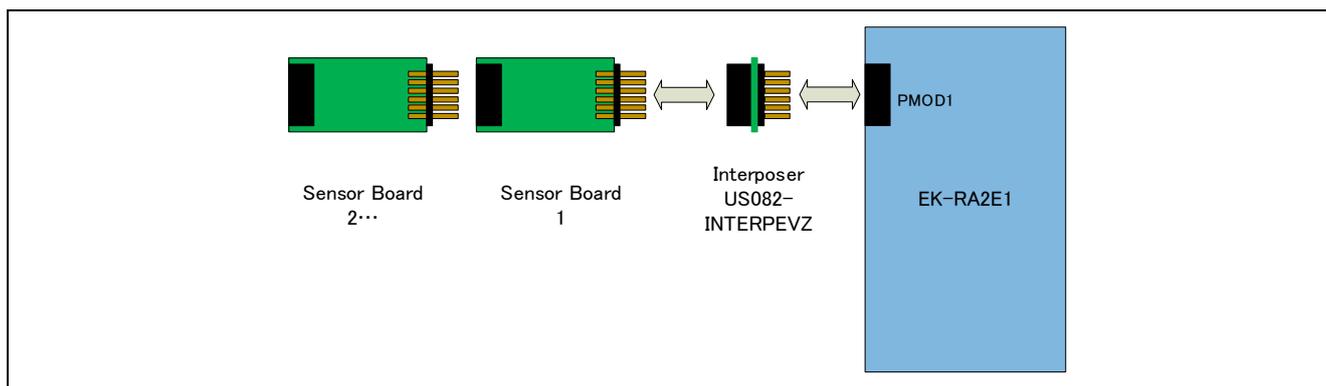


図 2-1 EK-RA2E1 HW 接続図

## (2) RA0E1 Fast Prototyping Board (FPB-RA0E1)

表 2-3 FPB-RA0E1 動作確認環境

項目	内容
デモボード	RTK7FPA0E1S00001BJ (FPB-RA0E1)
使用マイコン	RA0E1 (R7FA0E1073CFJ:32pin)
動作周波数	32MHz
動作電圧	5V
統合開発環境	Renesas Electronics e <sup>2</sup> studio 2024-07
C コンパイラ	GNU ARM Embedded 13.2.1.arm-13-7
オプション 設定	コンパイラのデフォルト設定に対して、以下の設定を追加： ISO C99 (-std = c99)、最適化レベル：デフォルト設定 (-Oz)
FSP	v5.4.0
エミュレータ	On board (J-LINK)
変換ボード	Interposer Board for Pmod Type2/3 to 6A (US082-INTERPEVZ)
センサボード	Digital Gas Sensor Module for Indoor Air Quality Applications (QCIOT-RRH46410POCZ)

表 2-4 RA0E1 使用メモリ量（最大 ROM サイズの動作モード設定時）

動作モード	領域	サイズ (Non-OS) [Bytes]
IAQ 2nd Gen	ROM	2,066
	RAM	276

注 メモリサイズはサンプルコードと RRH46410 Sensor Control Module、COMMS\_I2C を対象に計算しています。

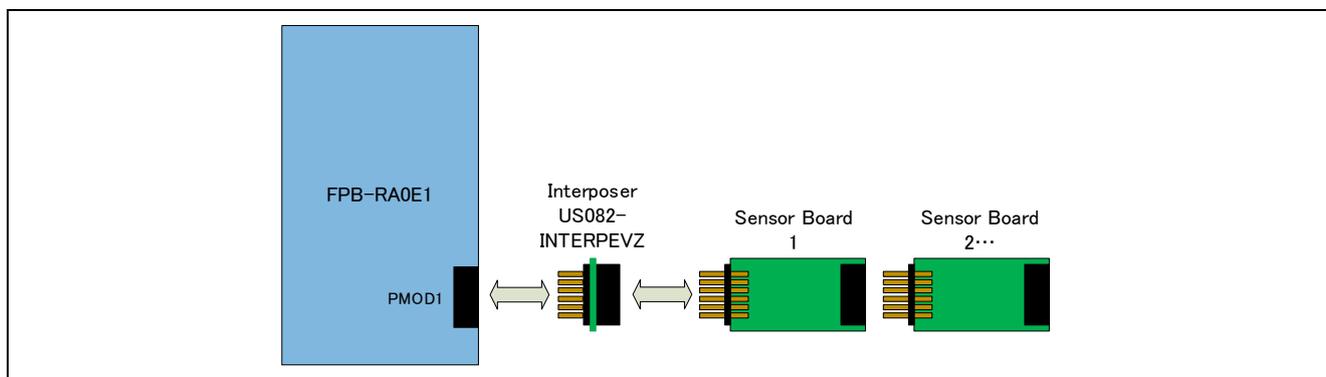


図 2-2 FPB-RA0E1 HW 接続図

(3) Interposer Board 利用について

Interposer Board は、SCI I/F の Pmod Type 2A/Type 3A コネクタに対して、簡易 IIC 機能に切り替えることで、Pmod Type 6A センサを接続するための I/F 変換ボードです。

そのため、SAU I/F の Pmod Type 2A/Type 3A コネクタに対しては、適用できません。ただし、IICA I/F に切り替えることで使用できる場合があります。MCU のハードウェアマニュアルを参照してください。

表 2-5 Pmod I/F と Serial I/F と Interposer Board 有無による動作可否

Pmod I/F	接続先の MCU Serial I/F	動作可否
Type 2A、 Type 3A	SCI I/F、 IICA I/F (注 1)	Interposer Board を利用すると動作します。(注 2)
	SAU I/F	Interposer Board 有無に関わらず、動作しません。
Type 6、 Type 6A	SCI I/F、IIC I/F SAU I/F、IICA I/F	Interposer Board 無しで動作します。(注 2)

注 1 : SAU I/F 用に設けられた端子ですが、マルチファンクション端子割り当てにより、IICA 端子に切り替え可能な場合に適用できます。切り替え可能な場合の信号接続を以下に示します。

Pmod Pin	Type 2A /Type 3A	接続先 SAU I/F ICU I/F GPIO I/F	Multi- Function IICA I/F 切り替え	Interposer Board	Renesas Pmod Type 6A Sensor Board
#1	CS/CTS	<b>GPIO</b>		↔	<b>IRQ# (注 3)</b>
#2	MOSI/TXD	SAU TXD	<b>SDAA</b>	↔	<b>RESET#</b>
#3	MISO/RXD	SAU RXD	<b>SCLA</b>	↔	<b>IIC_SCL</b>
#4	SCK/RTS	<b>GPIO</b>		↔	<b>IIC_SDA</b>
#7	INT	<b>IRQ#</b>		↔	<b>BUSY#</b>
#8	RESET	<b>GPIO</b>		↔	<b>ENABLE</b>
#9	CS2/GPIO	<b>GPIO</b>		↔	<b>POWER_ON</b>
#10	CS3/GPIO	<b>GPIO</b>		↔	<b>GPIO</b>

注 2 : IRQ 信号を利用する場合、Pmod #1 が MCU の IRQ 信号に接続されていることを確認してください。

注 3 : 割り込み信号回路に関して、「6.5 割り込み信号回路に関する注意」を参照してください。

適用例 : FPB-RA0E1 Pmod1 が該当します。

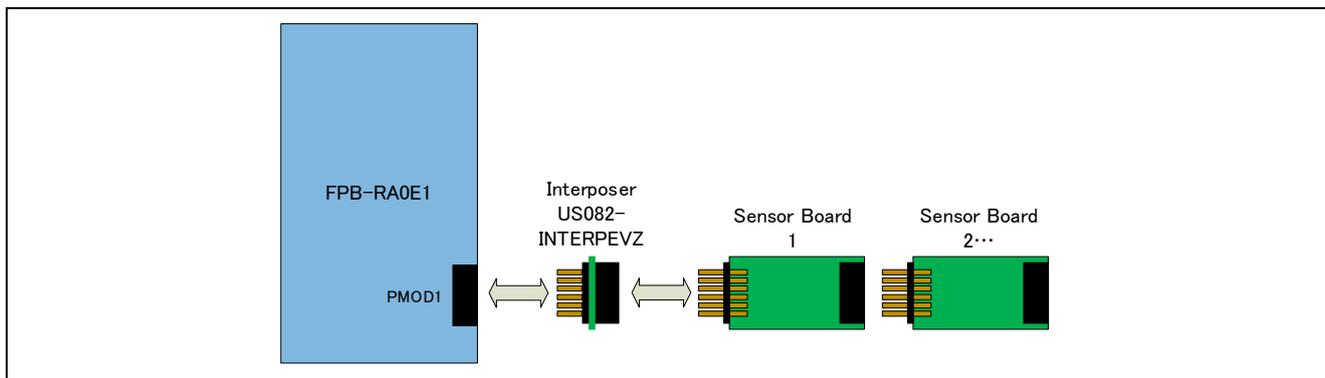


図 2-3 FPB-RA0E1 の Pmod1 Type 2A/Type 3A で IICA を使用した場合の接続図

## 2.2 RX 動作確認環境

本ソフトウェアの RX 動作確認環境を以下に示します。

### (1) Fast Prototyping Board for RX140 (FPB-RX140)

表 2-6 FPB-RX140 動作確認環境

項目	内容
デモボード	RTK5FP1400S00001BE (FPB-RX140)
使用マイコン	RX140 (R5F51406BGFN: 80pin)
動作周波数	48MHz
動作電圧	5V
統合開発環境	Renesas Electronics e <sup>2</sup> studio 2024-10
C コンパイラ	Renesas Electronics CC-RX V.3.06.00 GCC for Renesas RX 8.3.0.202405
オプション 設定	コンパイラのデフォルト設定に対して、以下の設定を追加： CC-RX : C99 言語規格(-lang = c99)、最適化レベル デフォルト設定 (レベル 2) GCC : ISO C99 (-std = c99)、最適化レベル デフォルト設定 (-Og)
FIT (RX Driver Package v1.45)	Board Support Packages (r_bsp) v7.51 GPIO Driver (r_gpio_rx) v5.10 RRH46410 Sensor Control Module (r_rrh46410_rx) v1.00 IIC Communication Middleware (r_comms_i2c_rx) v1.22 RIIC Multi Master I2C Driver (r_riic_rx) v3.00 Simple IIC Driver (r_sci_iic_rx) v2.80 IRQ Driver (r_irq_rx) v4.50 CMT Driver (r_cmt_rx) v5.70
RTOS	FreeRTOS Kernal 10.4.3-rx-1.0.9、FreeRTOS Object 10.4.3-rx-1.0.9
エミュレータ	On board (E2OB)
変換ボード	Interposer Board for Pmod Type2/3 to 6A (US082-INTERPEVZ)
センサボード	Digital Gas Sensor Module for Indoor Air Quality Applications (QCIOT-RRH46410POCZ)

表 2-7 RX140 使用メモリ量 (最大 ROM サイズの動作モード設定時)

動作モード	領域	サイズ (Non-OS) [Bytes] (CC-RX)	サイズ (FreeRTOS) [Bytes] (CC-RX)
IAQ 2nd Gen	ROM	2,855	3,012
	RAM	313	305

注 メモリサイズはサンプルコードと RRH46410 Sensor Control Module、COMMS\_I2C を対象に計算しています。FreeRTOS の場合は、スレッドのメモリサイズは計算に含めていません。

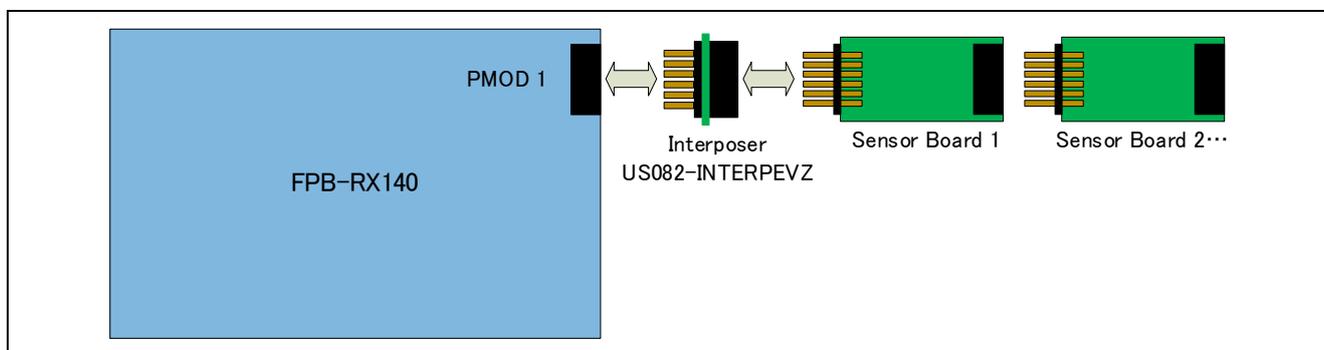


図 2-4 FPB-RX140 HW 接続図

### (2) Interposer Board 利用について

SCI I/F が接続された Pmod Type 2A/Type 3A コネクタに、Interposer Board を追加した場合、Pmod Type 6A Sensor Pmod Board を使用できます。

## 2.3 RL78 動作確認環境

本ソフトウェアの RL78/G23 動作確認環境を以下に示します。

### (1) RL78/G23-128p Fast Prototyping Board (RL78/G23-128p FPB)

表 2-8 RL78/G23-128p FPB 動作確認環境

項目	内容
デモボード	RTK7RLG230CSN000BJ (RL78/G23-128p FPB)
使用マイコン	RL78/G23 (R7F100GSN2DFB :128pin)
動作周波数	32MHz
動作電圧	3.3V
統合開発環境	Renesas Electronics e <sup>2</sup> studio 2024-07
C コンパイラ	Renesas Electronics CC-RL V1.14.00 LLVM for RL78 17.0.1.202409
オプション 設定	コンパイラのデフォルト設定に対して、以下の設定を追加： CC-RL : C99 言語規格(-lang = c99)、最適化レベル デフォルト設定 (-Odefault) LLVM : GNU ISO C99 (-std = gnu99)、最適化レベル デフォルト設定 (-Og)
SIS / CG	Board Support Packages (r_bsp) v1.70 ポート v1.5.0 RRH46410 Sensor Middleware (r_rrh46410) v1.00 IIC Communication Driver Interface Middleware (r_comms_i2c) v1.11 IIC 通信 (マスタモード) v1.6.0 割り込みコントローラ v1.5.0 インターバル・タイマ v1.5.0
エミュレータ	On board (COM Port)
センサボード	Digital Gas Sensor Module for Indoor Air Quality Applications (QCIOT-RRH46410POCZ)

表 2-9 RL78/G23 使用メモリ量 (最大 ROM サイズの動作モード設定時)

動作モード	領域	サイズ (Non-OS) [Bytes] (CC-RL)
IAQ 2nd Gen	ROM	2,986
	RAM	240

注 メモリサイズはサンプルコードと RRH46410 Sensor Control Module、COMMS\_I2C を対象に計算しています。

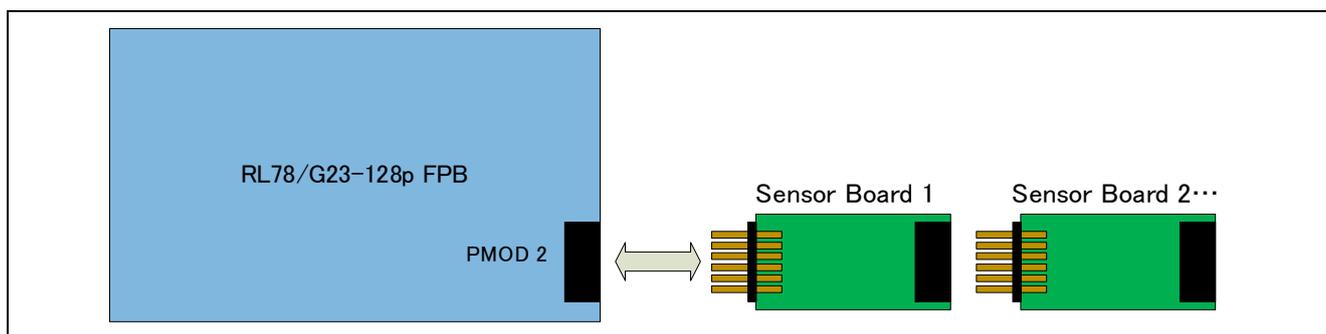


図 2-5 RL78/G23-128p FPB HW 接続図

### (2) Interposer Board 利用について

Interposer Board は、RA/RX 用 SCI I/F の Pmod Type 2A/Type 3A コネクタに対して、簡易 IIC 機能に切り替え、Pmod Type 6A センサを接続するための I/F 変換ボードです。

そのため、SAU I/F が接続された Pmod Type 2A/Type 3A コネクタに、Interposer Board を追加した場合であっても、Pmod Type 6A Sensor Pmod Board を使用できません。

### 3. RRH46410 センサ仕様

#### 3.1 センサ仕様概要

RRH46410 ガスセンサは、室内空気の質に関する、研究と国際規格に基づいて平均的な TVOC 汚染を検出するように設計しています。モジュールの特性を表すパラメータ等、センサモジュールについては、[RRH46410](#) のデータシートを参照してください。

#### 3.2 センサの機能と方式

RRH46410 アーキテクチャは、十分に機械学習を行ったシステムに対して、時間や温度、ガスの特性を用いることで、様々な「動作モード」を可能にしています。また、組み込み AI(人工知能)技術を使用しています。本章では RRH46410 の各動作モードについて説明します。現在、以下の動作モードがリリースされています。

動作モード 1 :	IAQ 2nd Gen	: IAQ、相対的 IAQ、TVOC および eCO <sub>2</sub>
動作モード 2 :	IAQ 2nd Gen ULP	: IAQ、相対的 IAQ、TVOC および eCO <sub>2</sub>
動作モード 3 :	PBAQ	: PBAQ 規格を満たす TVOC の測定

新しく設計する場合は IAQ 2nd Gen (動作モード 1)を使用することをお勧めします。

全ての動作モードに関するその他の技術的な情報は、ルネサスの ZMOD4410 アプリケーションノート (TVOC Sensing)で入手できます。アプリケーションノート、ホワイトペーパー、ブログ、マニュアルなどの詳細については、[RRH46410](#) のウェブページを参照してください。

##### 3.2.1 出力データの変換 (ファームウェア / API / アルゴリズム)

RRH46410 を動作させるために、API、ライブラリ、およびサンプルコードを含むルネサスが提供するファームウェアを使用してください。また、お客様固有のアプリケーションにセンサモジュールを実装するために、プログラミングに関する詳細情報を提供しています。これらのドキュメントは、[RRH46410](#) のウェブページを参照してください。

## 4. サンプルソフトウェア仕様

サンプルソフトウェアパッケージには、以下の 8 プロジェクトが含まれています。

- RA2E1 グループ： Non-OS 版と FreeRTOS 版
- RA0E1 グループ： Non-OS 版
- RX140 グループ： Non-OS 版(CC-RX / GCC)と FreeRTOS 版
- RL78/G23 グループ： Non-OS 版(CC-RL / LLVM)

RX の FreeRTOS の設定方法については [FAQ](#) を参照してください。

### 4.1 サンプルソフトウェア構成

サンプルソフトウェアのレイヤ構成を以下に示します。

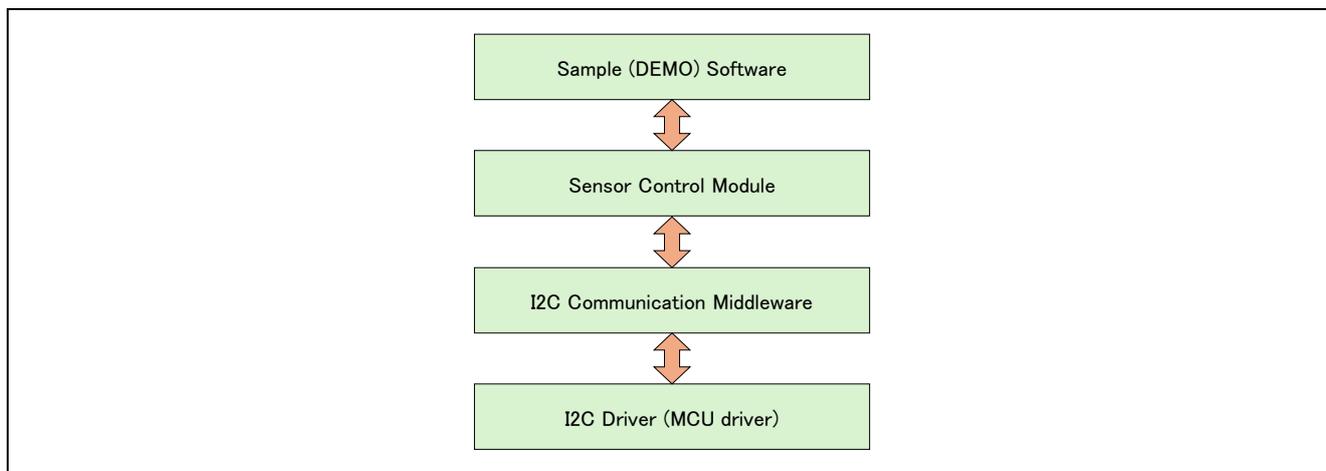


図 4-1 ソフトウェアのレイヤ図

### 4.2 Sensor Control Module API 関数仕様

#### 4.2.1 Sensor Control Module API 関数一覧

Sensor Control module API は以下の関数が含まれます。

関数 API の詳細は以下を参照してください。

RA Flexible Software Package Documentation

Renesas Sensor Control Modules Firmware Integration Technology (R01AN5892)

Renesas Sensor Control Modules Software Integration System (R01AN6192)

表 4-1 Sensor Control Module API 関数一覧

関数	機能
RM_RRH46410_Open()	センサ制御開始処理
RM_RRH46410_Close()	センサ制御終了処理
RM_RRH46410_MeasurementStart()	センサ測定開始処理
RM_RRH46410_MeasurementStop()	センサ測定停止処理
RM_RRH46410_Read()	センサデータ取得処理
RM_RRH46410_TemperatureAndHumiditySet()	湿度の設定処理 (IAQ_2nd_Gen と IAQ_2nd_Gen_ULP で有効)
RM_RRH46410_Iaq2ndGenDataCalculate()	IAQ 2nd Gen センサ測定結果の演算処理
RM_RRH46410_PbaqDataCalculate()	PBAQ センサ測定結果の演算処理

4.2.2 API 使用ガイド

API 関数の使用条件について、想定する関数コールの順番を遷移図として示します。

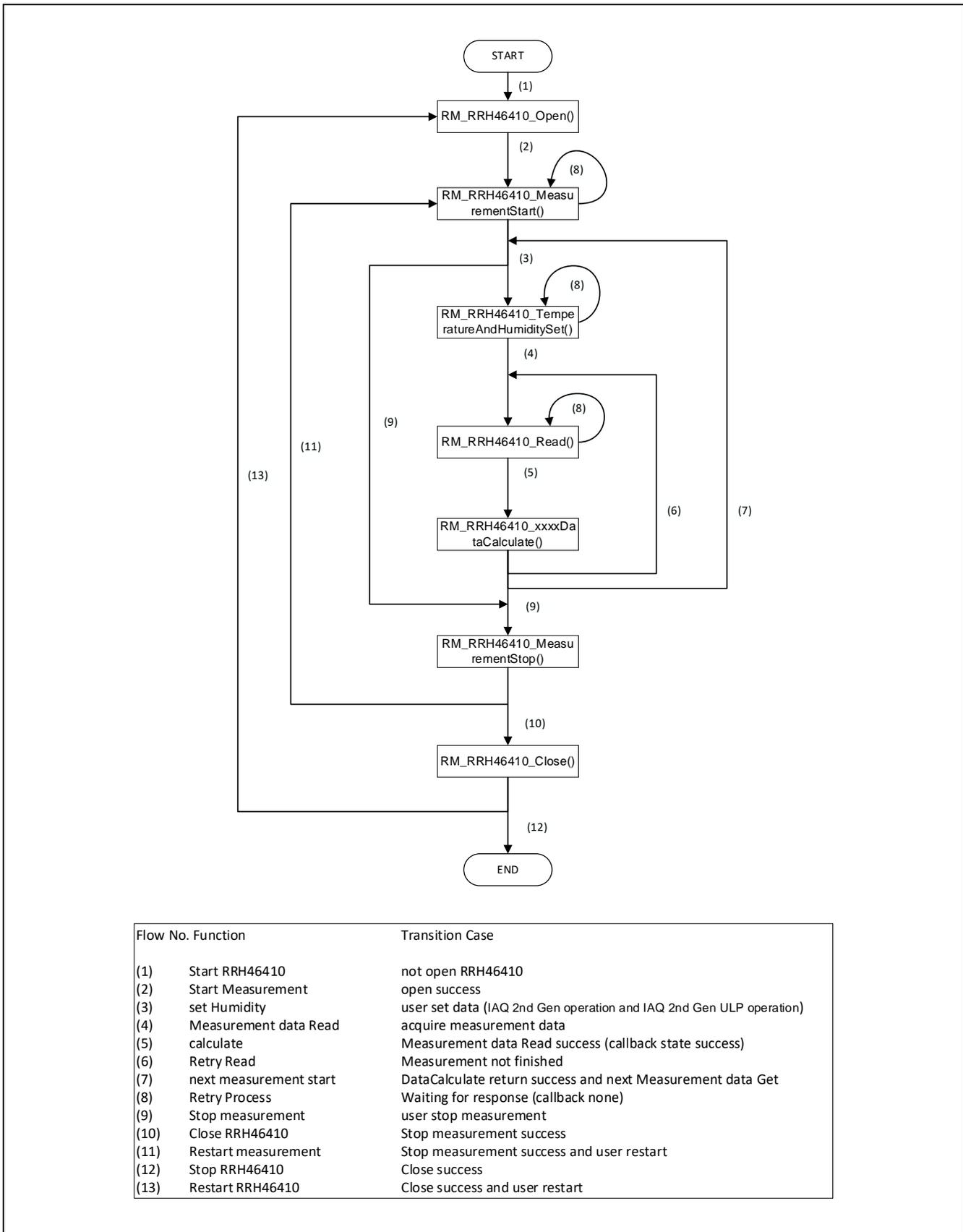


図 4-2 関数 API 遷移図

関数毎の呼び出し条件は以下のとおりです。

- RM\_RRH46410\_Open() : (1) RRH46410 開始時  
(13) RM\_RRH46410\_Close()後の再開始時
- RM\_RRH46410\_Close() : (10) RM\_RRH46410\_MeasurementStop()後
- RM\_RRH46410\_MeasurementStart() : (2) RM\_RRH46410\_Open()後の測定開始時  
(11) RM\_RRH46410\_MeasurementStop()後の測定再開始時  
(8) 測定開始応答待ちによる再試行時
- RM\_RRH46410\_MeasurementStop() : (9) 測定停止時
- RM\_RRH46410\_Read() : (4) 測定データ取得時  
(6) 測定未完了による再試行  
(8) データ取得応答待ちによる再試行
- RM\_RRH46410\_TemperatureAndHumiditySet() : (3) 湿度設定時 (IAQ 2nd Gen と IAQ 2nd Gen ULP の機能、測定精度向上)  
(8) 湿度設定応答待ちによる再試行時
- RM\_RRH46410\_xxxxDataCalculate() : (5) RM\_RRH46410\_Read()後のデータ演算時  
xxxx には各センサ機能名 (Iaq2ndGen、Pbaq) が入ります。

注意 :

RM\_RRH46410\_Open()で I2C ドライバの状態を確認するため、RM\_RRH46410\_Open()処理の前に必ず、I2C ドライバをオープンする必要があります。

I2C ドライバをオープンする方法については、サンプルソフトウェアの g\_comms\_i2c\_bus0\_quick\_setup()関数を参照してください。RL78 は、startup 処理内でオープンされるため必要ありません。

OS 使用時、複数のスレッド/タスクで同時にセンサを制御する場合はユーザによるセマフォを用いた bus 制御が必要となります。

### 4.3 サンプルソフトウェア Non-OS 版メイン処理フロー

サンプルソフトウェアは開始処理を行い、その後はデータ取得、測定結果演算の処理を繰り返します。

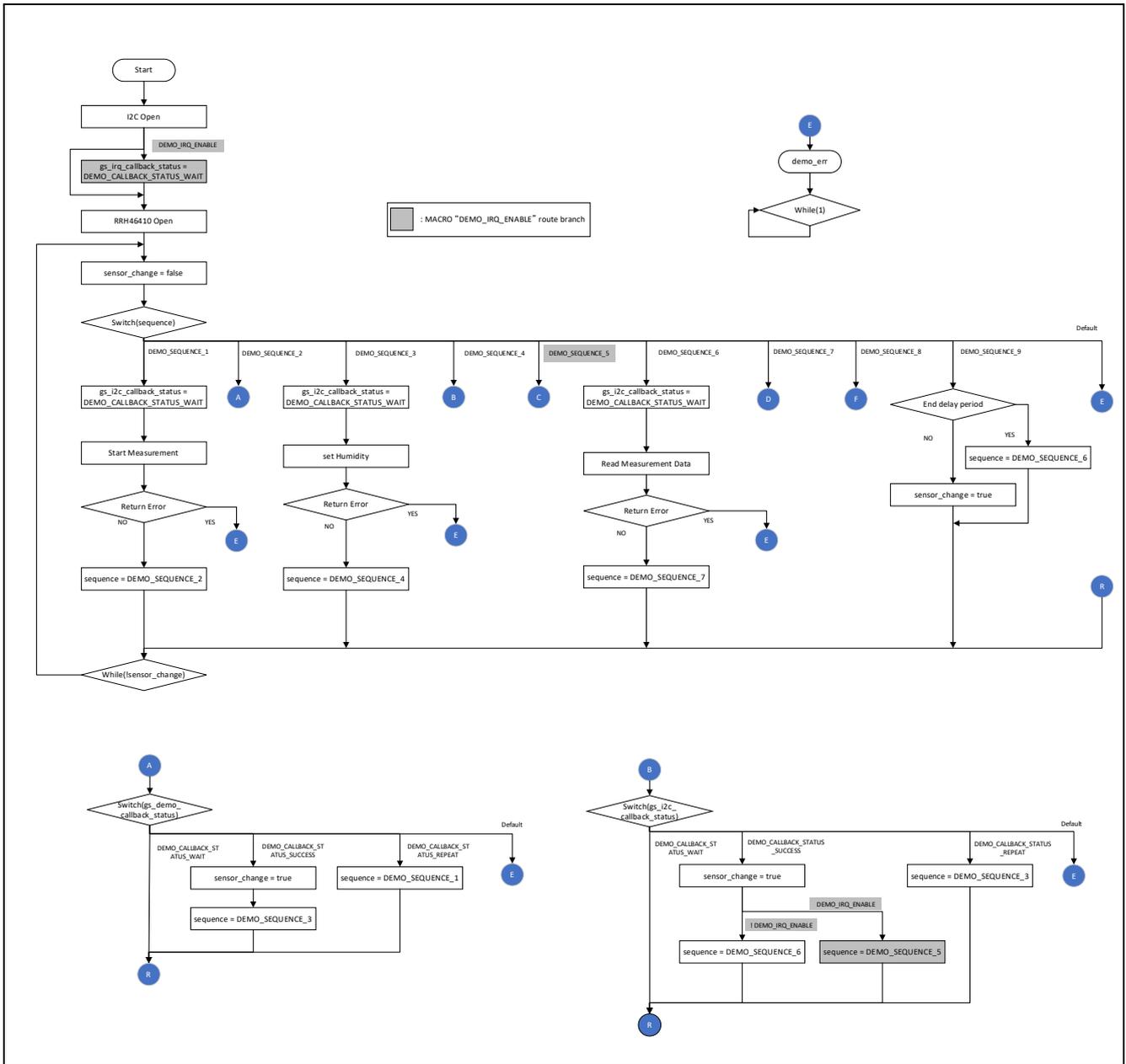


図 4-3 RRH46410 サンプルソフトウェア Non-OS 版メイン処理フロー1

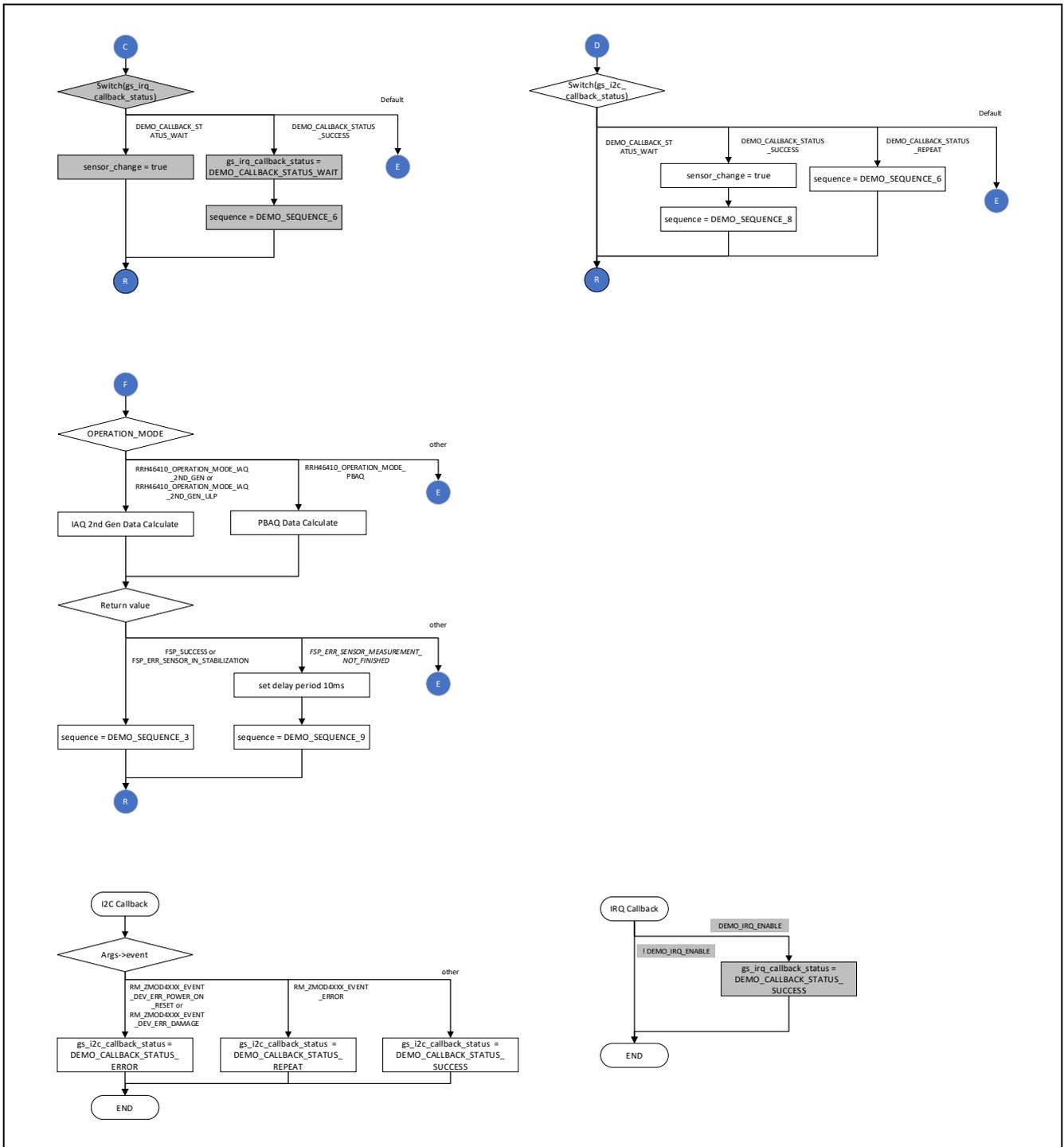


図 4-4 RRH46410 サンプルソフトウェア Non-OS 版メイン処理フロー2

### 4.4 サンプルソフトウェア OS 版メイン処理フロー

OS 版ではセマフォによる制御を行い、センサ制御を行うスレッドを動作させます。

スレッドのセンサ制御はドライバの開始処理を行い、その後はセンサデータ取得、測定結果演算の処理を繰り返します。

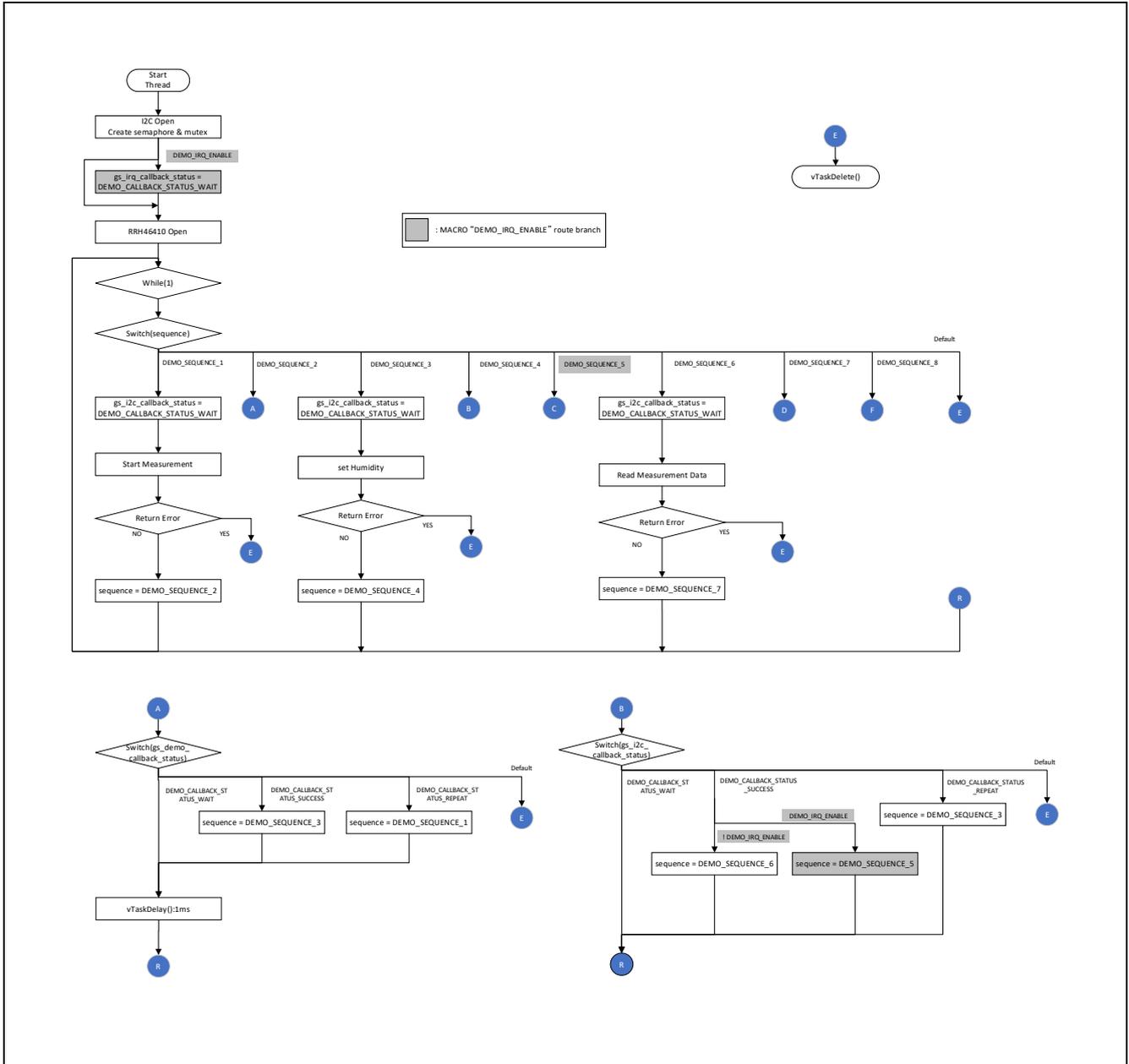


図 4-5 RRH46410 サンプルソフトウェア OS 版メイン処理フロー1

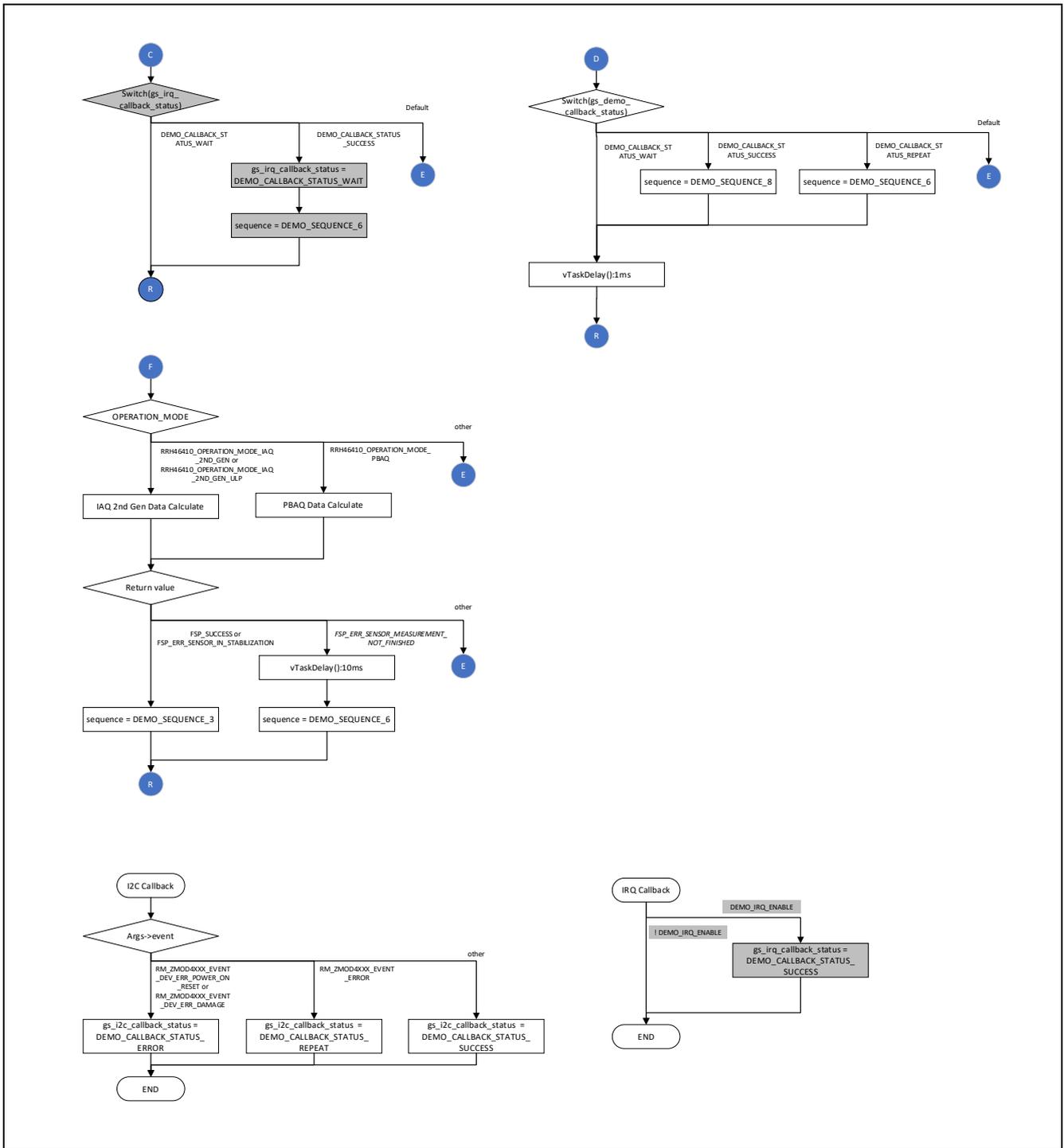


図 4-6 RRH46410 サンプルソフトウェア Non-OS 版メイン処理フロー2

## 5. Configuration 設定

設定可能な項目と設定値は以下のとおりです。

緑色の設定値はデフォルトで選択されている項目、橙色の設定値は変更不可の項目です。

モジュール名やコールバック関数名に関して、C 言語標準に準拠した名前を指定してください。

異なるバージョンのモジュールを利用する場合は、以下に示す設定項目や設定値が異なる場合があります。

### 5.1 RRH46410 Sensor Control Module 設定

#### 5.1.1 RA ファミリ

FSP Configurator の Stack タブで"rm\_rm46410"の Stack を選択することにより、Properties タブに設定可能な項目が表示されます。

表 5-1 RA RRH46410 設定一覧

設定項目	設定値	説明
Common		
Parameter Checking	Default (BSP)	パラメータチェック処理をコードに含めるか選択できます。 Disabled の場合、コードから省略します。 Enabled の場合、コードに含めます。
	Enabled	
	Disabled	
Operation Mode	IAQ 2nd Gen and Rel IAQ	RRH46410 のセンサ動作モードを設定します。
	IAQ 2nd Gen ULP and Rel IAQ ULP	
	Public Building AQ Standard (PBAQ)	
Module g_rrh46410_sensor0 RRH46410 Gas Sensor Module (rm_rrh46410)		
Name	g_rrh46410_sensor0	モジュール名を設定します。
Comms I2C Callback	rrh46410_comms_i2c_callback	ユーザコールバック関数名を設定します。 NULL を設定した場合は、コールバック関数は使用されません。
IRQ Callback	rrh46410_irq_callback	IRQ ユーザコールバック関数名を設定します。 NULL を設定した場合は、コールバック関数は使用されません。

## 5.1.2 RX ファミリ

Smart Configurator の Component タブで”r\_rrh46410\_rx”コンポーネントを選択することにより、Configure 領域に設定可能な項目が表示されます。

表 5-2 RX RRH46410 設定一覧

設定項目	設定値	説明
Configurations		
Parameter Checking	System Default	パラメータチェック処理をコードに含めるか 選択できます。 Disabled の場合、コードから省略します。 Enabled の場合、コードに含めます。
	Enabled	
	Disabled	
Number of RRH46410 Sensors	1	RRH46410 センサ数を設定します。
Operation mode of RRH46410 Sensor{x} (x = 0)	Not selected	RRH46410 センサ動作モードを設定します。
	IAQ 2nd Gen.	
	IAQ 2nd Gen. Ultra-Low Power	
	PBAQ	
I2C Communication device No. for RRH46410 sensor device{x} (x = 0)	I2C Communication Device{x} (x = 0 - 15)	I2C 通信デバイス番号を設定します。
I2C callback function for RRH46410 sensor device{x} (x = 0)	rrh46410_user_i2c_callback0	I2C コールバック関数名を設定します。
Enable IRQ from RRH46410 sensor device{x} (x = 0)	Disable	IRQ 許可を指定します。
	Enable	
IRQ Callback function for RRH46410 sensor device{x} (x = 0)	rrh46410_user_irq_callback0	IRQ コールバック関数名を設定します。
IRQ number for RRH46410 sensor device{x} (x = 0)	IRQ{x} (x = 0 - 15)	有効の IRQ 番号を指定します。
IRQ trigger for RRH46410 sensor device {x} (x = 0)	Low Level	IRQ トリガを指定します。 RRH46410 Sensor Pmod Board の場合は、 Falling に設定してください。（注 1）
	Falling	
	Rising	
	Both Edges	
IRQ interrupt priority for RRH46410 sensor device{x} (x = 0)	Priority{x} (x = 0 - 15)	IRQ 割り込みのプライオリティを指定します。

注 1 : 割り込みトリガの設定は、割り込み信号の回路に依存します。割り込み回路構成に関して、「6.5 割り込み信号回路に関する注意」を参照してください。

RRH46410 の INT 信号を MCU の割り込み端子に直接接続する回路構成の場合、“Rising”に設定してください。

## 5.1.3 RL78 ファミリ

Smart Configurator の Component タブで”r\_rrh46410”コンポーネントを選択することにより、Configure 領域に設定可能な項目が表示されます。

表 5-3 RL78 RRH46410 設定一覧

設定項目	設定値	説明
Configurations		
Parameter Checking	System Default	パラメータチェック処理をコードに含めるか 選択できます。 Disabled の場合、コードから省略します。 Enabled の場合、コードに含めます。
	Enabled	
	Disabled	
Number of RRH46410 Sensors	1	RRH46410 センサ数を設定します。
Operation mode of RRH46410 Sensor{x} (x = 0)	Not selected	RRH46410 センサ動作モードを設定します。
	IAQ 2nd Gen.	
	IAQ 2nd Gen. Ultra-Low Power	
	PBAQ	
I2C Communication device No. for RRH46410 sensor device{x} (x = 0)	g_comms_i2c_device{x} (x = 0 - 4)	I2C 通信デバイスインスタンスを設定します。
I2C callback function for RRH46410 sensor device{x} (x = 0)	rrh46410_user_i2c_callback0	I2C コールバック関数名を設定します。
Enable INTC from RRH46410 sensor device{x} (x = 0)	Disable	外部割込み(INTC)許可を指定します。
	Enable	
INTC Callback function for RRH46410 sensor device{x} (x = 0)	rrh46410_user_irq_callback0	外部割込み(INTC)コールバック関数名を設定 します。
INTC number for RRH46410 sensor device{x} (x = 0)	INTP{x} (x = 0 - 15)	外部割込み(INTC)の端子番号を指定します。

## 5.2 I2C Communication Middleware (COMMS\_I2C)設定

### 5.2.1 RA ファミリ

FSP Configurator の Stack タブで”**rm\_comms\_i2c**”の Stack を選択することにより、Properties タブに設定可能な項目が表示されます。

表 5-4 RA COMMS\_I2C 設定一覧

設定項目	設定値	説明
Common		
Parameter Checking	Default (BSP)	パラメータチェック処理をコードに含めるか選択できます。 Disabled の場合、コードから省略します。 Enabled の場合、コードに含めます。
	Enabled	
	Disabled	
Module g_comms_i2c_device0 I2C Communication Device (rm_comms_i2c)		
Name	g_comms_i2c_device0	モジュール名を設定します。
Semaphore Timeout	0xFFFFFFFF	RTOS プロジェクト時、semaphore のタイムアウト時間を設定します。
Slave Address	0x38	スレーブアドレスを設定します。 Sensor Control module により上書きされるため、設定不要です。
Address Mode	7-Bit	スレーブアドレスのビット幅を設定します。 Sensor Control module により上書きされるため、設定不要です。
Callback	rm_rrh46410_comms_i2c_callback	ユーザコールバック関数名を設定します。 Sensor Control module により上書きされるため、設定不要です。
Module g_comms_i2c_bus0 I2C Shared Bus (rm_comms_i2c)		
Name	g_comms_i2c_bus0	I2C モジュール名を設定します。
Bus Timeout	0xFFFFFFFF	I2C バスのタイムアウト時間を設定します
Semaphore for blocking	Unuse	RTOS プロジェクト時、Blocking 処理の有効/無効を設定します。
	Use	
Recursive Mutex for Bus	Unuse	RTOS プロジェクトかつ、Blocking 処理が有効の時、再帰動作の有効/無効を設定します。
	Use	
Channel	0	使用するチャンネル番号を設定します。 I2C ドライバが r_iic_master の場合のみ、この設定が有効です。 他の I2C ドライバの場合、この設定は無効です。
Rate	Standard	ビットレートを設定します。 RRH46410 の場合は、Standard もしくは Fast-mode の設定が可能です。同一バス上の他デバイスが接続されている場合、そのデバイスの設定可能な転送レートも考慮して設定してください。 I2C ドライバが r_iica_master の場合は、電気的特性のため Standard に設定してください。 I2C ドライバが r_sci_i2c の場合は、この設定は無効です。
	Fast-mode	
	Fast-mode plus	

## 5.2.2 RX ファミリ

Smart Configurator の Component タブで”r\_comms\_i2c\_rx”コンポーネントを選択することにより、Configure 領域に設定可能な項目が表示されます。

表 5-5 RX COMMS\_I2C 設定一覧

設定項目	設定値	説明
Configurations		
Parameter Checking	System Default	パラメータチェック処理をコードに含めるか選択できます。 Disabled の場合、コードから省略します。 Enabled の場合、コードに含めます。
	Enabled	
	Disabled	
Number of I2C Shared Buses	Unused	接続可能とする I2C バス数を設定します。
	1	
	2 - 16	
Number of I2C Communication Devices	Unused	接続可能とする I2C デバイスを設定します。
	1	
	2 - 16	
Blocking operation supporting with RTOS	Disabled	RTOS プロジェクト時のブロッキング動作を設定します。
	Enabled	
Bus lock operation supporting with RTOS	Disabled	RTOS プロジェクト時のバスロック動作を設定します。
	Enabled	
I2C Driver Type for I2C Shared bus{x} (x = 0 - 15)	RIIC	通信バスが使用する I2C バスの種別を設定します。 RIIC を使用する場合は、r_riic_rx、SCI IIC を使用する場合は、r_sci_iic_rx が必要となります。 使用しない FIT モジュールを削除すると、警告が表示されますが、動作に問題はありません。
	SCI IIC	
	Not selected	
Channel No. for I2C Shared bus{x} (x = 0 - 15)	0	通信バスが使用する I2C バスのチャンネル番号を設定します。
Timeout for the bus lock of I2C Shared Bus{x} (x = 0 - 15)	0xFFFFFFFF	I2C バスの I2C バスロックタイムアウト時間を設定します。
I2C Shared Bus No. for I2C Communication Device{x} (x = 0 - 15)	I2C Shared Bus{x} (x = 0 - 15)	通信バスが使用する I2C バスのコンフィグレーションを設定します。
Slave address for I2C Communication device{x} (x = 0 - 15)	0x00	通信バスに接続されるデバイスのスレーブアドレスを設定します。 RRH46410 の場合は、0x38 に設定してください。
Address mode for I2C Communication device{x} (x = 0 - 15)	7 bit address mode	スレーブアドレスモードを設定します。 RRH46410 の場合は、7 bit address mode に設定してください。
Callback function for I2C Communication device{x} (x = 0 - 15)	comms_i2c_user_callback{x} (x = 0 - 15)	ユーザコールバック関数名を設定します。 r_rrh46410_rx を使用する場合は、rm_rrh46410_callback{y} (y = 0) を設定します。
Timeout for the blocking bus of I2C Communication device{x} (x = 0 - 15)	0xFFFFFFFF	I2C バスの I2C バスブロッキングタイムアウト時間を設定します。

## 5.2.3 RL78 ファミリ

Smart Configurator の Component タブで”r\_comms\_i2c”コンポーネントを選択することにより、Configure 領域に設定可能な項目が表示されます。

表-6 RL78 COMMS\_I2C 設定一覧

設定項目	設定値	説明
Configurations		
Parameter Checking	System Default	パラメータチェック処理をコードに含めるか選択できます。 Disabled の場合、コードから省略します。 Enabled の場合、コードに含めます。
	Enabled	
	Disabled	
Number of I2C Shared Buses	Unused	接続可能とする通信バス数を設定します。
	1	
	2 - 5	
Number of I2C communication Devices	Unused	接続可能とする I2C デバイス数を設定します。
	1	
	2 - 5	
I2C Driver Type for I2C Shared Bus{x} (x = 0 - 4)	IICA	通信バスが使用する I2C バスの種別を設定します。 RRH46410 の場合は、IICA に設定してください。
	SAU IIC	
	Not selected	
Component name for the I2C Bus{x} (x = 0 - 4)	Config_IIC00	通信バスが使用する I2C バスのコンポーネント名を指定します。
I2C Shared Bus No. for I2C Communication Device{x} (x = 0 - 4)	I2C bus0	通信バスが使用する I2C バスのコンフィギュレーションを設定します。
	I2C bus1	
	I2C bus2	
	I2C bus3	
	I2C bus4	
Slave address for I2C Communication Device{x} (x = 0 - 4)	0x00	通信バスに接続されるデバイスのスレーブアドレスを設定します。 RRH46410 の場合は、0x38 に設定してください。
Callback function for I2C Communication Device{x} (x = 0 - 4)	comms_i2c_user_callback{x} (x = 0 - 4)	ユーザコールバック関数名を設定します。 r_rrh46410 を使用する場合は、 rm_rrh46410_callback{y} (y = 0)を設定します。

## 5.3 I2C ドライバ設定

### 5.3.1 RA ファミリ

FSP Configurator の Stack タブで”r\_iic\_master”、”r\_sci\_i2c”もしくは”r\_iica\_master”の Stack を選択することにより、Properties タブに設定可能な項目が表示されます。

**Simplified I2C using Serial Array Unit (SAU)**は、クロック・ストレッチ機能を未サポートのため使用できません。

#### (1) r\_iic\_master

表 5-7 RA r\_iic\_master 設定一覧

設定項目	設定値	説明
Common		
Parameter Checking	Default (BSP)	パラメータチェック処理をコードに含めるか選択できます。
	Enabled	Disabled の場合、コードから省略します。
	Disabled	Enabled の場合、コードに含めます。
DTC on Transmission and Reception	Enabled	送受信に DTC を使用するか設定します。
	Disabled	
10-bit slave addressing	Enabled	10-bit addressing をサポートするか設定します。
	Disabled	COMMS_I2C により上書きされるため、設定不要です。
Module g_i2c_master0 I2C Master (r_iic_master)		
Name	g_i2c_master0	モジュール名を設定します。
Channel	0	使用するチャンネル番号を設定します。 COMMS_I2C により上書きされるため、設定は不要です。
Rate	Standard	ビットレートを設定します。
	Fast-mode	COMMS_I2C により上書きされるため、設定は不要です。
	Fast-mode plus	
Custom Rate (bps)	0	カスタムビットレートを設定します。 0 以外の場合に設定が有効です。Rate の設定範囲内の低ビットレートを設定したい場合に利用してください。
Rise Time (ns)	120	SCL の立ち上がり時間を設定します。使用するボードに合わせて設定してください。
Fall Time (ns)	120	SCL の立ち下がり時間を設定します。使用するボードに合わせて設定してください。
Duty Cycle (%)	50	SCL のデューティ比を設定します。
Slave Address	0x00	接続するデバイスのスレーブアドレスを設定します。 COMMS_I2C により上書きされるため、設定不要です。
Address Mode	7-Bit	接続するデバイスのスレーブアドレスモードを設定します。
	10-Bit	COMMS_I2C により上書きされるため、設定不要です。
Timeout Mode	Short Mode	I2C バスのタイムアウト時間を設定します
	Long Mode	
Timeout during SCL low	Enabled	SCL がタイムアウト モードで設定されている時間よりも長い時間 Low に保持されている場合にタイムアウトするか設定する。
	Disabled	
Callback	rm_comms_i2c_callback	ユーザコールバック関数名を設定します。 COMMS_I2C により上書きされるため、設定不要です。
Interrupt Priority Level	Priority 0 (highest)	I2C バスドライバの割り込み優先レベルを設定します。
	Priority 1	
	Priority 2	
	Priority 3	
Pins		
SDA	Pxxx	ドライバが使用する端子番号が表示されます。 端子の設定は、Pins タブで行います。
SCL	Pxxx	

## (2) r\_sci\_i2c

表 5-8 RA r\_sci\_i2c 設定一覧

設定項目	設定値	説明
Common		
Parameter Checking	Default (BSP)	パラメータチェック処理をコードに含めるか選択できます。
	Enabled	Disabled の場合、コードから省略します。
	Disabled	Enabled の場合、コードに含めます。
DTC on Transmission and Reception	Enabled	送受信に DTC を使用するか設定します。
	Disabled	
10-bit slave addressing	Enabled	10-bit addressing をサポートするか設定します。
	Disabled	COMMS_I2C により上書きされるため、設定不要です。
Module g_i2c0 I2C Master (r_sci_i2c)		
Name	g_i2c0	モジュール名を設定します。
Channel	0	使用するチャンネル番号を設定します。
Slave Address	0x00	接続するデバイスのスレーブアドレスを設定します。COMMS_I2C により上書きされるため、設定不要です。
Address Mode	7-Bit	接続するデバイスのスレーブアドレスモードを設定します。
	10-bit	COMMS_I2C により上書きされるため、設定不要です。
Rate	Standard	ビットレートを設定します。
	Fast-mode	RRH46410 の場合は、Standard もしくは Fast-mode の設定が可能です。同一バス上の他デバイスが接続されている場合、そのデバイスの設定可能な転送レートも考慮して設定してください。
Custom Rate (bps)	0	カスタムビットレートを設定します。0 以外の場合に設定が有効です。Rate の設定範囲内の低ビットレートを設定したい場合に利用してください。
SDA Output Delay (nano seconds)	300	SDA 出力遅延時間を設定します。
Noise filter setting	Use clock signal divided by 1 with noise filter	入力信号のノイズフィルタ使用を設定します。
	Use clock signal divided by 2 with noise filter	
	Use clock signal divided by 4 with noise filter	
	Use clock signal divided by 8 with noise filter	
Bit Rate Modulation	Enable	Bit Rate Modulation 機能の使用を設定します。
	Disable	
Callback	rm_comms_i2c_callback	ユーザコールバック関数名を設定します。COMMS_I2C により上書きされるため、設定不要です。
Interrupt Priority Level	Priority 0 (highest)	I2C 割り込みの割り込み優先レベルを設定します。
	Priority 1	
	Priority 2	
	Priority 3	
RX Interrupt Priority Level [Only used when DTC is enabled]	Priority 0 (highest)	DTC を使用した場合の受信割り込みの割り込み優先レベルを設定します。
	Priority 1	
	Priority 2	
	Priority 3	
	Disabled	
Pins		
SDA	Pxxx	ドライバが使用する端子番号が表示されます。
SCL	Pxxx	端子の設定は、Pins タブで行います。

## (3) r\_iica\_master

FSP v5.4.0 以降で IICA を設定する場合は、Stacks タブの”SCLA Pin”と”SDAA Pin”をピン番号のみ設定してください。

表 5-9 RA r\_iica\_master 設定一覧

設定項目	設定値	説明
Common		
Parameter Checking	Default (BSP)	パラメータチェック処理をコードに含めるか選択できます。
	Enabled	Disabled の場合、コードから省略します。
	Disabled	Enabled の場合、コードに含めます。
10-bit slave addressing	Enabled	10-bit addressing をサポートするか設定します。
	Disabled	COMMS_I2C により上書きされるため、設定不要です。
Module g_iica_master0 IICA Master (r_iica_master)		
Name	g_iica_master0	モジュール名を設定します。
Rate	Standard	ビットレートを設定します。
	Fast-mode	COMMS_I2C により上書きされるため、設定不要です。
	Fast-mode plus	
Custom Rate (bps)	0	カスタムビットレートを設定します。 0 以外の場合に設定が有効です。Rate の設定範囲内の低ビットレートを設定したい場合に利用してください。
Signal Rising Times (us)	0	SCL の立ち上がり時間を設定します。使用するボードに合わせて設定してください。
Signal Falling Times (us)	0	SCL の立ち下がり時間を設定します。使用するボードに合わせて設定してください。
Duty Cycle (%)	53	SCL のデューティ比を設定します。
Digital Filter	Enabled	デジタルフィルタ機能の使用を設定します。
	Disabled	
Address Mode	7-Bit	接続するデバイスのスレーブアドレスモードを設定します。
	10-bit	COMMS_I2C により上書きされるため、設定不要です。
Slave Address	0x00	接続するデバイスのスレーブアドレスを設定します。 COMMS_I2C により上書きされるため、設定不要です。
Communication reservation	Enabled Disabled	通信予約機能の許可を設定します。
Callback	rm_comms_i2c_callback	ユーザコールバック関数名を設定します。 COMMS_I2C により上書きされるため、設定不要です。
IICA0 communication interrupt priority	Priority 0 (highest)	I2C 割り込みの割り込み優先レベルを設定します。
	Priority 1	
	Priority 2	
	Priority 3	
SCLA Pin	Pxxx	ドライバが使用する端子番号を設定します。
SDAA Pin	Pxxx	Pins タブでの設定は不要です。

**g\_iica\_master0 IICA Master (r\_iica\_master)**

Settings	Property	Value
API Info	> Common	
	Module g_iica_master0 IICA Master (r_iica_master)	
	Name	g_iica_master0
	Rate	Standard
	Signal Rising Time (us)	0
	Signal Falling Time (us)	0
	Duty Cycle (%)	53
	Digital Filter	Disabled
	Address Mode	7-Bit
	Slave Address	0x00
	Communication reservation	Disabled
	Callback	rm_comms_i2c_callback
	IICA0 communication interrupt priority	Priority 2
	SCLA Pin	P100
	SDAA Pin	P101

## 5.3.2 RX ファミリ

Smart Configurator の Component タブで”r\_riic\_rx”もしくは”r\_sci\_iic\_rx”コンポーネントを選択することにより、Configure 領域に設定可能な項目が表示されます。

## (1) r\_riic\_rx

表 5-10 RX r\_riic\_rx 設定一覧

設定項目	設定値	説明
Common		
Set parameter checking enable	System Default	パラメータチェック処理をコードに含めるか選択できます。
	Not	Not の場合、コードから省略します。
	Include	Include の場合、コードに含めます。
MCU supported channels for CH{x} (x = 0 - 2)	Not supported	該当チャンネルに関する処理をコードに含めるか選択できません。
	Supported	Not supported の場合、コードから省略します。 Supported の場合、コードに含めます。
CH{x} RIIC bps(kbps) (x = 0 - 2)	400	ビットレートを設定できます。 RRH46410 の場合は、400kbps 以下を設定してください。 同一バス上の他デバイスが接続されている場合、そのデバイスの設定可能な転送レートも考慮して設定してください。
Digital filter for CH{x} (x = 0 - 2)	Not	指定したチャンネルのノイズフィルタの段数を選択できます。 Not の場合、ノイズフィルタは無効となります。
	One IIC phi	
	Two IIC phi	
	Three IIC phi	
	Four IIC phi	
Setting port setting processing	Not include port setting	ポートを SCL, SDA 端子として使用するための設定処理をコードに含めるかを選択します。 Not include port setting の場合、コードから省略します。 Include port setting の場合、コードに含めます。
	Include port setting	
Master arbitration lost detection function for CH{x} (x = 0 - 2)	Unused	指定したチャンネルのマスターアービトレーションロスト検出機能の有効/無効を選択できます。 マルチマスタで使用する場合は、”Used”にしてください。 Unused の場合、無効にします。 Used の場合、有効にします。
	Used	
Address {y} format for CH{x} (x = 0 - 2, y = 0 - 2)	Not	スレーブアドレスのフォーマットを 7-bit/10-bit から選択できます。 RRH46410 の場合は、7-bit address format に設定してください。同一バス上に異なるアドレスフォーマットのデバイスを接続しないでください。
	7 bit address format	
	10 bit address format	
Slave Address {y} for CH{x} (x = 0 - 2, y = 0 - 2)	0x0025	指定したデバイスのスレーブアドレスを設定します。 COMMS_I2C により上書きされるため、設定不要です。
General call address for CH{x} (x = 0 - 2)	Unused	指定したチャンネルのゼネラルコールアドレスの有効/無効が選択できます。 Unused の場合、無効にします。 Used の場合、有効にします。
	Used	
CH{x} RXI INT Priority Level (x = 0 - 2)	Level 1	指定したチャンネルの受信データフル割り込み(RXI)の優先レベルを選択できます。
	Level 2	
	...	
	Level 14	
	Level 15 (highest)	

CH{x} TXI INT Priority Level (x = 0 - 2)	Level 1	指定したチャンネルの送信データエンプティ割り込み(TXI)の優先レベルを選択できます。
	Level 2	
	...	
	Level 14	
	Level 15 (highest)	
CH{x} EEI INT Priority Level (x = 0 - 2)	Level 1	指定したチャンネルの通信エラー/イベント発生割り込み(EEI)の優先レベルを選択できます。
	Level 2	
	...	
	Level 14	
	Level 15 (highest)	
CH{x} TEI INT Priority Level (x = 0 - 2)	Level 1	指定したチャンネルの送信終了割り込み(TEI)の優先レベルを選択できます。
	Level 2	
	...	
	Level 14	
	Level 15 (highest)	
Timeout function for CH{x} (x = 0 - 2)	Unused	指定したチャンネルのタイムアウト検出機能を有効にできません。 Unused の場合、タイムアウト検出機能無効 Used の場合、タイムアウト検出機能有効
	Used	
Timeout detection time for CH{x} (x = 0 - 2)	Long mode	指定したチャンネルのタイムアウト検出時間を選択できます。 Long mode の場合、ロングモードを選択。 Short mode の場合、ショートモードを選択。
	Short mode	
Count up during low period of timeout detection for CH{x} (x = 0 - 2)	Unused	指定したチャンネルのタイムアウト検出機能有効時、SCL ラインが Low 期間中にタイムアウト検出機能の内部カウンタのカウンタアップを有効にできます。 Unused の場合、カウンタアップ禁止。 Used の場合、カウンタアップ有効。
	Used	
Count up during high period of timeout detection for CH{x} (x = 0 - 2)	Unused	指定したチャンネルのタイムアウト検出機能有効時、SCL ラインが High 期間中にタイムアウト検出機能の内部カウンタのカウンタアップを有効にできます。 Unused の場合、カウンタアップ禁止。 Used の場合、カウンタアップ有効。
	Used	
Set Counter of checking bus busy	1000	API 関数のバスチェック処理時に、ソフトウェアによりタイムアウトカウンタ(バス確認回数)を設定できます。
Resources		
SCLx 端子	Checked	使用する端子を設定します。 使用する端子を Checked に設定してください。
	Unchecked	
SDAx 端子	Checked	
	Unchecked	

## (2) r\_sci\_iic\_rx

表 5-11 RX r\_sci\_iic\_rx 設定一覧

設定項目	設定値	説明
Configurations		
Set parameter checking enable	System Default	パラメータチェック処理をコードに含めるか選択できます。
	Not	Not の場合、コードから省略します。
	Include	Include の場合、コードに含めます。
MCU supported channels for CH{x} (x = 0 - 12)	Not supported	該当チャネルに関する処理をコードに含めるか選択できます。
	Supported	Not supported の場合、コードから省略します。 Supported の場合、コードに含めます。
SCI IIC bitrate (bps) for CH{x} (x = 0 - 12)	384000	ビットレートを設定してください。 RRH46410 の場合は、384000bps 以下を設定してください。同一バス上の他デバイスが接続されている場合、そのデバイスの設定可能な転送レートも考慮して設定してください。
Interrupt Priority for CH{x} (x = 0 - 12)	Level 1	コンディション割り込み、受信割り込み、送信空割り込み、送信完了割り込みの優先レベルを設定してください。
	Level 2	
	...	
	Level 14	
	Level 15 (highest)	
Digital noise filter (NFEN bit) for CH{x} (x = 0 - 12)	Disable	SSCL、SSDA 入力信号のノイズ除去機能を使用するか選択できます。 Disable の場合、無効にします。 Enable の場合、有効にします。
	Enable	
Noise Filter Setting Register (NFCS bit) for CH{x} (x = 0 - 12)	The clock divided by 1	デジタルノイズフィルタのサンプリングクロックを選択します。
	The clock divided by 2	
	The clock divided by 4	
	The clock divided by 8	
I2C Mode Register 1 (IICDL bit) for CH{x} (x = 0 - 12)	18	SSCL 端子出力の立ち下がりに対する SSDA 端子出力の遅延を選択します。 1 ~ 31 の範囲で設定してください。
Software bus busy check counter	1000	バスビジー判定のカウント数を設定します。 簡易 I2C の API 関数のバスチェック処理時の、タイムアウトカウンタ(バス確認回数)を設定できます。
Port Setting Processing	Not include port setting	ポートを SSCL、SSDA 端子として使用するための設定処理をコードに含めるか選択できます。 Not include port setting の場合、コードから省略します。 Include port setting の場合、コードに含めます。
	Include port setting	
Resources		
SSCLx 端子	Checked	使用する端子を設定します。 使用する端子を Checked に設定してください。
	Unchecked	
SSDAx 端子	Checked	
	Unchecked	

## 5.3.3 RL78 ファミリ

Smart Configurator の Component タブで IIC 通信(マスタモード)からリソースに”IICAx”を選択することにより、Configure 領域に設定可能な項目が表示されます。

**Simplified I2C using Serial Array Unit (SAU)は、クロック・ストレッチ機能を未サポートのため使用できません。**

## (1) IICAx

表 5-12 RL78 IICAx 設定一覧

設定項目	設定値	説明
Configurations		
クロック・モード設定	fCLK	カウント・クロックを設定します。
	fCLK/2	
アドレス	16	自局アドレスを設定します。
動作モード	標準	動作モードを設定します。
	ファスト・モード	RRH46410 の場合は、標準もしくはファスト・モードの設定が可能です。同一バス上の他デバイスが接続されている場合、そのデバイスの設定可能な転送レートも考慮して設定してください。
	ファスト・モード・プラス	
デジタル・フィルタ・オン	Checked	デジタルフィルタ機能の使用を設定します。
	Unchecked	
転送クロック (fSCL)	100000	ビットレートを設定します。 IICAx の電気的特性のため、100000bps 以下に設定してください。
tR と tF の手動入力設定	Checked	SDAAn, SCLAn 信号の立ち上がり時間、立ち下がり時間を手動で設定します。
	Unchecked	
tR	0	SDAAn, SCLAn 信号の立ち上がり時間を設定します。
tF	0	SDAAn, SCLAn 信号の立ち下がり時間を設定します。
通信完了割り込み優先順位(INTIICAx)	レベル 0(高優先順位)	通信完了割り込みの割り込み優先順位を設定します。
	レベル 1	
	レベル 2	
	レベル 3(低優先順位)	
マスタ送信完了	Checked	マスタ送信完了によるコールバック機能を設定します。
	Unchecked	
マスタ受信完了	Checked	マスタ受信完了によるコールバック機能を設定します。
	Unchecked	
マスタ・エラー	Checked	通信エラーによるコールバック機能を設定します。
	Unchecked	
マスタ送信/受信完了コールバック時にストップ・コンディションを生成	Checked	コールバック時のストップ・コンディション生成を設定します。Unchecked に設定してください。
	Unchecked	

## 5.4 IRQ ドライバ設定

割り込み信号回路の使用に関して、「6.5 割り込み信号回路に関する注意」を参照してください。

### 5.4.1 RA ファミリ

FSP Configurator の Stack タブで”r\_icu”の Stack を選択することにより、Properties タブに設定可能な項目が表示されます。

以下は RRH46410 Sensor Pmod Board の設定例です。

表 5-13 RA r\_icu 設定一覧

設定項目	設定値	説明
Common		
Parameter Checking	Default (BSP)	パラメータチェック処理をコードに含めるか選択できます。
	Enabled	Disabled の場合、コードから省略します。
	Disabled	Enabled の場合、コードに含めます。
Module g_external_irq0 External IRQ (r_icu)		
Name	g_external_irq0	モジュール名を設定します。
Channel	0	使用するチャンネル番号を設定します。
Trigger	Falling	トリガを設定します。
	Rising	RRH46410 Sensor Pmod Board の場合は、Falling に設定してください。(注 1)
	Both Edges	
	Low Level	
Digital Filtering	Enabled	デジタルフィルタ機能の使用を設定します。
	Disabled	
Digital Filtering Sample Clock (Only valid when Digital Filtering is Enabled)	PCLK / 1	デジタルフィルタのサンプリングクロックを選択します。
	PCLK / 8	
	PCLK / 32	
	PCLK / 64	
Callback	rm_rrh46410_irq_callback	ユーザコールバック関数名を設定します。 r_icu により自動的に設定されます。
Pin Interrupt Priority	Priority 0 (highest)	IRQ ドライバの割り込み優先レベルを設定します。
	Priority 1	
	Priority 2	
	Priority 3	
Pins		
IRQ	Pxxx	ドライバが使用する端子番号が表示されます。 端子の設定は、Pins タブで行います。

注 1 : 割り込みトリガの設定は、割り込み信号の回路に依存します。割り込み回路構成に関して、「6.5 割り込み信号回路に関する注意」を参照してください。

RRH46410 の INT 信号を MCU の割り込み端子に直接接続する回路構成の場合、“Rising”に設定してください。

### 5.4.2 RX ファミリ

Smart Configurator の Component タブで”r\_irq\_rx”コンポーネントを選択することにより、Configure 領域に設定可能な項目が表示されます。

以下は RRH46410 Sensor Pmod Board の設定例です。

表 5-14 RX r\_irq\_rx 設定一覧

設定項目	設定値	説明
Configurations		
Locking function for IRQ APIs	Enable	IRQ のロック機能の有効/無効を設定します。
	Disable	
Set parameter checking enable	System Default	パラメータチェック処理をコードに含めるか選択できます。Not の場合、コードから省略します。Include の場合、コードに含めます。
	Not	
	Include	
Filter for IRQ{x} (x = 0 - 15)	Enable	デジタルフィルタ機能の使用を設定します。
	Disable	
Filter clock divisor for IRQ{x} (x = 0 - 15)	Divisor 1	デジタルフィルタのサンプリングクロックを選択します。
	Divisor 8	
	Divisor 32	
	Divisor 64	
IRQx 端子	Checked	使用する端子を設定します。
	Unchecked	使用する端子を Checked に設定してください。

### 5.4.3 RL78 ファミリ

Smart Configurator の Component タブで”割り込みコントローラ”を選択することにより、Configure 領域に設定可能な項目が表示されます。

以下は RRH46410 Sensor Pmod Board の設定例です。

表 5-15 RL78 割り込みコントローラ設定一覧

設定項目	設定値	説明
Configurations		
INTP{x} (x = 0 - 11)	Checked	INTP{x}の有効/無効を設定します。 (x = 0 - 11)
	Unchecked	
有効エッジ	立ち下がりエッジ	トリガを設定します。 RRH46410 Sensor Pmod Board の場合は、立ち下がりエッジに設定してください。(注 1)
	立ち上がりエッジ	
	両エッジ	
優先順位	レベル 0(高優先順位)	INTP{x}の割り込み優先レベルを設定します。 (x = 0 - 11)
	レベル 1	
	レベル 2	
	レベル 3(低優先順位)	

注 1 : 割り込みトリガの設定は、割り込み信号の回路に依存します。割り込み回路構成に関して、「6.5 割り込み信号回路に関する注意」を参照してください。

RRH46410 の INT 信号を MCU の割り込み端子に直接接続する回路構成の場合、“立ち上がりエッジ”に設定してください。

## 6. デバイス変更ガイド

サンプルプロジェクトを異なるデバイスで動作させるには、以下の手順に従ってください。

また、以下の回路に関して、設定を見直す必要があります。

- 割り込み信号回路：「6.5 割り込み信号回路に関する注意」を参照してください。
- RESET 信号回路：「6.6 RESET 信号回路に関する注意」を参照してください。

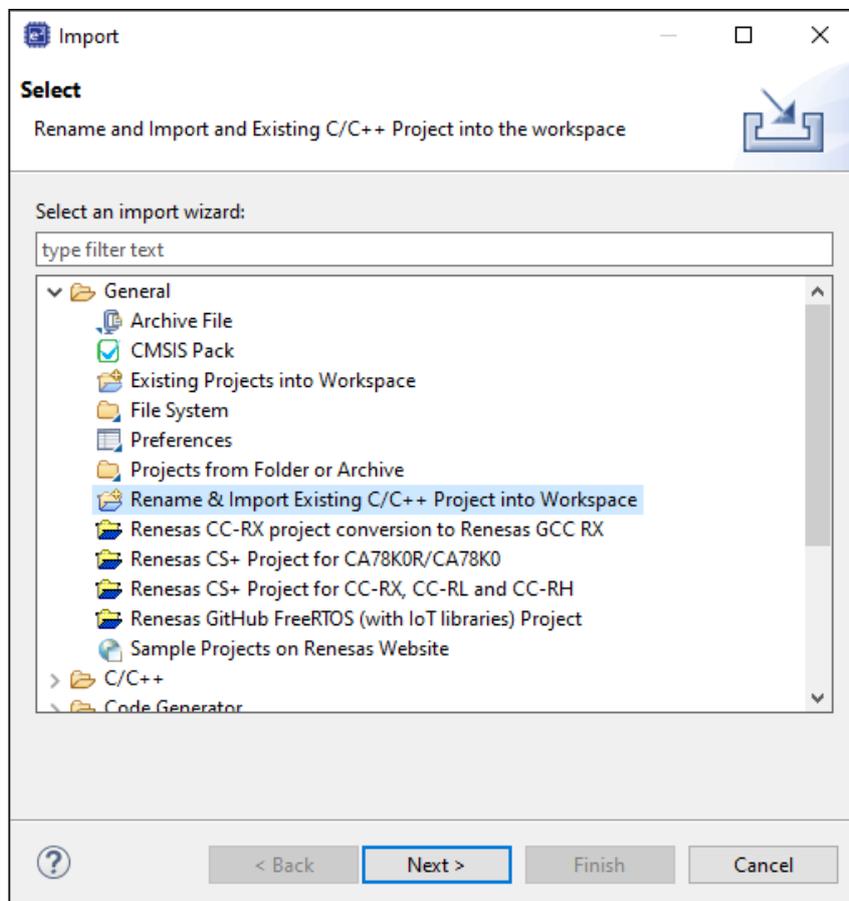
### 6.1 サンプルプロジェクトのインポート

サンプルプロジェクトのデバイスを変更するには、インポートする必要があります。

サンプルプロジェクトのインポートは以下の手順で行います。

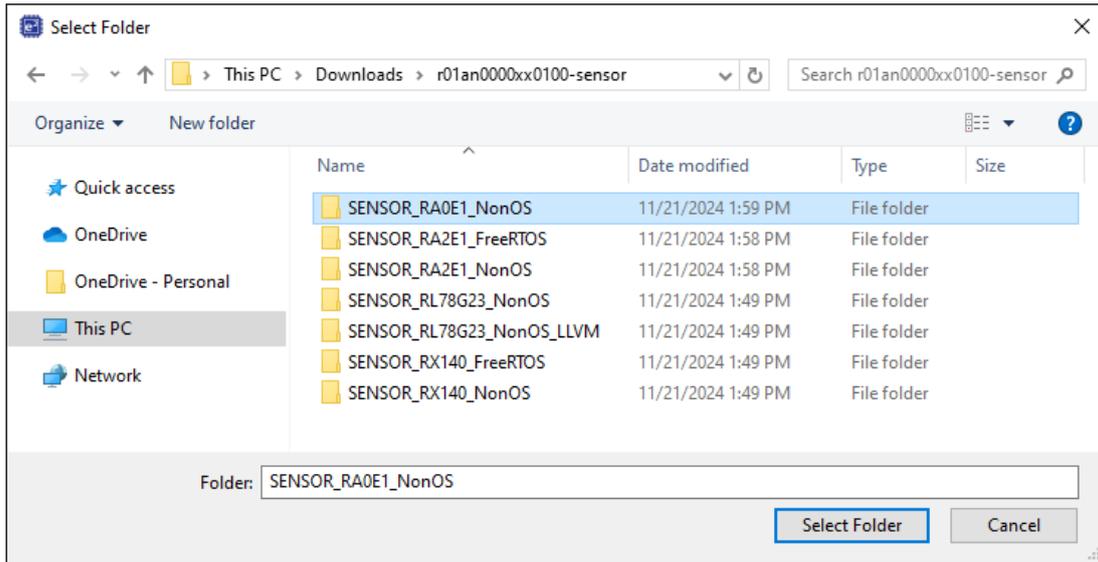
1. メニューから、[Import]を選択します。

表示されたインポートウィンドウで、“Rename & Import Existing C/C++ Project into Workspace”を選択し、[Next]ボタンを押下します。

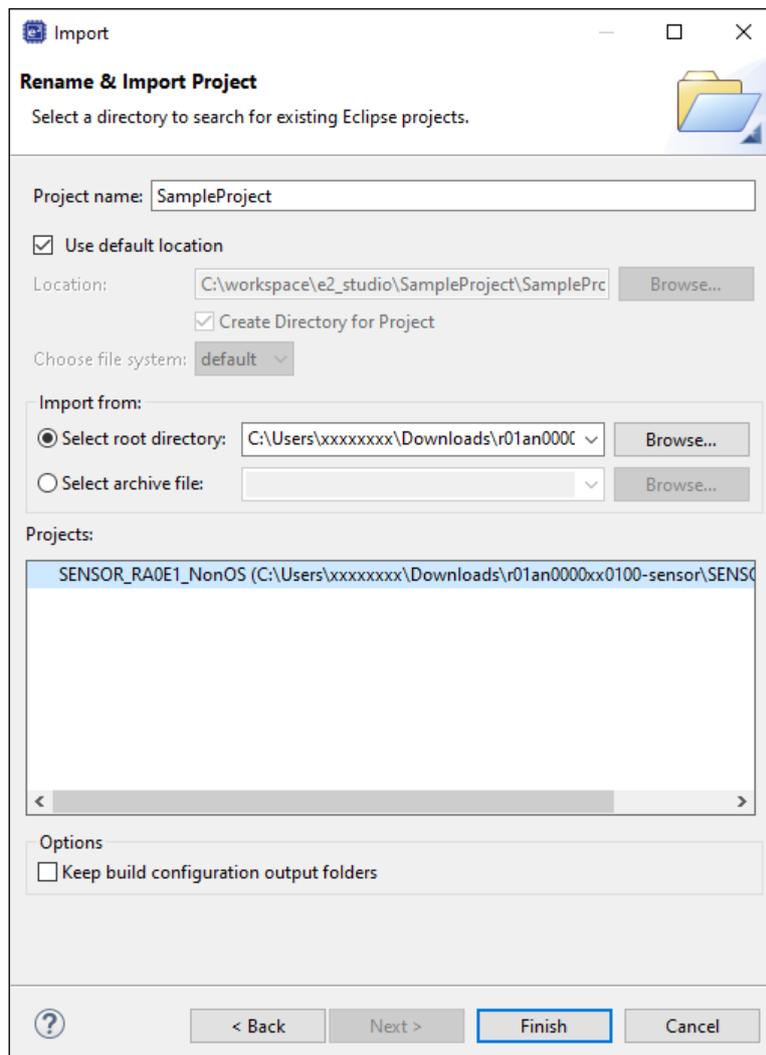


2. [Browse]ボタンを押下し、フォルダの選択ウィンドウを表示します。

任意のサンプルプロジェクトのフォルダを選択し、[フォルダの選択]ボタンを押下します。



3. 任意のプロジェクト名を入力および、移行元デバイスのプロジェクトを選択し、[Finish]ボタンを押下します。



## 6.2 RA サンプルプロジェクト

サンプルプロジェクトをインポート後に以下の手順を実行してください。インポートの方法は「6.1 サンプルプロジェクトのインポート」を参照してください。

以下のボード変更例での変更手順を説明します。なお、Pmod Type 2A/3A コネクタ使用時は Interposer Board が必要です。

- サンプルプロジェクト “RRH46410\_RA2E1\_NonOS”:  
Pmod1 (Type 2A/3A: SCIO)  
→ EK-RA6M4 ボードの Pmod1 (Option Type 6A: IIC1)もしくは Pmod2 (Type 2A: SCIO)

### 6.2.1 FSP Configurator の設定変更

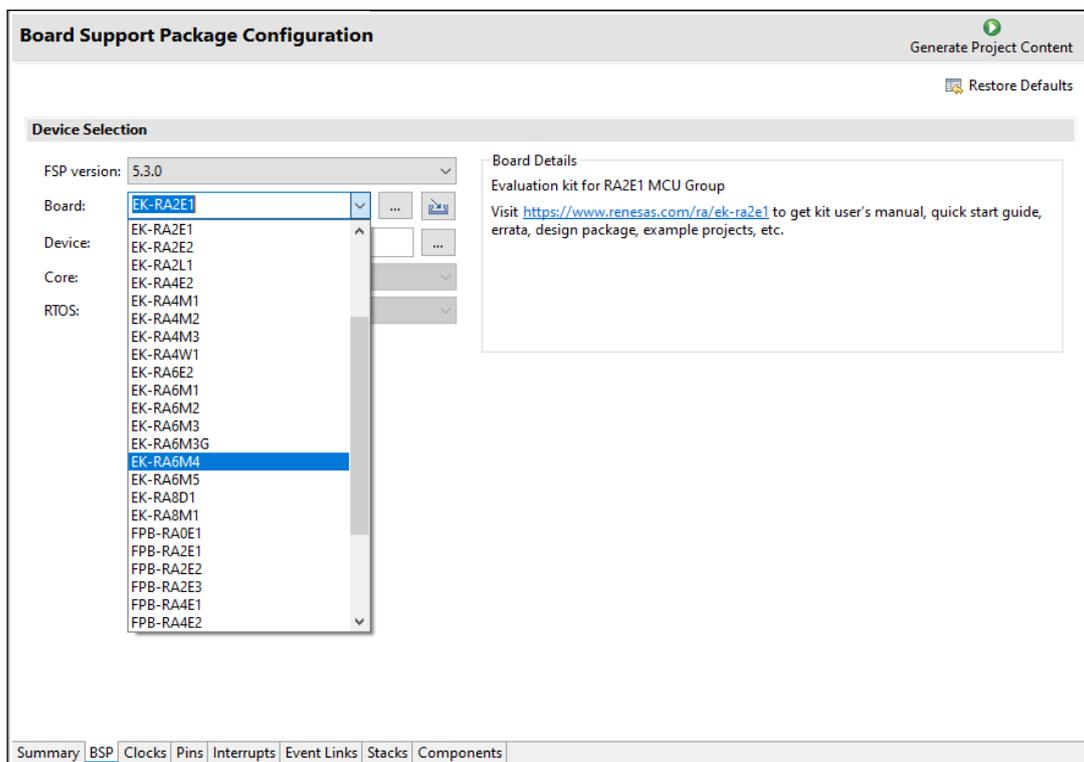
プロジェクトツリーの Configuration.xml をダブルクリックし、FSP Configurator を開きます。

#### (1) BSP

BSP タブで Board および Device を変更します。

ルネサス製ボードに変更する場合は、Board の設定のみ変更してください。

ルネサス製以外のボードに変更する場合は、Board を”Custom User Board (Any Device)”に変更後、Device を使用するデバイスに変更してください。

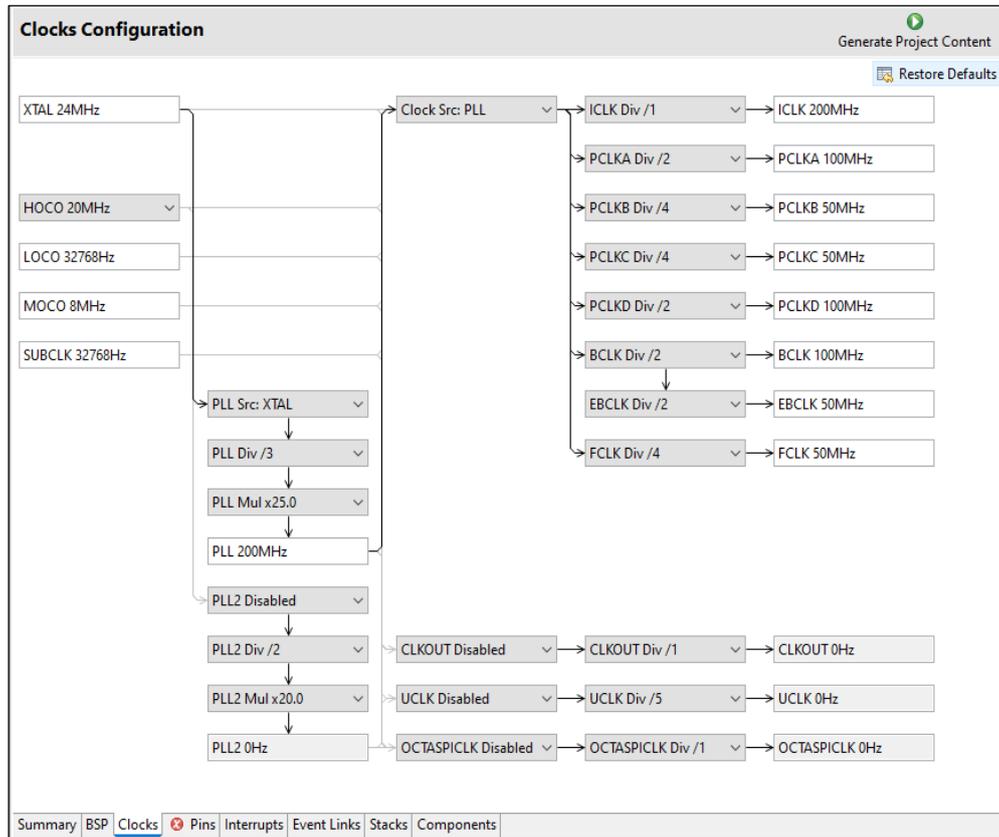


## (2) Clocks

Clocks タブで、クロック設定を変更します。

Board を”Custom User Board (Any Device)”に変更した場合は、使用するボードに合わせてクロック設定を変更してください。

Board をルネサス製ボードに変更した場合は、自動的に設定が変更されます。

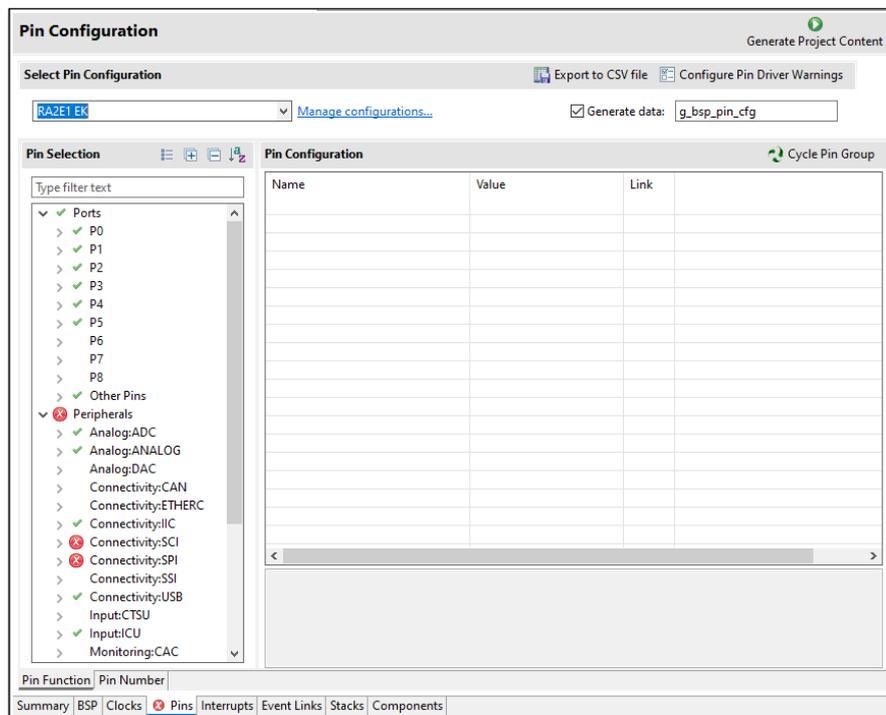


## (3) Pins

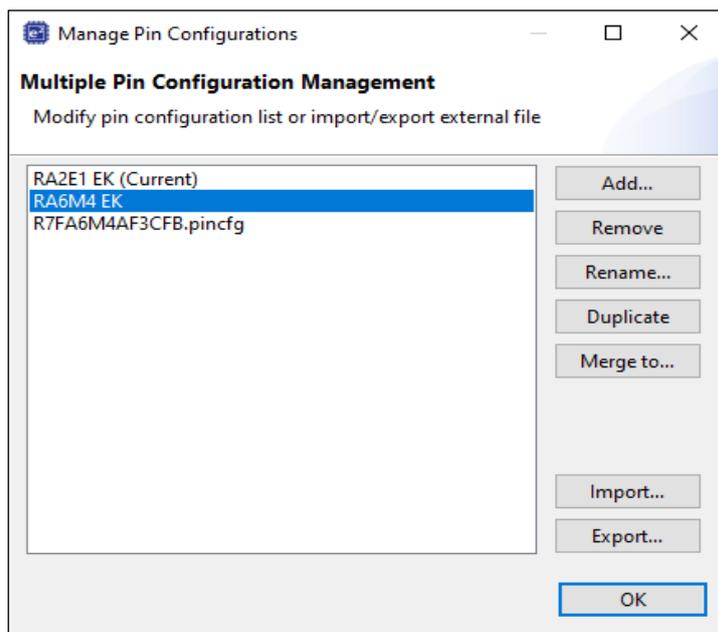
## (a) ボード変更

Pins タブで、使用するボードに合わせて、端子設定を変更します。

ルネサス製ボードを使用する場合は、Select Pin Configuration を"RA2E1 EK"から使用するボードに変更することで、自動的に割り当てが行われます。



Select Pin Configuration のドロップダウンリストに変更したいボードが表示されない場合は、[Manage Configuration]をクリックし、表示された Manage Pin Configuration ウィンドウで、使用したいボードを選択してください。



## (b) I2C I/F 端子変更

EK-RA6M4 ボードの場合、上述の「(a)ボード変更」での割り当てでは Pmod Type 2A に対応した SPI 通信の端子設定が適用されます。

このサンプルソフトウェアでは、Pmod Type 6A を使用するため、Pmod Type 6A に対応した I2C 通信の端子設定への変更が必要です。

EK-RA6M4 ボードでは、Pmod1 に IIC1 (Pmod1 #3 P512 SCL1 と Pmod1 #4 P511 SDA1)、Pmod2 に SCIO (Pmod2 #3 P410 SCL0 と Pmod2 #2 P411 SDA0) が割り当てられています。

そのため、I2C 通信を行う端子は以下となるため、Select Pin Configuration の自動割り当て後、Pin Configuration にて再設定を行います。

- Pmod1 (Option Type 6A)使用時 : SCL1 を P512 に、SDA1 を P511 に設定
- Pmod2 使用時 (Interposer Board を利用) : SCL0 を P410 に、SDA0 を P411 に設定

The screenshot shows the 'Pin Configuration' tool interface. At the top, it says 'RA6M4 EK' and 'Generate Project Content'. Below that, there are buttons for 'Export to CSV file' and 'Configure Pin Driver Warnings'. A dropdown menu shows 'RA6M4 EK' and a 'Manage configurations...' link. A checkbox 'Generate data:' is checked with the value 'g\_bsp\_pin\_cfg\_6m4'.

The main area is split into two panes. The left pane, 'Pin Selection', has a search filter and a tree view showing categories like 'Other Pins', 'Peripherals', 'Analog:ADC', 'Analog:ANALOG', 'Analog:DAC', 'Connectivity:CAN', 'Connectivity:ETHERC', 'Connectivity:IIC', and 'Connectivity:SCI'. Under 'Connectivity:SCI', 'SCIO' is selected.

The right pane, 'Pin Configuration', has a 'Cycle Pin Group' button and a table with columns: Name, Value, Lock, and Link. The table shows the following configuration:

Name	Value	Lock	Link
Pin Group Selection	Mixed		
Operation Mode	Simple I2C		
Input/Output			
TXD0	None		
RXD0	None		
SCK0	None		
CTS0	None		
SDA0	✓ P411		
SCL0	✓ P410		
CTSRTS0	None		

Below the table, there is a section for 'Module name: SCIO' and 'Usage: When using Simple I2C mode, ensure port pins output type is n-ch open drain. When switching between I2C and other modes, first disable.'

At the bottom, there are tabs for 'Pin Function', 'Pin Number', 'Summary', 'BSP', 'Clocks', 'Pins', 'Interrupts', 'Event Links', 'Stacks', and 'Components'. The 'Pins' tab is currently selected.

## (c) IRQ 端子変更

割り込み信号回路の端子設定に関して、「6.5 割り込み信号回路に関する注意」を参照してください。  
センサの割り込み信号端子に合わせて、IRQ 端子を設定してください。

EK-RA6M4 ボードでは、Pmod1 #1 に P301 IRQ06、Pmod2 #7 に P414 IRQ09 が割り当てられています。

そのため、IRQ 端子を以下のように設定します。

- Pmod1 (Option Type 6A)使用時 : IRQ06 端子を P301 に設定
- Pmod2 使用時 (Interposer Board を利用) : IRQ09 端子を P414 に設定

**Pin Configuration** Generate Project Content

Select Pin Configuration Export to CSV file Configure Pin Driver Warnings

RA6M4 EK Manage configurations...  Generate data: g\_bsp\_pin\_cfg\_6m4

Pin Selection	Pin Configuration	Lock	Link
> Other Pins	Name		
> ✓ Peripherals	IRQ00	None	None
> ✓ Analog:ADC	IRQ01	None	None
> ✓ Analog:ANALOG	IRQ02	None	None
> Analog:ANALOG	IRQ03	None	None
> Analog:DAC	IRQ04	None	None
> Connectivity:CAN	IRQ05	None	None
> ✓ Connectivity:ETHERC	IRQ06	✓ P409	None
> ✓ Connectivity:IIC	IRQ07	None	None
> ✓ Connectivity:SCI	IRQ08	✓ P002	None
> ✓ Connectivity:SPI	IRQ09	✓ P414	None
> Connectivity:SSI	IRQ10	✓ P005	None
> ✓ Connectivity:USB	IRQ11	✓ P006	None
> Input:CTSU	IRQ12	✓ P008	None
> ✓ Input:ICU	IRQ13	None	None
> ✓ ICU0	IRQ14	None	None
> Monitoring:CAC			
> ✓ Storage:OSPI			
> ✓ Storage:QSPI			
> Storage:SDHI			

Module name: ICU0  
Usage: To use IRQ function with output or peripheral modes, change directly in port dialog

Pin Function | Pin Number

Summary | BSP | Clocks | Pins | Interrupts | Event Links | Stacks | Components

## (d) 汎用 I/O ポート端子変更 : RESET

RESET 信号回路の端子設定に関して、「6.6 RESET 信号回路に関する注意」を参照してください。

RESET 信号端子に合わせて、端子を設定してください。

EK-RA6M4 ボードでは、Pmod1 #2 に P203 Port、Pmod2 #4 に P412 Port が割り当てられています。

そのため、RESET 端子を以下のように設定します。

- Pmod1(Option Type 6A)使用時 : P203 を GPIO Port 出力端子に設定
- Pmod2 使用時 (Interposer Board を利用) : P412 を GPIO Port 出力端子に設定  
共に"Mode"は"Output mode (Initial High)"を選択してください。

The screenshot shows the Pin Configuration interface for the RA6M4 EK board. The 'Generate data' checkbox is checked, and the data file is named 'g\_bsp\_pin\_cfg\_6m4'. The 'Pin Selection' pane shows P412 selected. The 'Pin Configuration' table shows the following details for P412:

Name	Value	Link
Symbolic Name	PMOD2_CLK	
Comment		
Mode	Output mode (Initial High)	
Pull up	None	
Drive Capacity	Low	
Output type	CMOS	
Input/Output	GPIO	

Additional details for P412:

- Module name: P412
- Port Capabilities: AGT1: AGTEE1, CTSU0: TS08, ETHERC\_MII0: ETO\_ETXD0

デバイス変更により、「Generate data」が無効状態になります。次ページに有効化方法を説明します。

The screenshot shows the Pin Configuration interface for the RA6M4 EK board. The 'Generate data' checkbox is now unchecked and highlighted with a red box, indicating it is disabled. The rest of the interface remains the same as in the previous screenshot.

端子設定の生成を有効にするには、[Generate data]にチェックを入れ、テキストボックスに任意の名称を入力してください。

入力された名称は設定した端子構成に紐付けられるため、他の端子構成と重複しないよう、固有の名称としてください。

以下は、名称を”g\_bsp\_pin\_cfg\_6m4”にする例です。

The screenshot shows the 'Pin Configuration' software interface. At the top right, there is a 'Generate Project Content' button. Below it, there are two buttons: 'Export to CSV file' and 'Configure Pin Driver Warnings'. The 'Select Pin Configuration' section shows a dropdown menu set to 'RA6M4 EK' and a 'Manage configurations...' link. A red box highlights the 'Generate data:' checkbox, which is checked, and the text box next to it containing 'g\_bsp\_pin\_cfg\_6m4'. Below this, there is a 'Pin Selection' tree view on the left and a 'Pin Configuration' table on the right. The table has columns for 'Name', 'Value', and 'Link'. At the bottom, there are tabs for 'Pin Function', 'Pin Number', 'Summary', 'BSP', 'Clocks', 'Pins', 'Interrupts', 'Event Links', 'Stacks', and 'Components'.

(4) Stacks

Stacks タブで、各コンポーネント設定を変更します。

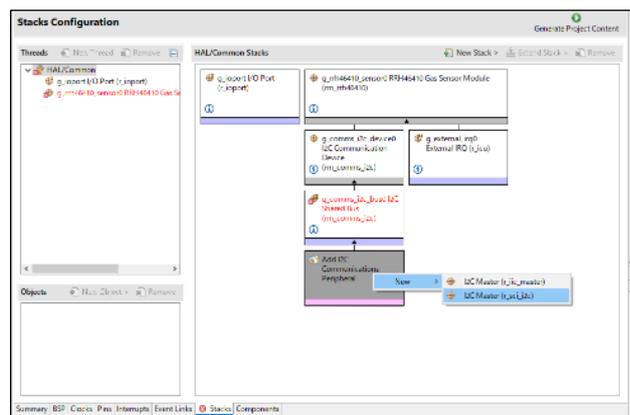
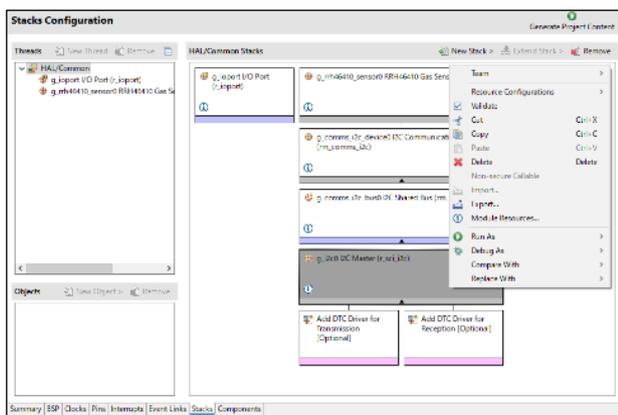
(a) COMMS\_I2C と I2C ドライバ設定変更

使用するボードに合わせて、COMMS\_I2C と I2C ドライバの設定を変更してください。

I2C I/F の端子を使用する場合は、不要な stack を削除し、新たに使用する stack を追加してください。

表 6-1 EK-RA6M4 における I2C I/F とチャンネルの設定

EK-RA6M4	I2C I/F	g_comms_i2c_bus0 I2C Shared Bus (rm_comms_i2c)	g_i2c_master0 I2C Master
Pmod1 Option Type 6A	IIC1	Channel: 1	Pins を確認
Pmod2 Type 2A	SCI0	Channel: 0	Pins を確認



g_comms_i2c_bus0 I2C Shared Bus (rm_comms_i2c)		
Settings	Property	Value
API Info	Common	
	Parameter Checking	Default (BSP)
	Module g_comms_i2c_bus0 I2C Shared Bus (rm_comms_i2c)	
	Name	g_comms_i2c_bus0
	Bus Timeout	0xFFFFFFFF
	Semaphore for Blocking (RTOS only)	Use
	Recursive Mutex for Bus (RTOS only)	Use
Channel	1	
Rate	Standard	

g_i2c_master0 I2C Master (r_iic_master)		
Settings	Property	Value
API Info	Common	
	Parameter Checking	Default (BSP)
	DTC on Transmission and Reception	Disabled
	10-bit slave addressing	Disabled
	Module g_i2c_master0 I2C Master (r_iic_master)	
	Name	g_i2c_master0
	Channel	1
	Rate	Standard
	Custom Rate (bps)	0
	Rise Time (ns)	120
	Fall Time (ns)	120
	Duty Cycle (%)	50
	Slave Address	0x00
	Address Mode	7-Bit
	Timeout Mode	Short Mode
	Timeout during SCL Low	Enabled
	Callback	rm_comms_i2c_callback
	Interrupt Priority Level	Priority 12
	Pins	
SDA1	P511	
SCL1	P512	

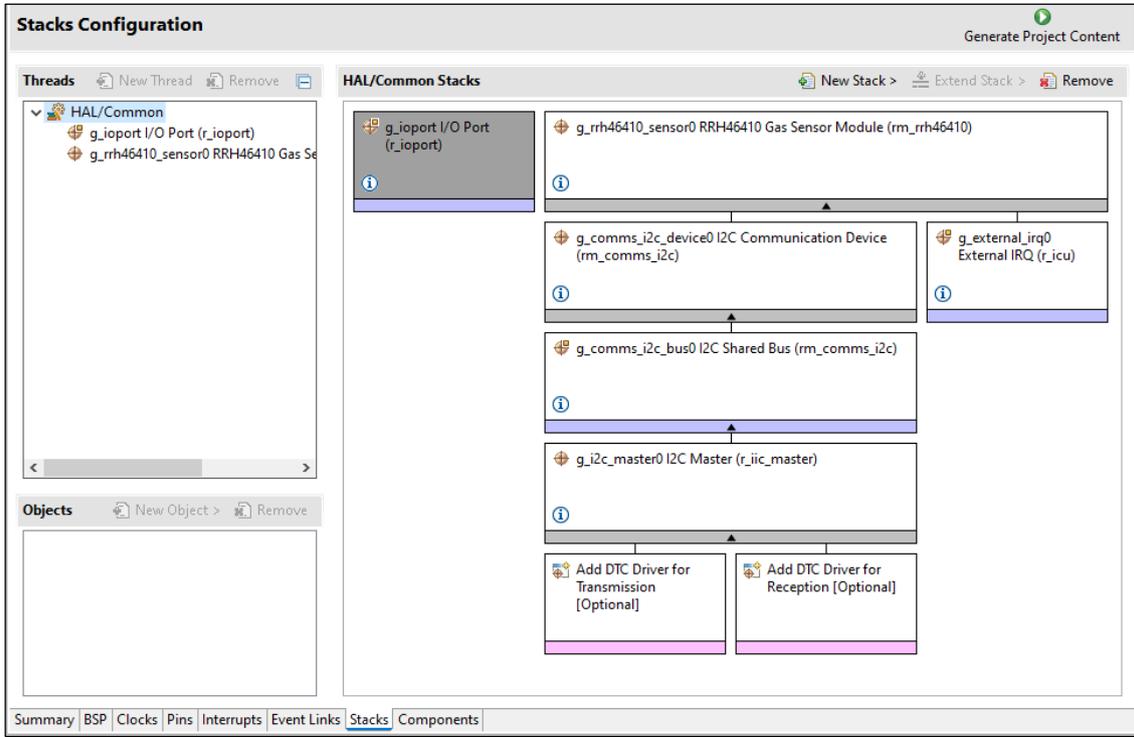
g_i2c0 I2C Master (r_sci_i2c)		
Settings	Property	Value
API Info	Common	
	Parameter Checking	Default (BSP)
	DTC on Transmission and Rec	Disabled
	10-bit slave addressing	Disabled
	Module g_i2c0 I2C Master (r_sci_	
	Name	g_i2c0
	Channel	0
	Slave Address	0x00
	Address Mode	7-Bit
	Rate	Standard
	Custom Rate (bps)	0
	SDA Output Delay (nano secc)	300
	Noise filter setting	Use clock signal divided by 1 with noise filter
	Bit Rate Modulation	Enable
	Callback	rm_comms_i2c_callback
	Interrupt Priority Level	Priority 2
	RX Interrupt Priority Level [O]	Disabled
	Pins	
	SDA0	P411
	SCL0	P410

## (b) 汎用 I/O ポートドライバ設定変更 : RESET

RESET 信号制御のソース変更方法については、「6.2.2(1) RESET 信号制御」を参照してください。

“g\_ioport I/O Port”の Pin Configuration Name に、使用する端子構成の名称を入力してください。

以下は、名称を“g\_bsp\_pin\_cfg\_6m4”にする例です。



g_ioport I/O Port (r_ioport)		
Settings	Property	Value
API Info	Common	
	Parameter Checking	Default (BSP)
	Module g_ioport I/O Port (r_ioport)	
	Name	g_ioport
	1st Port ELC Trigger Source	Disabled
	2nd Port ELC Trigger Source	Disabled
	3rd Port ELC Trigger Source	Disabled
	4th Port ELC Trigger Source	Disabled
	Pin Configuration Name	g_bsp_pin_cfg_6m4
	Pins	
	TCK	P300
	TDI	P110
	TDO	P109
	TMS	P108
	SWCLK	<unavailable>
	SWDIO	<unavailable>
	TRACESWO	<unavailable>
	TCLK	<unavailable>
	TDATA0	<unavailable>
	TDATA1	<unavailable>
	TDATA2	<unavailable>

## (c) IRQ ドライバ設定変更

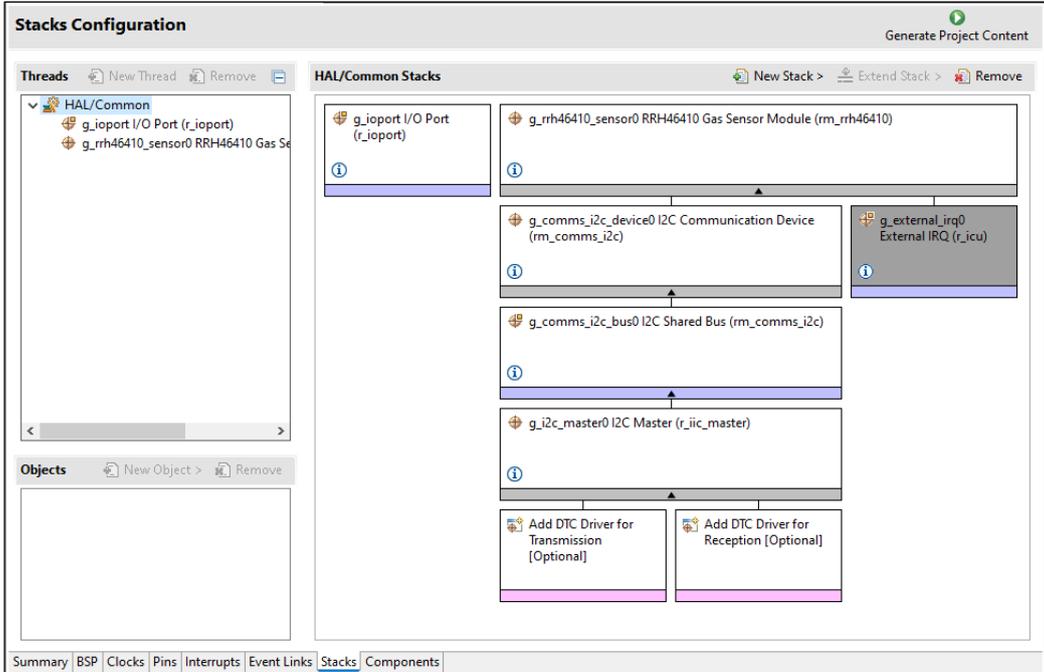
割り込み信号回路の使用に関して、「6.5 割り込み信号回路に関する注意」を参照してください。

使用するボードに合わせて、`r_icu` の設定を変更してください。

EK-RA6M4 ボードでは、Pmod1 #1 に P301 IRQ6、Pmod2 #7 に P414 IRQ9 が割り当てられています。

そのため、IRQ channel を以下のように設定します。

- Pmod1 (Option Type 6A)使用時 : "Channel"を 6 に設定
- Pmod2 使用時 (Interposer Board を利用) : "Channel"を 9 に設定



Property	Value
Settings	
API Info	
Common	
Parameter Checking	Default (BSP)
Module <code>g_external_irq0 External IRQ (r_icu)</code>	
Name	<code>g_external_irq0</code>
Channel	9
Trigger	Falling
Digital Filtering	Disabled
Digital Filtering Sample Clock (Only valid when Digital Filtering is Enabled)	PCLK / 64
Callback	<code>rm_rrh46410_irq_callback</code>
Pin Interrupt Priority	Priority 2
Pins	
IRQ09	P414

他 stack でエラーが表示されている場合は、表示されたエラーに従って指定の項目を変更してください。

## (5) コード生成

修正完了後、[Generate Project Content]を押下して、ファイルを生成します。

「6.2.2 サンプルソースの変更」を実施後、プロジェクトをビルドします。

メニューから[Debug Configurations]を選択し、使用するボードに接続するエミュレータに合わせて、Debugger の設定を変更してください。

## 6.2.2 サンプルソースの変更

### (1) RESET 信号制御

端子の設定については、「6.2.1(3)(d) 汎用 I/O ポート端子変更：RESET」を参照してください。

hal\_entry.c (Non-OS)または、rrh46410\_sensor\_thread\_entry.c (FreeRTOS) を開き、センサリセット制御処理を変更してください。また、適切なりセット制御論理処理に変更してください。

使用するボードに合わせて RESET 端子の指定を修正してください。

以上の設定により、EK-RA6M4 ボードでは、以下のように設定されています。

- Pmod1 (Option Type 6A)使用時：RESET 端子を P203 に割り当て
- Pmod2 使用時 (Interposer Board を利用)：RESET 端子を P412 に割り当て

```

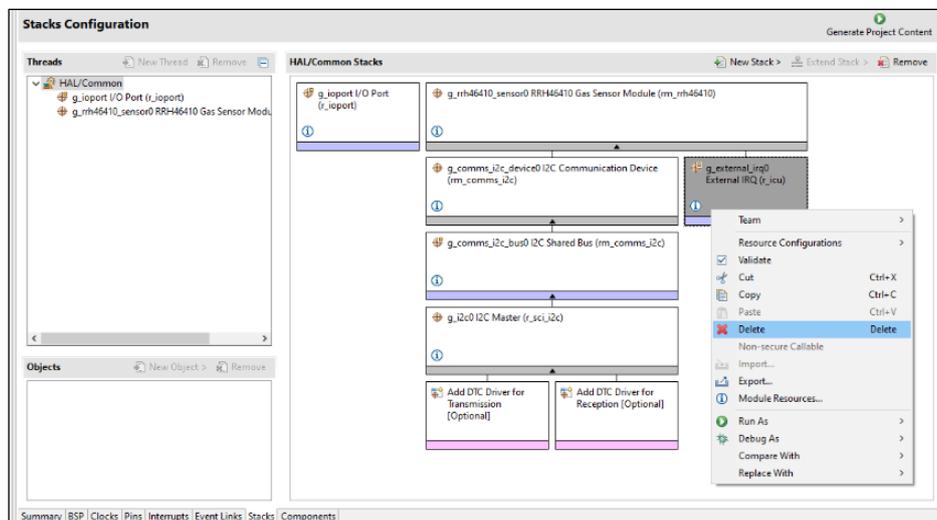
/* Reset RRH46410 sensor (active low). Please change to the IO port connected to
the RES_N pin of the RRH46410 sensor on the customer board. */
R_IOPORT_PinWrite(&g_ioport_ctrl, BSP_IO_PORT_04_PIN_12, BSP_IO_LEVEL_HIGH);
R_BSP_SoftwareDelay(10, BSP_DELAY_UNITS_MILLISECONDS);
R_IOPORT_PinWrite(&g_ioport_ctrl, BSP_IO_PORT_04_PIN_12, BSP_IO_LEVEL_LOW);
R_BSP_SoftwareDelay(10, BSP_DELAY_UNITS_MILLISECONDS);
R_IOPORT_PinWrite(&g_ioport_ctrl, BSP_IO_PORT_04_PIN_12, BSP_IO_LEVEL_HIGH);
R_BSP_SoftwareDelay(10, BSP_DELAY_UNITS_MILLISECONDS);

```

### 6.2.3 IRQ を使用しない場合の変更

IRQ を使用しない場合は stack の削除、定数の値の変更を行います。

Stacks タブで、"g\_external\_irq0 External IRQ"を削除してください。



RA\_RRH46410.c (Non-OS)または、rrh46410\_sensor\_thread\_entry.c (FreeRTOS) を開き、G\_RRH46410\_SENSOR0\_IRQ\_ENABLE の値を 0 に変更してください。

```

/* TODO: Enable if you want to open RRH46410 */
#define G_RRH46410_SENSOR0_IRQ_ENABLE (0)

```

### 6.2.4 ツールチェーン設定変更

GCC ARM Embedded ツールチェーン以外のツールチェーンを使用する場合は、本プロジェクトから RA\_RRH46410.c (Non-OS)または、rrh46410\_sensor\_thread\_entry.c および sensor\_thread\_common.c、sensor\_thread\_common.h (FreeRTOS)をコピーしてプロジェクトを作成してください。

## 6.3 RX サンプルプロジェクト

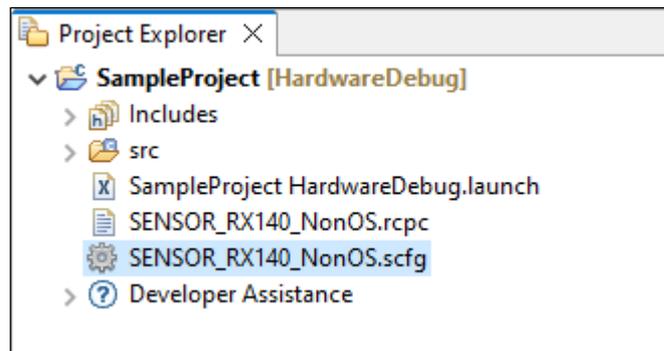
サンプルプロジェクトをインポート後に以下の手順を実行してください。インポートの方法は「6.1 サンプルプロジェクトのインポート」を参照してください。

以下のボード変更例での変更手順を説明します。なお、Pmod Type 2A/3A コネクタ使用時は Interposer Board が必要です。

- サンプルプロジェクト “RRH46410\_RX140\_NonOS”:  
Pmod1 (Type 2A: SCI5)  
→ EK-RX671 ボードの Pmod1 (Option Type 6A: RIIC0) もしくは Pmod2 (Option Type 6A: SCI11)

### 6.3.1 Smart Configurator 設定の変更

インポートしたプロジェクトの.scfg ファイルをダブルクリックし、Smart Configurator を表示します。



#### (1) Board

1. Board タブで [...] ボタンを押下します。

**Device selection**

**Device selection**

Board:  ...

Device:

[Download more boards...](#)

**Feature Selection**

To add a component, make the selection from the table below and click on the "Add" button.  
The configurations for each added component can be further configured in the "Components" page.

Features	Components	Action	Link
Application Header	Ports	Add	
LEDs	Ports	Add	
PMOD 1 (UART/SPI/IIC)	▼ SCI Driver - UART (r_sci_rx)	Add	
PMOD 2 (UART/SPI/IIC)	▼ SCI Driver - UART (r_sci_rx)	Add	
User Switches	IRQ Driver (r_irq_rx)	Apply	

Overview

Board

Clocks

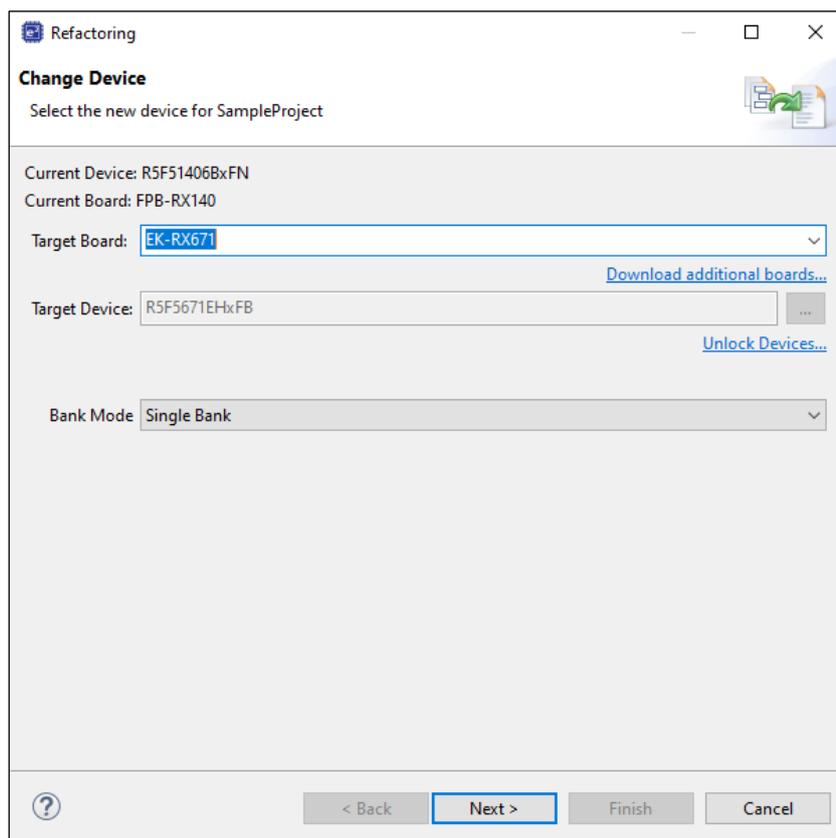
System

Components

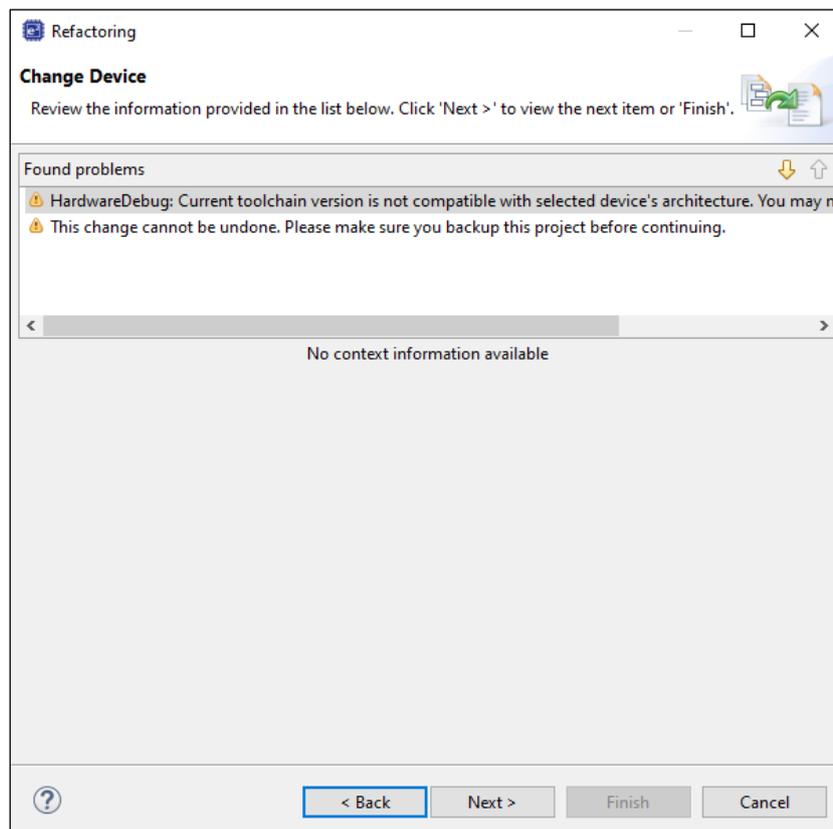
Pins

Interrupts

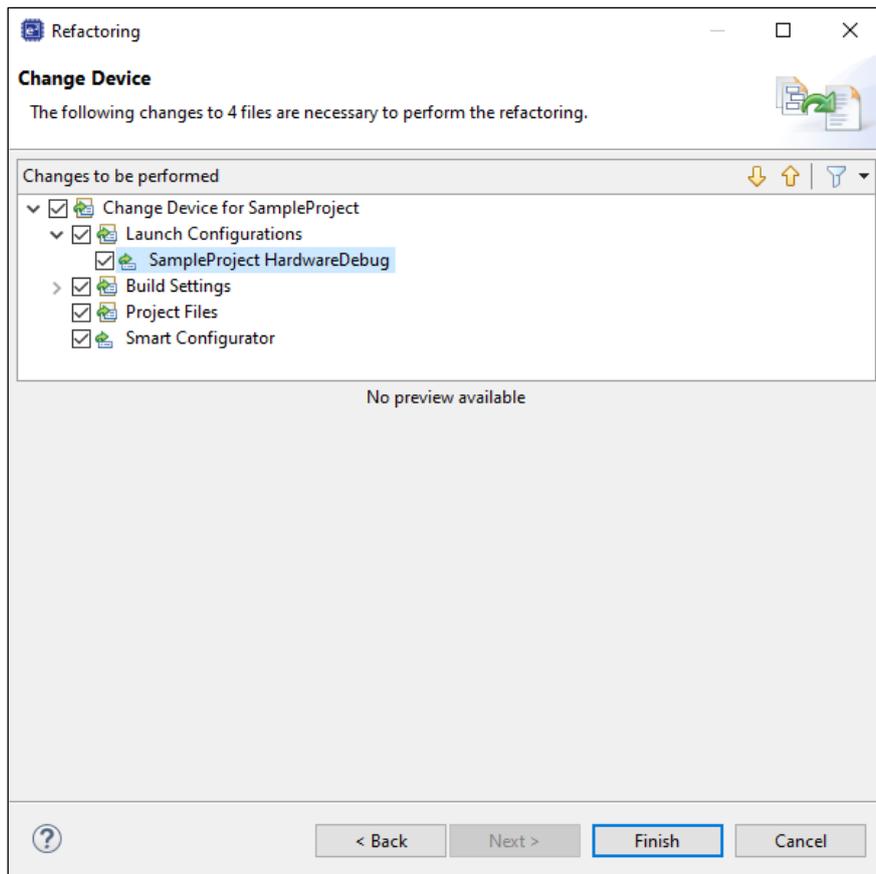
2. Change Device ウィンドウで、変更したいボードもしくは、デバイスを選択し、[Next]ボタンを押下します。



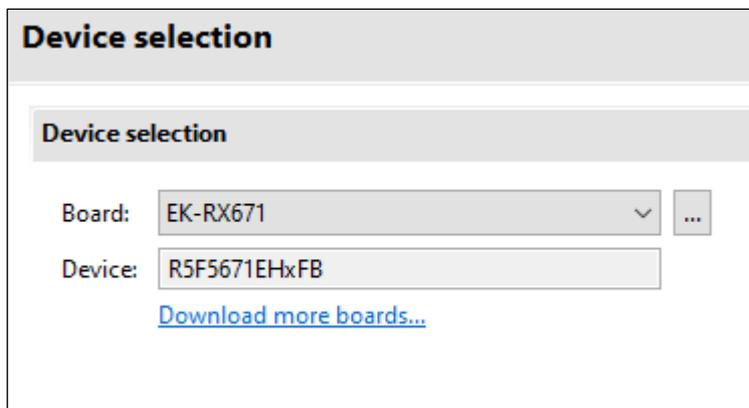
3. Warning が表示された場合、内容を確認し問題無ければ、[Next]を押下します。



4. 変更内容が表示されるため、[Finish]ボタンを押下して、変更を実行します。



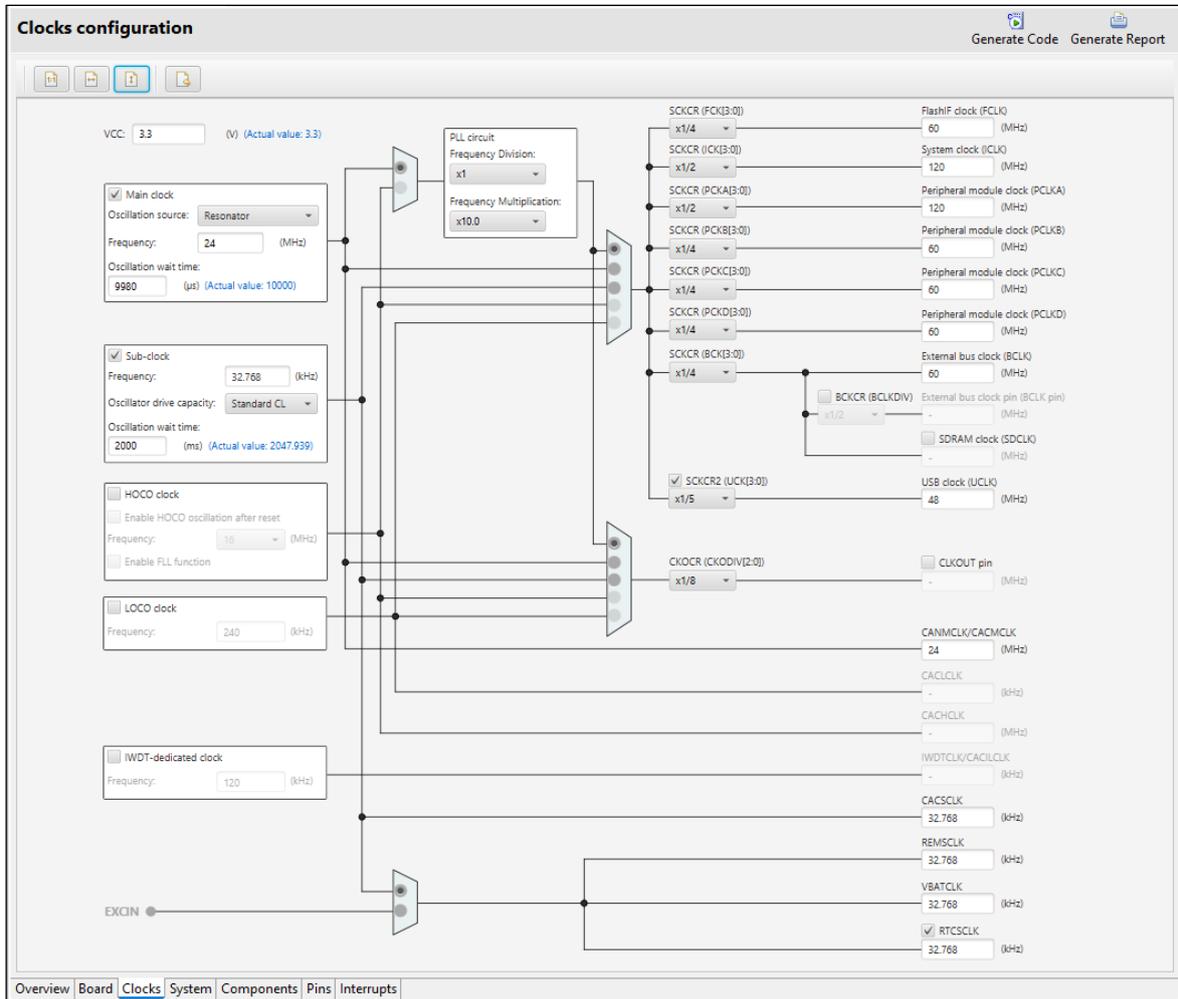
5. Board タブで、変更したボード、デバイスに変更されていることを確認します。



(2) Clocks

Board を”Custom User Board (Any Device)”に変更した場合は、使用するボードに合わせてクロック設定を変更してください。

Board をルネサス製ボードに変更した場合は、自動的に設定が変更されます。



## (3) Components

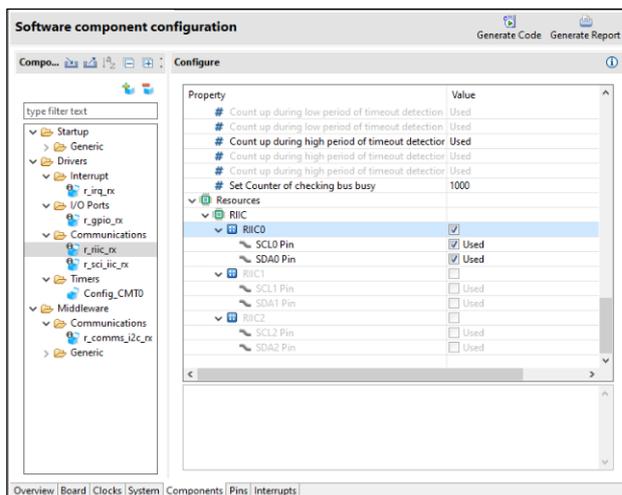
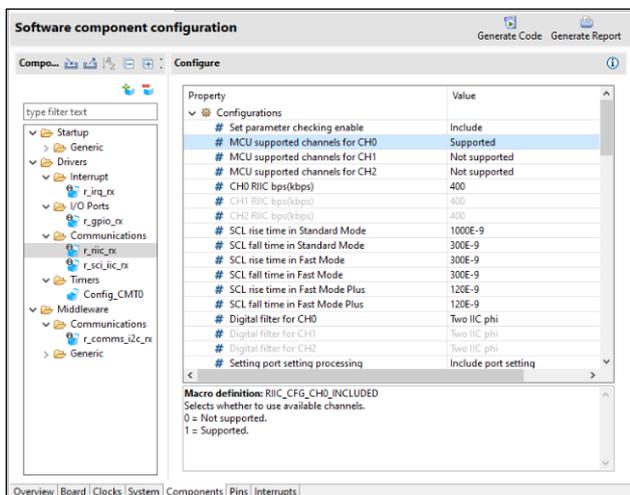
Components タブで、使用するボードに合わせて、各 component の設定を変更します。

## (a) I2C ドライバ設定変更

EK-RX671 では、Pmod1 に RIIC0、Pmod2 に SCI11 が割り当てられています。

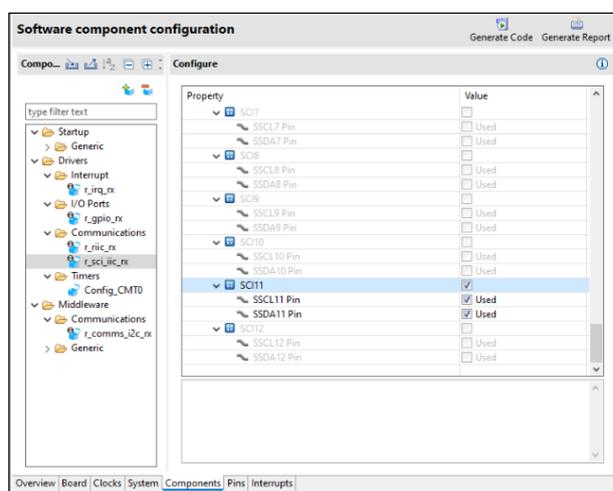
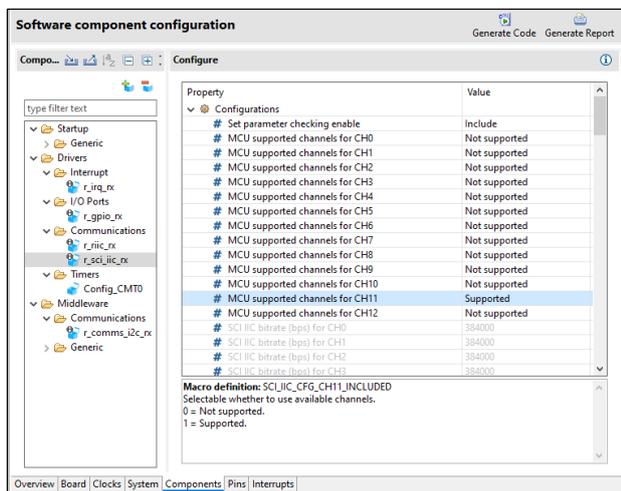
Pmod1 (Option Type 6A)を使用する場合は、以下のように設定してください。

- r\_riic\_rx の”MCU supported channels for CH0”を”Supported”に設定
- r\_riic\_rx の”Resources”の”RIIC0”, “SCL0 Pin”, “SDA0 Pin”をチェック



Pmod2 (Option Type 6A)を使用する場合は、以下のように設定してください。

- r\_sci\_iic\_rx の”MCU supported channels for CH5”を”Not supported”に設定、  
”MCU supported channels for CH11”を”Supported”に設定
- r\_sci\_iic\_rx の”Resources”の”SCI11”, “SSCL11 Pin”, “SSDA11 pin”をチェック

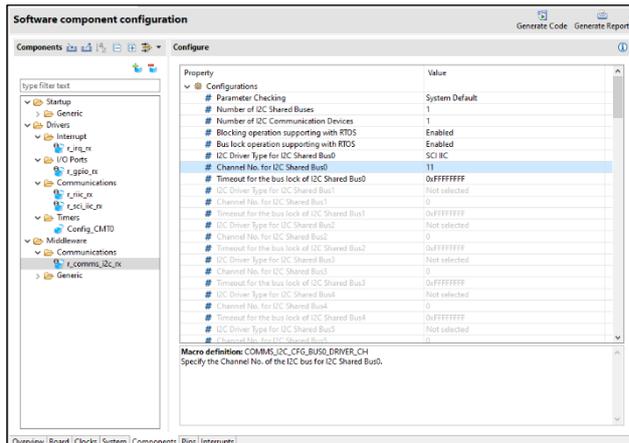
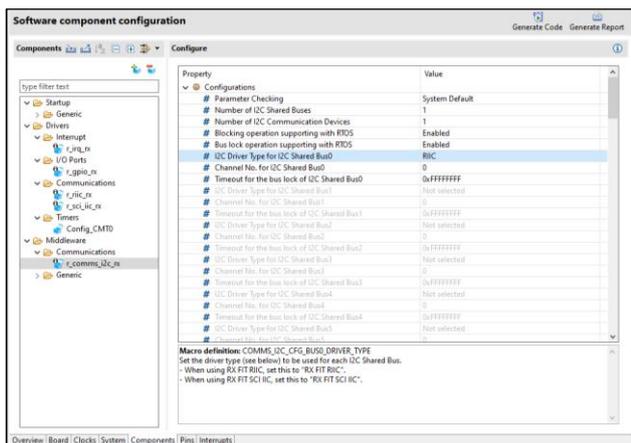


## (b) COMMS\_I2C 設定変更

I2C ドライバもしくはチャンネルを変更した場合、設定を変更する必要があります。

`r_comms_i2c_rx` の I2C Driver Type for I2C Shared BusX (X: Bus No.)と Channel No. for I2C Shared BusX (X: Bus No.)を以下のように設定します。

- Pmod1 (Option Type 6A)を使用する場合 :  
 "I2C Driver Type for I2C Shared Bus0"を"RIIC"に設定  
 "Channel No. for I2C Shared Bus0"を"0"に設定
- Pmod2 (Option Type 6A)を使用する場合 :  
 "I2C Driver Type for I2C Shared Bus0"を"SCI\_IIC"に設定  
 "Channel No. for I2C Shared Bus0"を"11"に設定



(c) IRQ ドライバ設定変更

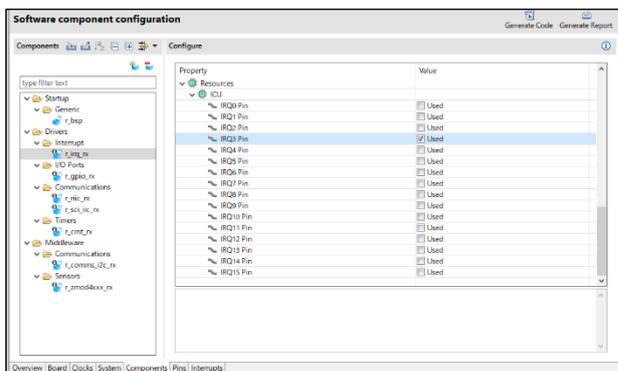
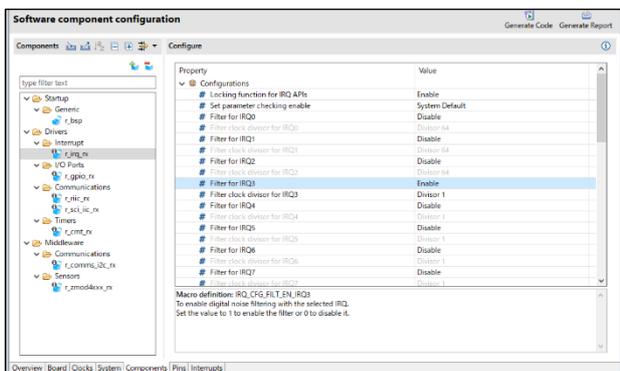
割り込み信号回路の使用に関して、「6.5 割り込み信号回路に関する注意」を参照してください。

割り込みの設定に関しては、センサ制御モジュールで共通の IRQ ドライバを使用するため、ZMOD4XXX FIT モジュール(r\_zmod4xxx\_r)を例に説明します。以下では、ZMOD4XXX を RRH46410 として扱ってください。

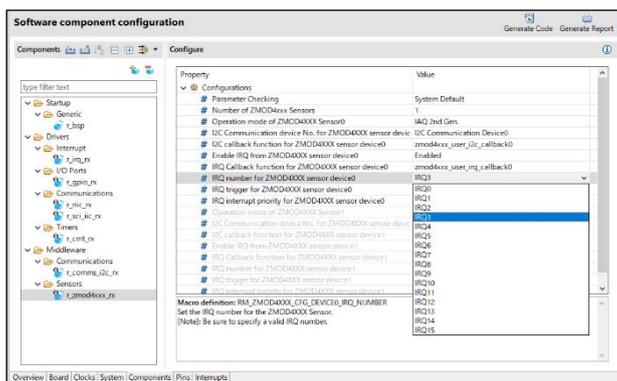
EK-RX671 ボードでは、Pmod1 #1 に P83 IRQ3、Pmod2 #1 に P74 IRQ12 が割り当てられています。

そのため、r\_irq\_rx の IRQ 端子を以下のように設定します。また、不要な設定を無効化します。

- Pmod1 (Option Type 6A)を使用する場合：
  - “Filter for IRQ4”を”Disable”に変更
  - “Filter for IRQ3”を”Enable”に変更
  - “Resources”の”IRQ4 pin”のチェックを削除
  - “Resources”の”IRQ3 pin”のチェックを追加
- Pmod2 (Option Type 6A)を使用する場合：
  - “Filter for IRQ4”を”Disable”に変更
  - “Filter for IRQ12”を”Enable”に変更
  - “Resources”の”IRQ4 pin”のチェックを削除
  - “Resources”の”IRQ12 pin”のチェックを追加



r\_zmod4xxx\_rx の”IRQ number for ZMOD4XXX sensor device0”を”IRQ3”もしくは”IRQ12”に変更します。RRH46410 の場合、同様に”IRQ number for RRH46410 sensor device0”があります。同様に変更します。



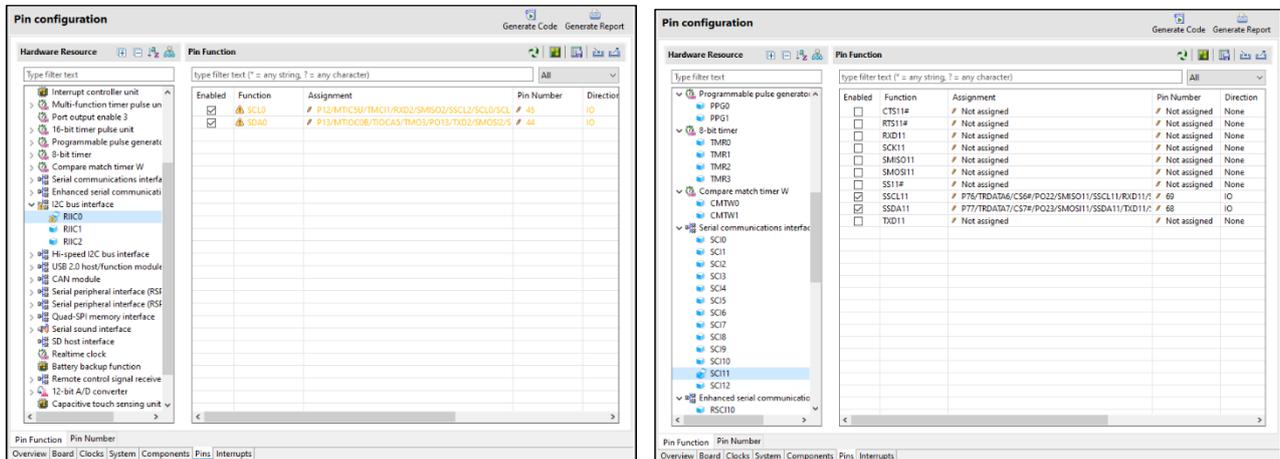
## (4) Pins

## (a) I2C I/F 端子変更

EK-RX671 ボードでは、Pmod1 に RIIC0、Pmod2 に SCI11 が割り当てられています。

そのため、Pins タブの Pin Function で I2C 通信端子を以下のように設定します。

- Pmod1 (Option Type 6A)使用時 : RIIC0 と P12 SCL0 と P13 SDA0 を有効化
- Pmod2 (Option Type 6A)使用時 : SCI11 と P76 SSCL11 と PB7 SSDA11 を有効化



EK-RX671 のボード情報は、Pmod1 を High-Speed I2C Bus Interface (RIICH5)で使用するよう割り当てられているため、RIIC で使用することは、Warning が表示されますが問題ありません。

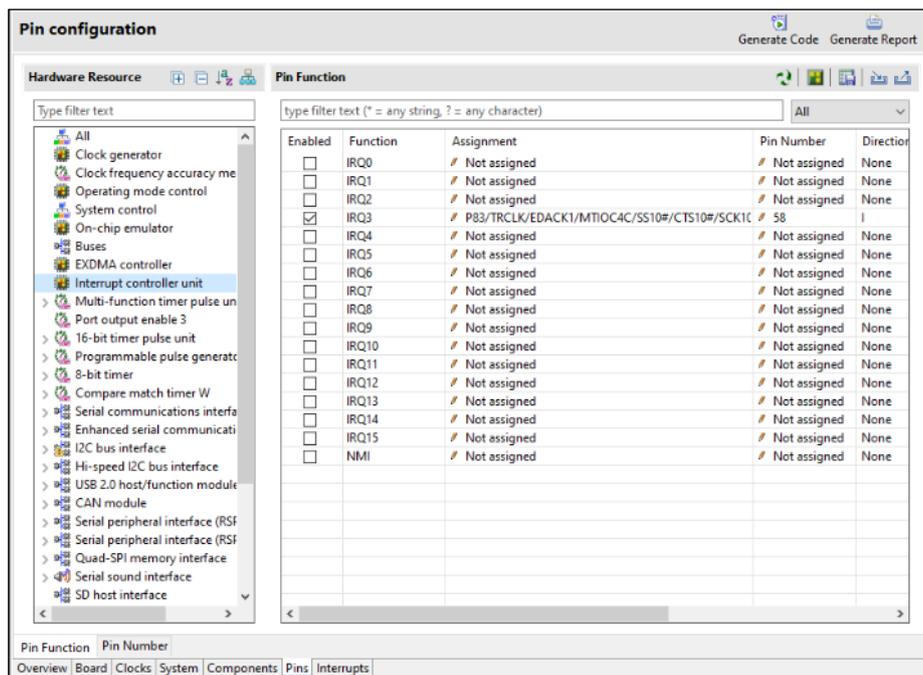
## (b) IRQ 端子変更

割り込み信号回路の端子設定に関して、「6.5 割り込み信号回路に関する注意」を参照してください。

EK-RX671 ボードでは、Pmod1 #1 に P83 IRQ3、Pmod2 #1 に P74 IRQ12 が割り当てられています。

そのため、Pins タブの Pin Function で Interrupt controller unit を選択し、以下のように設定します。

- Pmod1 (Option Type 6A)使用時 : IRQ3 の有効化と IRQ3 端子を P83 に設定
- Pmod2 (Option Type 6A)使用時 : IRQ12 の有効化と IRQ12 端子を P74 に設定

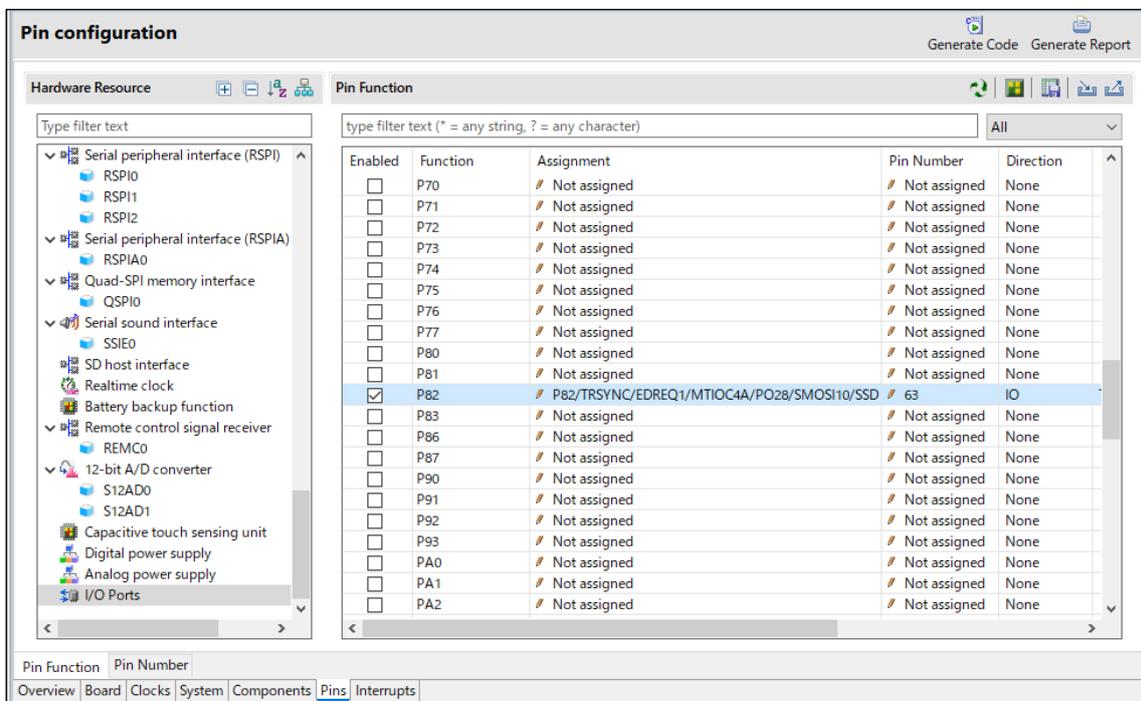


## (c) 汎用 I/O ポート端子変更 : RESET

RESET 信号回路の端子設定に関して、「6.6 RESET 信号回路に関する注意」を参照してください。

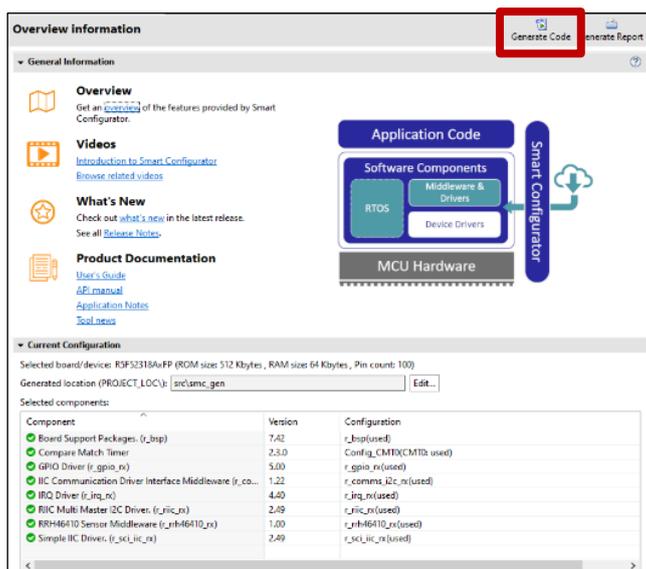
EK-RX671 ボードでは、RESET 端子が Pmod1 #2 P82 Port、Pmod2 #2 P77 Port が割り当てられています。そのため、Pins タブの Pin Function で I/O Ports を選択し、以下のように設定します。

- Pmod1 (Option Type 6A)使用時 : P82 を有効化に設定
- Pmod2 (Option Type 6A)使用時 : P77 を有効化に設定



## (5) コード生成とビルドの実行

修正完了後、[Generate Code]アイコンを押下して、コード生成を行います。



「6.3.2 サンプルソースの変更」を実施後、プロジェクトをビルドします。

メニューから[Debug Configurations]を選択し、使用するボードに接続するエミュレータに合わせて、Debugger の設定を変更してください。

### 6.3.2 サンプルソースの変更

#### (1) RESET 信号制御

端子の設定については、「6.3.1(4)(c) 汎用 I/O ポート端子変更：RESET」を参照してください。

main.c を開き、センサリセット制御処理を変更してください。また、適切なリセット制御論理処理に変更してください。

使用するボードに合わせて RESET 端子の指定を修正してください。

以上の設定により、EK-RX671 ボードでは、以下のように設定されています。

- Pmod1 (Option Type 6A)使用時：RESET 端子を P82 に割り当て
- Pmod2 (Option Type 6A)使用時：RESET 端子を P77 に割り当て

```

/* Reset RRH46410 sensor (active low). Please change to the IO port
connected to the RES_N pin of the RRH46410 sensor on the customer board. */
R_GPIO_PinWrite(GPIO_PORT_8_PIN_2, GPIO_LEVEL_HIGH);
R_GPIO_PinDirectionSet(GPIO_PORT_8_PIN_2, GPIO_DIRECTION_OUTPUT);
R_BSP_SoftwareDelay(10, BSP_DELAY_MILLISECS);
R_GPIO_PinWrite(GPIO_PORT_8_PIN_2, GPIO_LEVEL_LOW);
R_BSP_SoftwareDelay(10, BSP_DELAY_MILLISECS);
R_GPIO_PinWrite(GPIO_PORT_8_PIN_2, GPIO_LEVEL_HIGH);
R_BSP_SoftwareDelay(10, BSP_DELAY_MILLISECS);

```

### 6.3.3 ツールチェイン設定変更

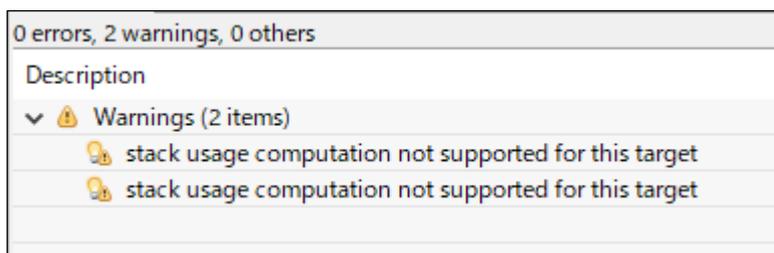
CC-RX ツールチェイン以外のツールチェインを使用する場合は、本プロジェクトから main.c と RX\_RRH46410.c (Non-OS)または、main.c と rrh46410\_sensor\_thread\_entry.c (FreeRTOS)をコピーしてプロジェクトを作成してください。

### 6.3.4 GCC ビルド時の注意

GCC プロジェクトをビルド時、以下の Warning が発生します。

これらの Warning は、コンパイラオプションにてスタックサイズの制限を指定した上で、対象部分（インラインアセンブラ処理部分）は、スタック使用量計算の対象外であるため発生しています。

Warning が表示されますが、動作に問題はありません。



### 6.3.5 IAR 社統合開発環境「IAR Embedded Workbench」利用時

RX スマート・コンフィグレータを利用して、ソースファイルを IAR Embedded Workbench にインポートできます。手順については、以下を参照してください。

[RX Smart Configurator User's Guide: IAREW](#)

[RX スマート・コンフィグレータ ユーザーガイド: IAREW 編](#)

## 6.4 RL78 サンプルプロジェクト

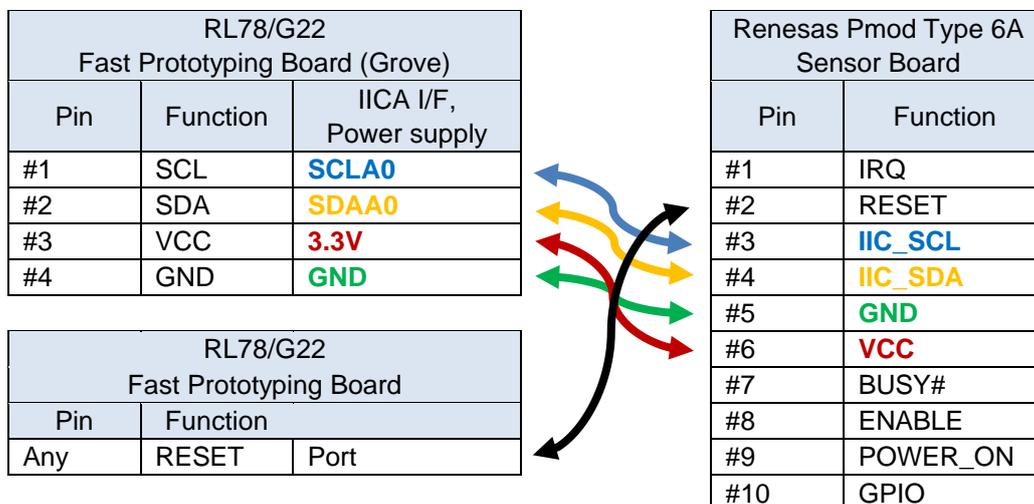
サンプルプロジェクトをインポート後に以下の手順を実行してください。インポートの方法は「6.1 サンプルプロジェクトのインポート」を参照してください。

以下のボード変更例での変更手順を説明します。

- サンプルプロジェクト “RRH46410\_RL78G23\_NonOS”:  
Pmod2 (Type 6A: IICA1)  
→ RL78/G22 Fast Prototyping Board の Grove (IICA0)  
割り込み制御有り  
→ 割り込み制御無し  
RESET 制御端子  
→ 任意の Port 端子に変更

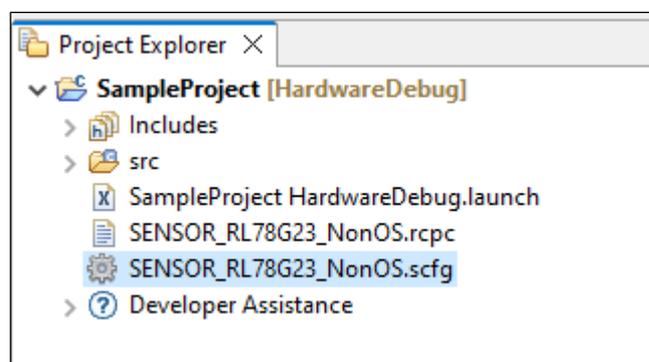
なお、Grove の VDD を 3.3V にするために、J17 を 2-3 に設定してください。

また、以下のようにジャンパ線で接続してください。



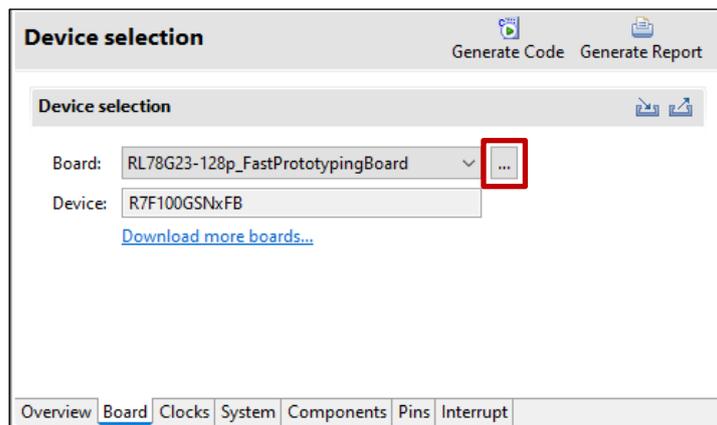
### 6.4.1 Smart Configurator 設定の変更

プロジェクトツリーで、インポートしたプロジェクトの.scfg ファイルをダブルクリックし、Smart Configurator を表示します。

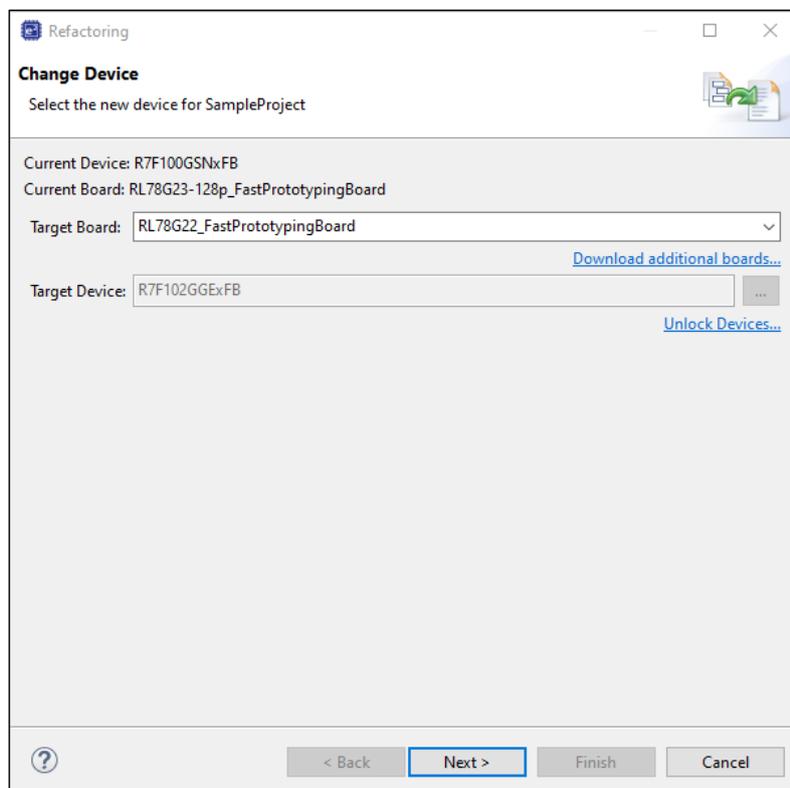


## (1) Board

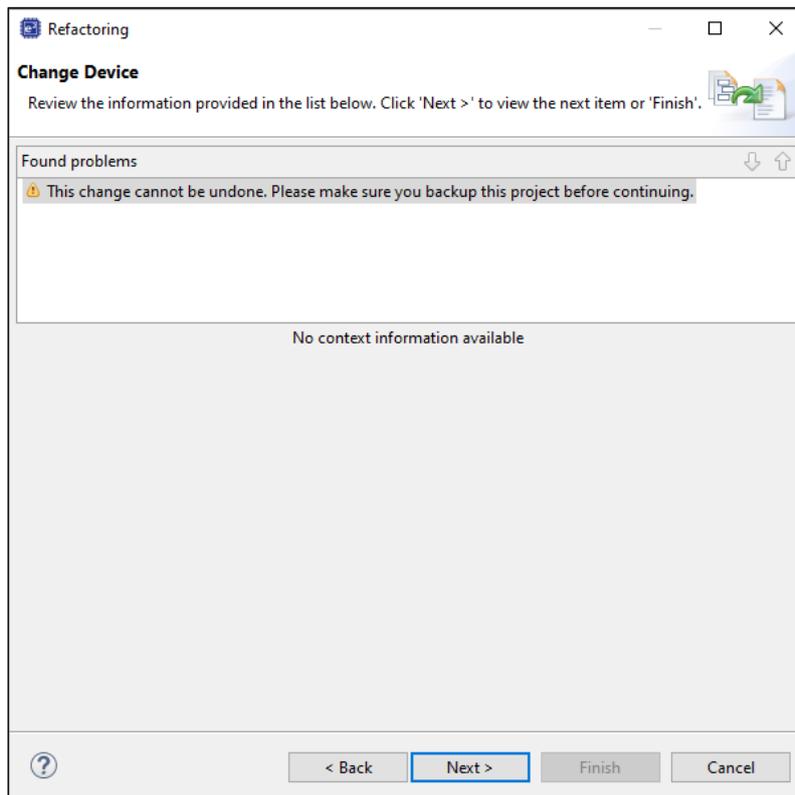
1. Board タブで、[...]ボタンを押下します。



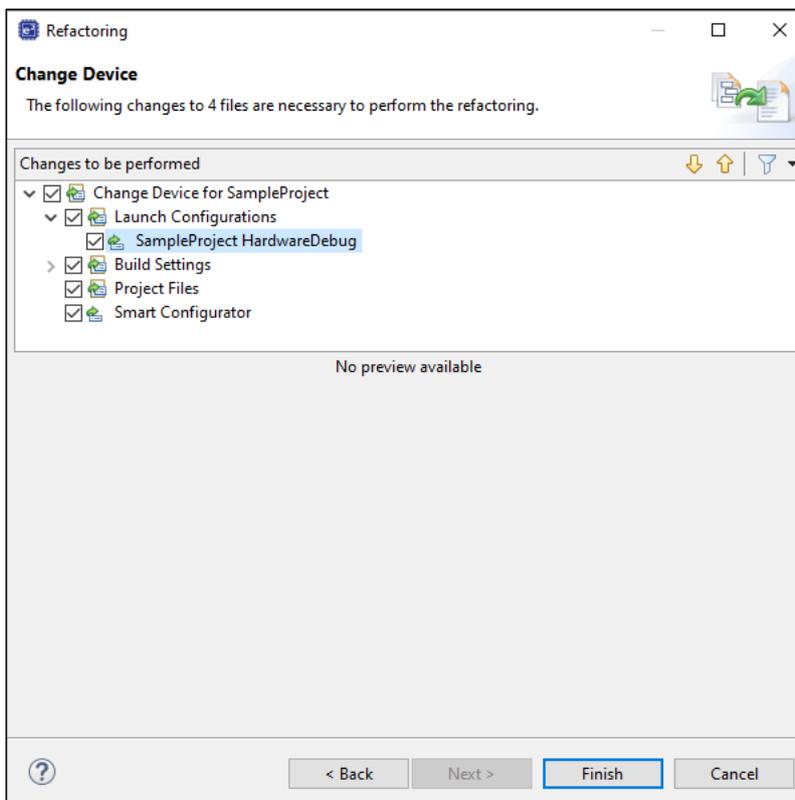
2. Change Device ウィンドウで、変更したいボードもしくは、デバイスを選択し、[Next]ボタンを押下します。



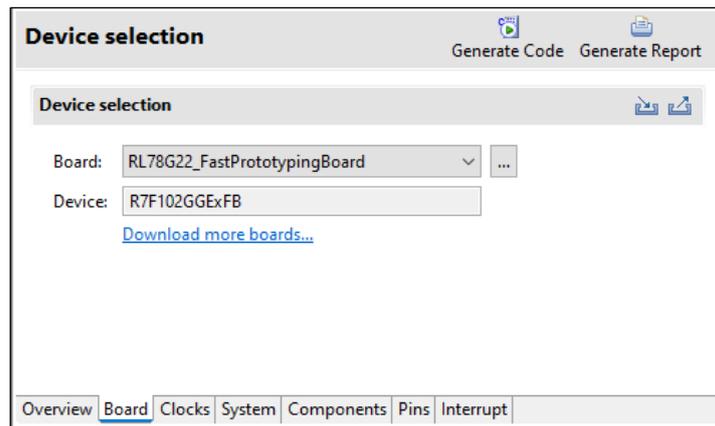
3. Warning が表示された場合、内容を確認し問題無ければ、[Next]を押下します。



4. 変更内容が表示されるため、[Finish]ボタンを押下して、変更を実行します。



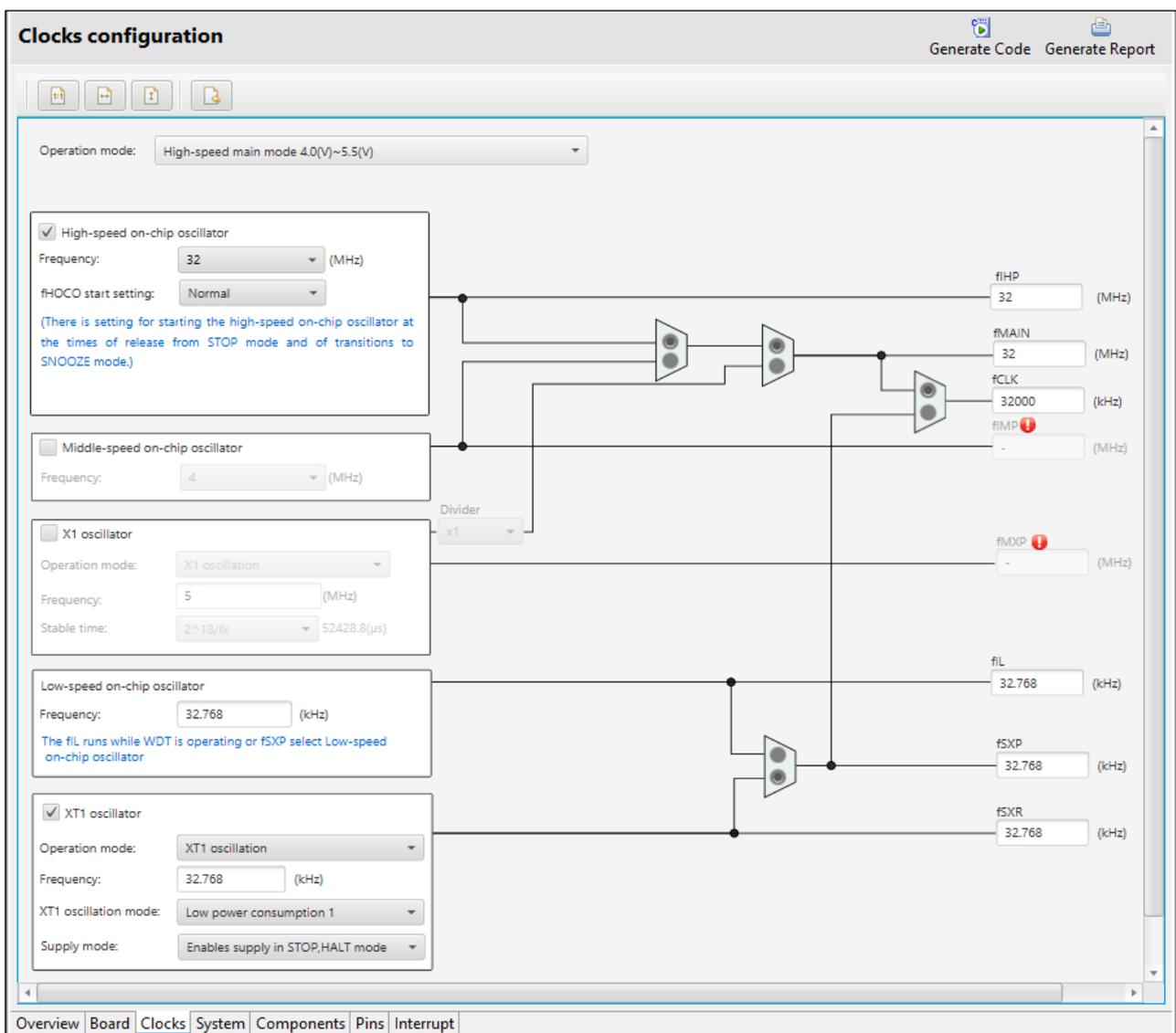
5. Board タブで、変更したボード、デバイスに変更されていることを確認します。



## (2) Clocks

Board を”Custom User Board (Any Device)”に変更した場合は、使用するボードに合わせてクロック設定を変更してください。

Board をルネサス製ボードに変更した場合は、自動的に設定が変更されます。



## (3) Components

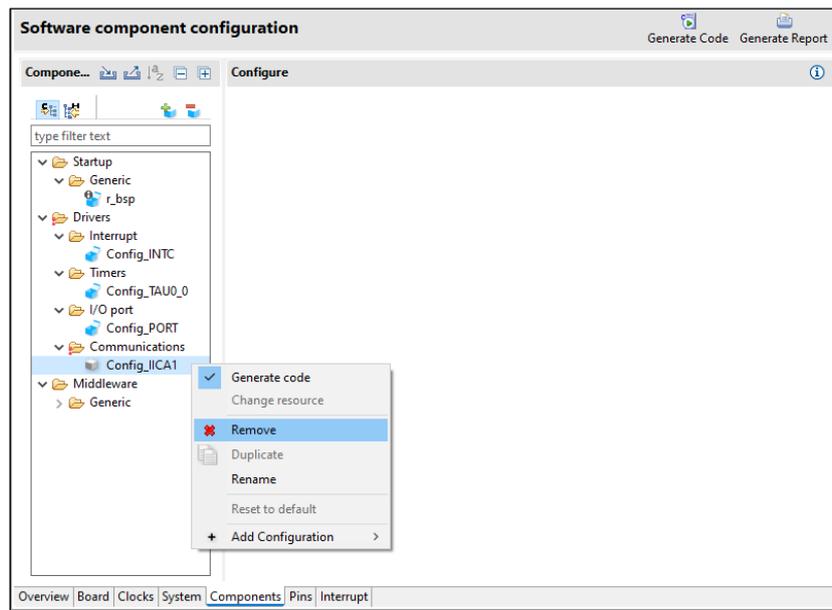
Components タブで、使用するボードに合わせて、各 component の設定を変更します。

## (a) I2C ドライバ設定変更

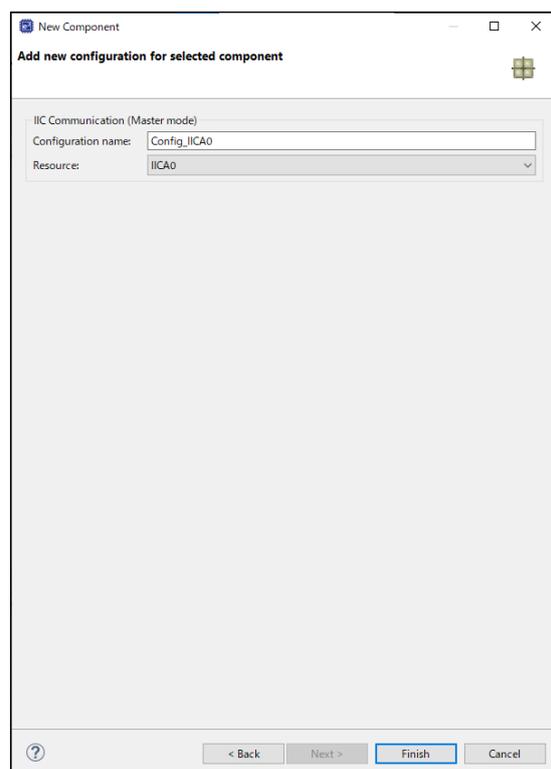
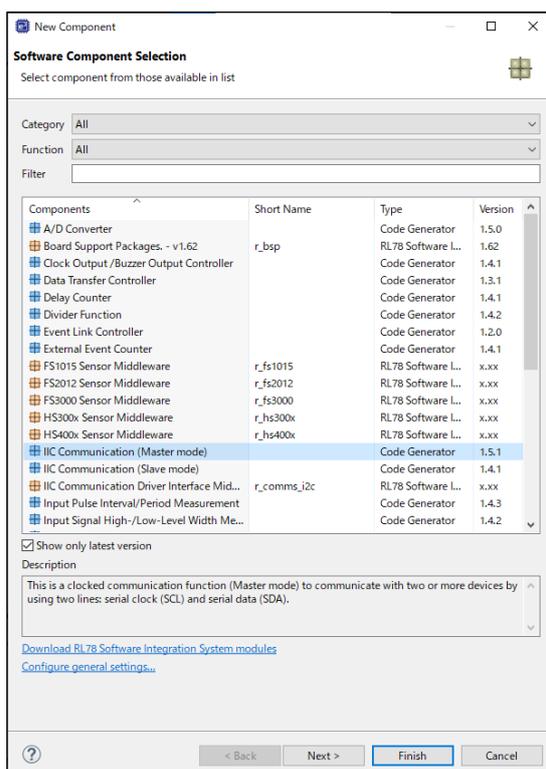
SAU は RRH46410 センサが要求するクロック・ストレッチ機能を未サポートのため使用できません。

I2C ドライバ設定の変更は以下の手順で行います。

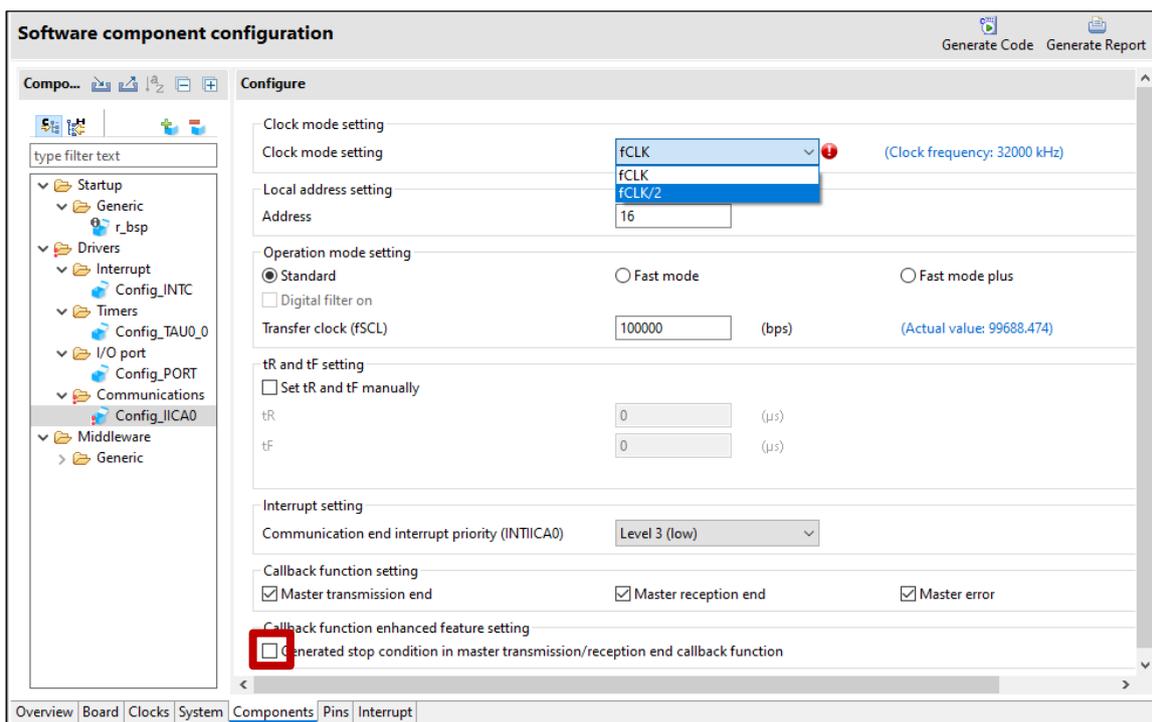
1. RL78/G22 では、IICA として使用可能なリソースが IICA0 のみのため、既存の Config\_IICA1 コンポーネントを削除します。



2. ソフトウェアコンポーネントの選択から、IIC 通信(マスタモード)を選択し、リソースに"IICA0"を指定してください。



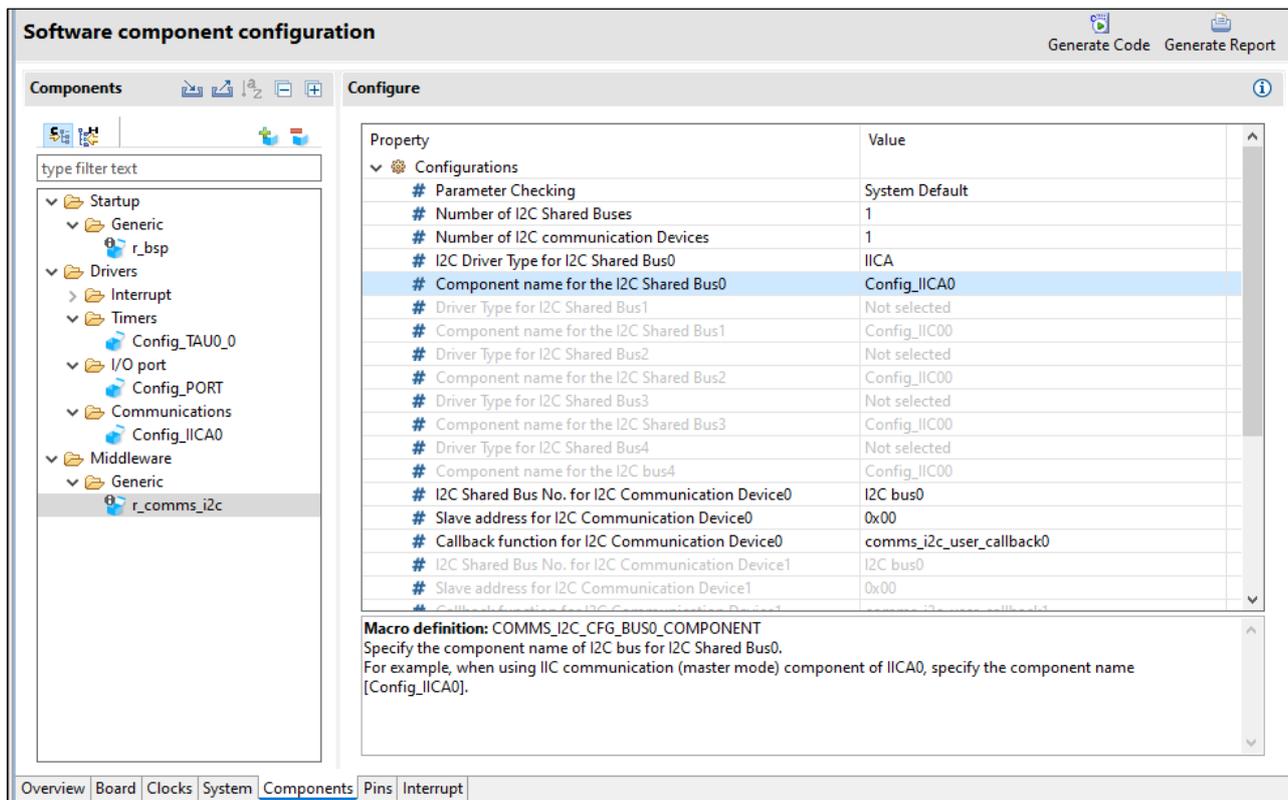
3. Config\_IICA0 で Clock mode setting を”fCLK/2”に変更し、”Generated stop condition in master transmission/reception end callback function”のチェックを外します。



(b) COMMS\_I2C 設定変更

設定が妥当か見直してください。I2C ドライバを変更した場合、見直しが必要です。

r\_comms\_i2c で”Component name for the I2C Shared Bus0”を”Config\_IICA0”に変更します。



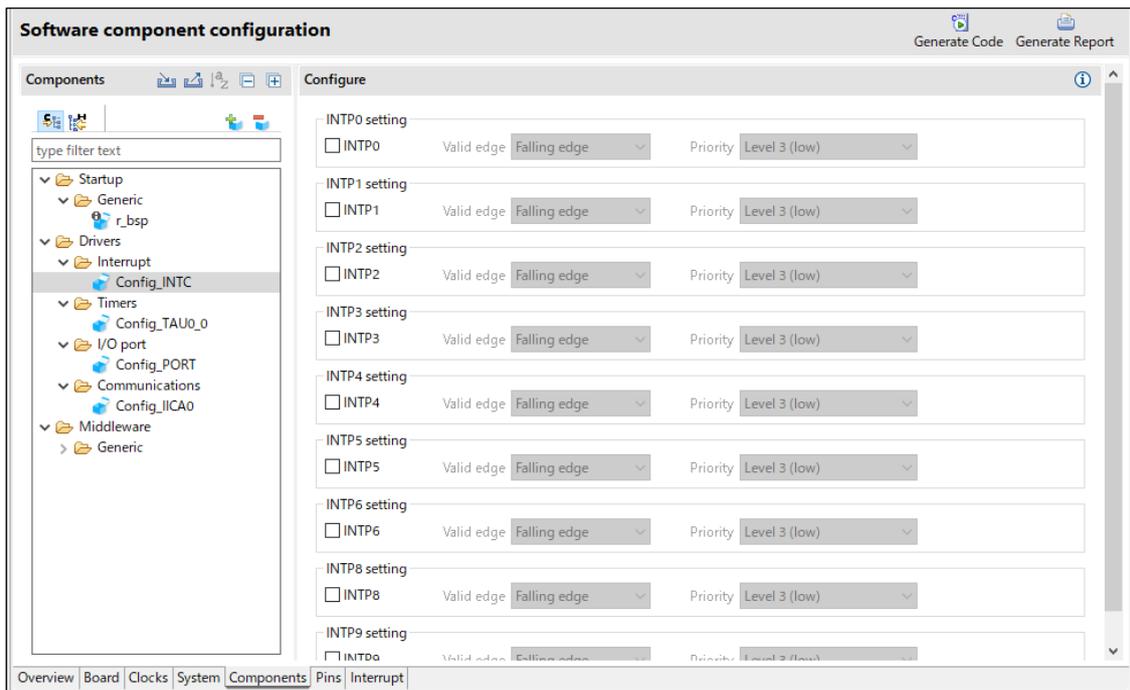
(c) INTC ドライバ設定変更

割り込み信号回路の使用に関して、「6.5 割り込み信号回路に関する注意」を参照してください。

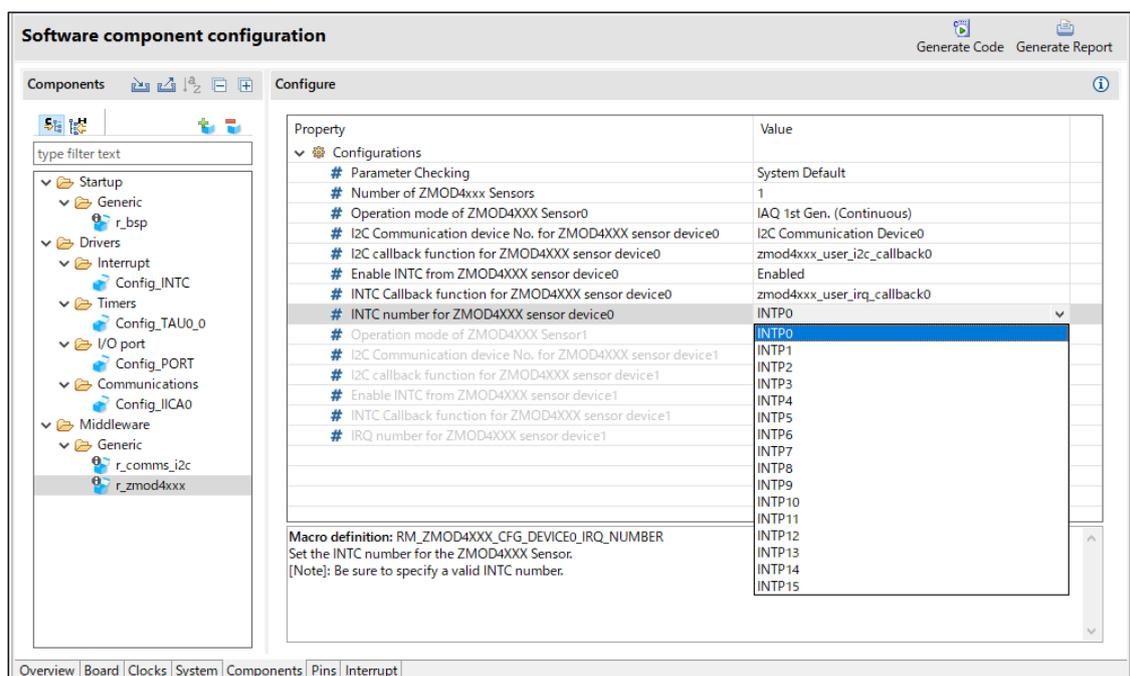
RL78/G22 Fast Prototyping Board の Grove に割り込み端子の割り当てはありません。必要に応じて別端子を使用してください。

割り込みの設定に関しては、センサ制御モジュールで共通の INTC ドライバを使用するため、ZMOD4XXX SIS モジュール(r\_zmod4xxx\_rl)を例に説明します。以下では、ZMOD4XXX を RRH46410 として扱ってください。

1. Config\_INTC で INTP 端子を使用するボードに合わせて設定してください。



2. r\_zmod4xxx\_rl の"INTC number for ZMOD4XXX sensor device0"を上記で設定した INTP 端子に変更します。



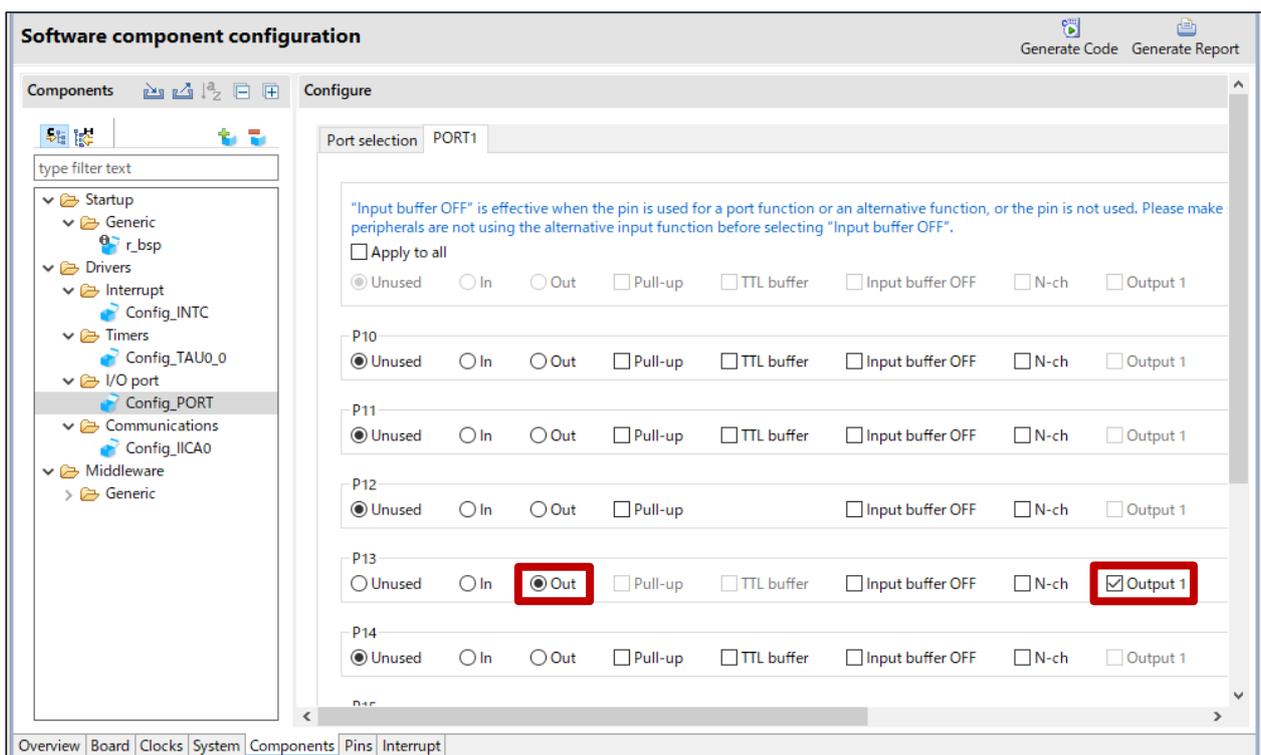
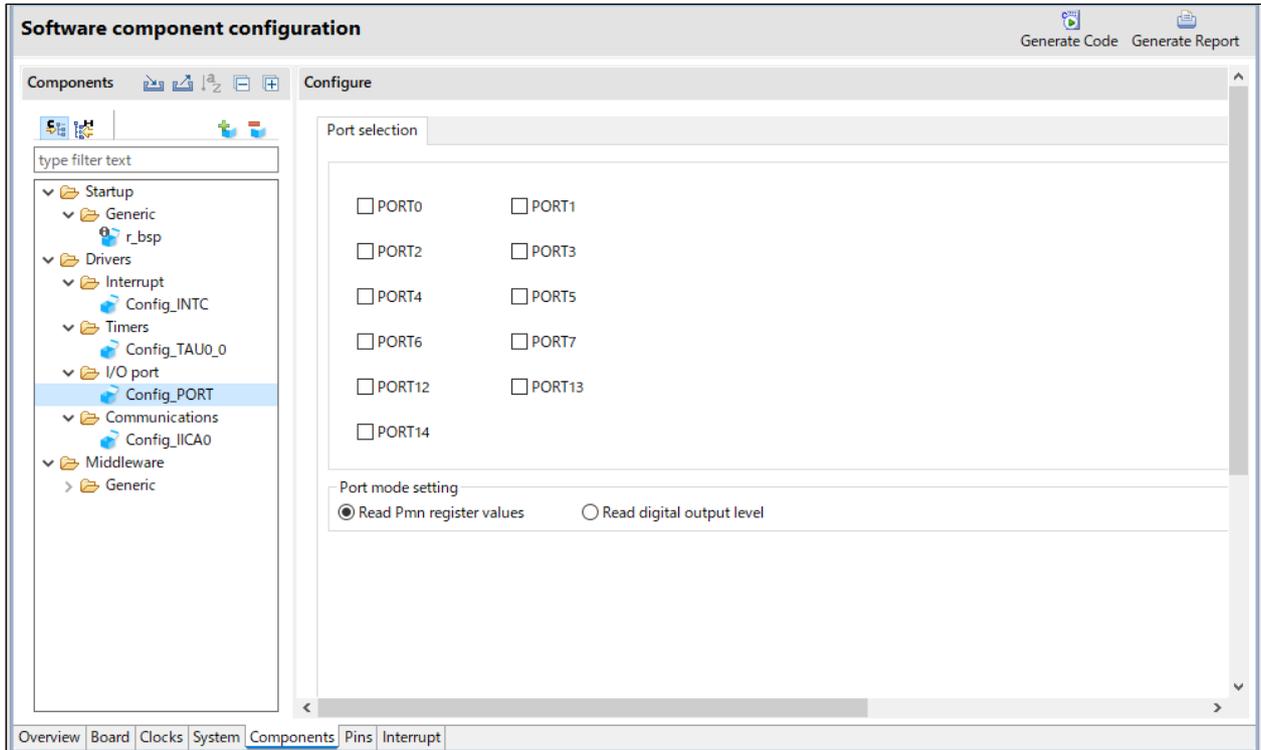
## (d) 汎用 I/O ポートドライバ設定変更 : RESET

RESET 信号制御のソース変更方法については、「6.4.3(1) RESET 端子の設定」を参照してください。

RL78/G22 Fast Prototyping Board の Grove に RESET 端子の割り当てはありません。そのため、任意の Port を割り当ててください。

Config\_PORT で、Pmod の RESET 端子を使用するボードに合わせて I/O ポートに設定します。

Port selection タブと PORT(x)タブで、使用するポートを”Out”に設定し、”Output 1”を有効にします。



(4) Pins

(a) I2C I/F 端子変更

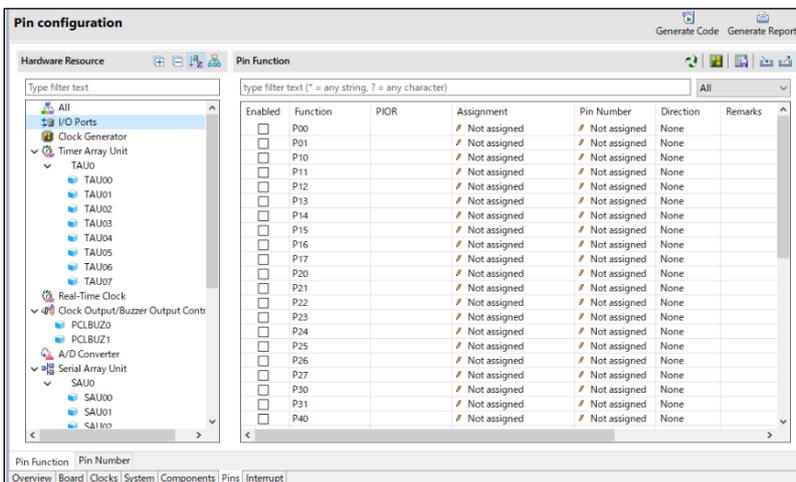
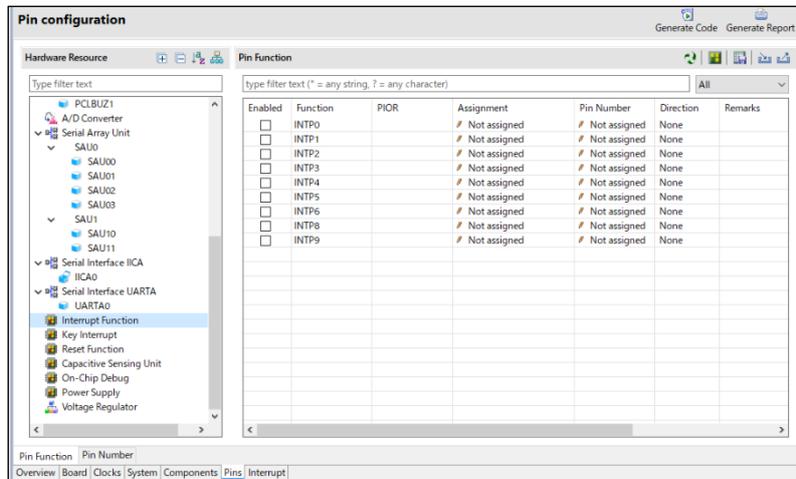
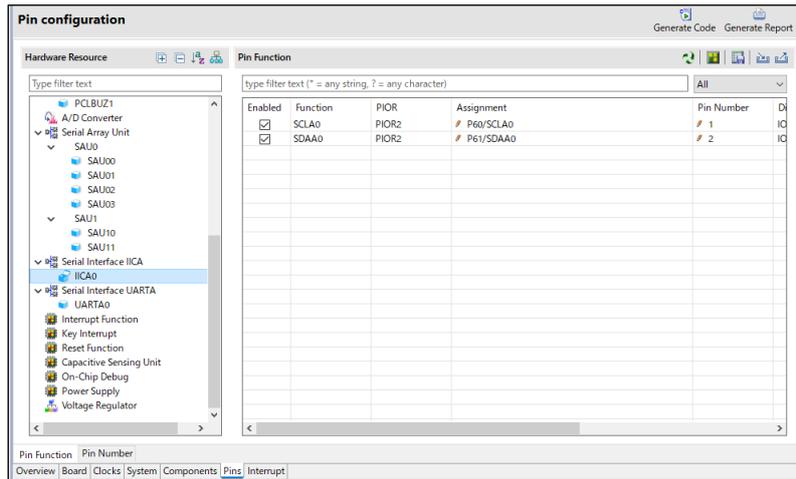
(b) INTP 端子変更

(c) 汎用 I/O ポート端子変更 : RESET

割り込み信号回路の端子設定に関して、「6.5 割り込み信号回路に関する注意」を参照してください。

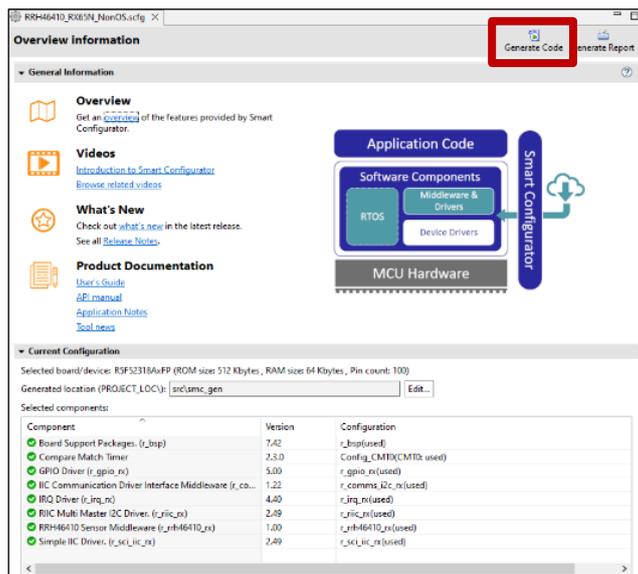
RESET 信号回路の端子設定に関して、「6.6 RESET 信号回路に関する注意」を参照してください。

Pins タブの IICA0、Interrupt Function および I/O Ports を選択し、Pin Function で IICA 端子、INTP 端子および RESET 端子に端子機能が割り当てられていることを確認します。



## (5) コード生成とビルドの実行

[Generate Code]アイコンを押下して、コード生成を行います。



「6.4.2 生成コードの変更」と「6.4.3 サンプルソースの変更」を実施後、プロジェクトをビルドします。

メニューから[Debug Configurations]を選択し、使用するボードに接続するエミュレータに合わせて、Debugger の設定を変更してください。

## 6.4.2 生成コードの変更

Config\_IICA0\_user.c を開き、以下のコードを追加します。

r\_comms\_i2c\_if.h のインクルード定義

```

/*****
Includes
*****/
#include "r_cg_macrodriver.h"
#include "r_cg_userdefine.h"
#include "Config_IICA0.h"
/* Start user code for include. Do not edit comment generated here */
#include "r_comms_i2c_if.h"
/* End user code. Do not edit comment generated here */

```

コールバック関数に rm\_comms\_i2c\_bus0\_callback()関数を追加してください。

送受信完了コールバックは、引数を false に、エラーコールバックは、引数を true に設定してください。

```

/*****
* Function Name: r_Config_IICA0_callback_master_sendend
* Description : This function is a callback function when IICA0 finishes master
transmission.
* Arguments : None
* Return Value : None
*****/
static void r_Config_IICA0_callback_master_sendend(void)
{
/* Start user code for r_Config_IICA0_callback_master_sendend. Do not edit comment
generated here */
rm_comms_i2c_bus0_callback(false);
/* End user code. Do not edit comment generated here */
}

/*****
* Function Name: r_Config_IICA0_callback_master_receiveend
* Description : This function is a callback function when IICA0 finishes master
reception.
* Arguments : None
* Return Value : None
*****/
static void r_Config_IICA0_callback_master_receiveend(void)
{
/* Start user code for r_Config_IICA0_callback_master_receiveend. Do not edit comment
generated here */
rm_comms_i2c_bus0_callback(false);
/* End user code. Do not edit comment generated here */
}

/*****
* Function Name: r_Config_IICA0_callback_master_error
* Description : This function is a callback function when IICA0 master error occurs.
* Arguments : flag -
* status flag
* Return Value : None
*****/
static void r_Config_IICA0_callback_master_error(MD_STATUS flag)
{
/* Start user code for r_Config_IICA0_callback_master_error. Do not edit comment
generated here */
rm_comms_i2c_bus0_callback(true);
/* End user code. Do not edit comment generated here */
}

```

### 6.4.3 サンプルソースの変更

#### (1) RESET 端子の設定

端子の設定については、「6.4.1(4)(c) 汎用 I/O ポート端子変更 : RESET」を参照してください。

RL78/G22 Fast Prototyping Board の Grove に RESET 端子の割り当てはありません。そのため、任意の Port を割り当ててください。

RRH46410\_RL78G23\_NonOS.c を開き、センサリセット制御処理を変更します。

使用するボードに合わせて RESET 端子の指定を修正してください。また、適切なリセット制御論理処理に変更してください。

以下は P13 を RESET 端子として使用する場合の例です。

```
/* Reset RRH46410 sensor (active low). Please change to the IO port
connected to the RES_N pin of the RRH46410 sensor on the customer board. */
P1 = _00_Pn7_OUTPUT_0 | _00_Pn6_OUTPUT_0 | _00_Pn5_OUTPUT_0 |
_00_Pn4_OUTPUT_0 | _08_Pn3_OUTPUT_1 |
_00_Pn2_OUTPUT_0 | _00_Pn1_OUTPUT_0 | _00_Pn0_OUTPUT_0;
R_BSP_SoftwareDelay(10, BSP_DELAY_MILLISECS);
P1 = _00_Pn7_OUTPUT_0 | _00_Pn6_OUTPUT_0 | _00_Pn5_OUTPUT_0 |
_00_Pn4_OUTPUT_0 | _00_Pn3_OUTPUT_0 |
_00_Pn2_OUTPUT_0 | _00_Pn1_OUTPUT_0 | _00_Pn0_OUTPUT_0;
R_BSP_SoftwareDelay(10, BSP_DELAY_MILLISECS);
P1 = _00_Pn7_OUTPUT_0 | _00_Pn6_OUTPUT_0 | _00_Pn5_OUTPUT_0 |
_00_Pn4_OUTPUT_0 | _08_Pn3_OUTPUT_1 |
_00_Pn2_OUTPUT_0 | _00_Pn1_OUTPUT_0 | _00_Pn0_OUTPUT_0;
R_BSP_SoftwareDelay(10, BSP_DELAY_MILLISECS);
```

#### 6.4.4 ツールチェイン設定変更

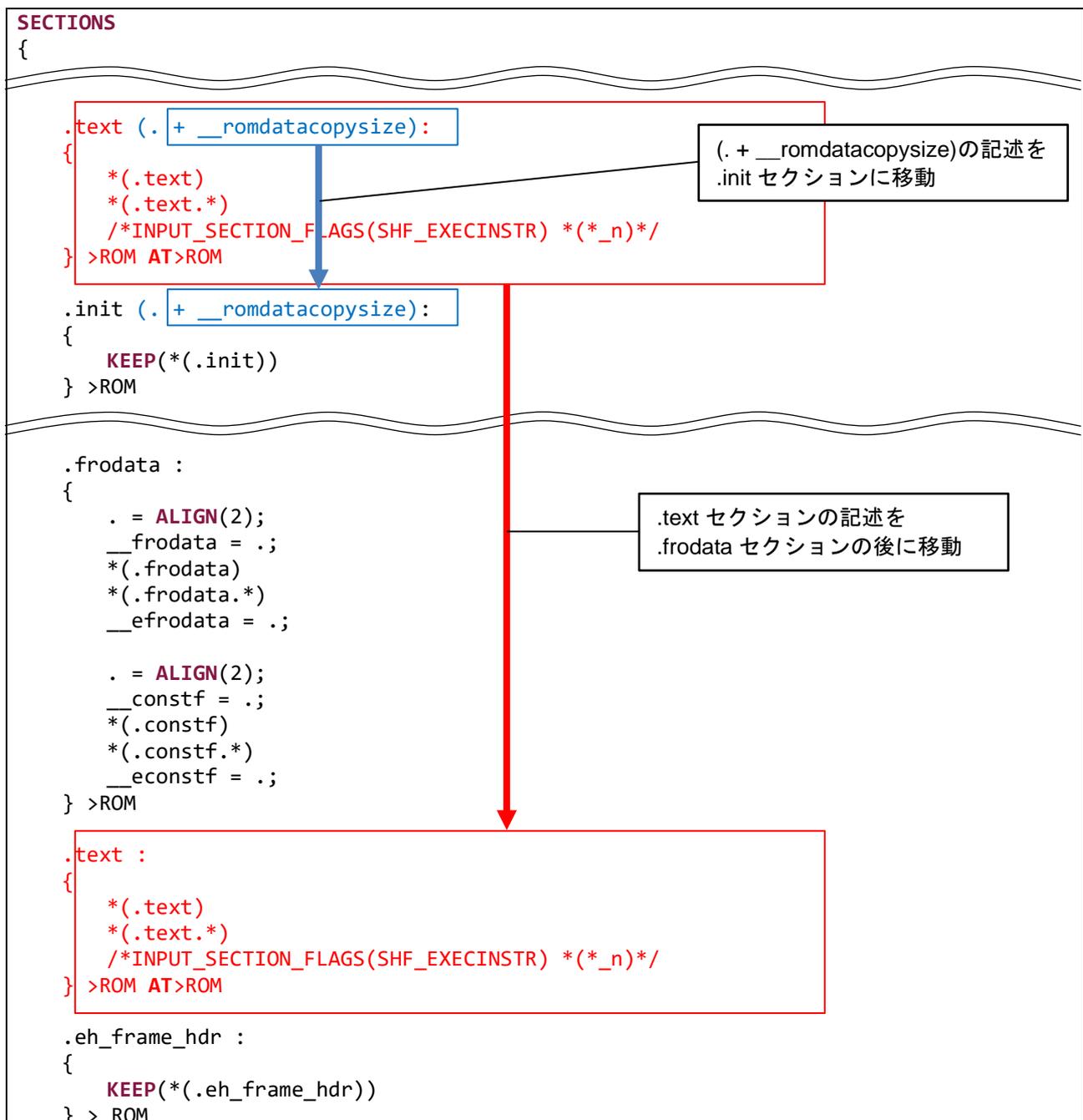
LLVM ツールチェインに変更する場合は、サンプルプロジェクト”RRH46410\_RL78G23\_NonOS\_LLVM”を使用してください。

CC-RL または、LLVM ツールチェイン以外のツールチェインを使用する場合は、本プロジェクトからRRH46410\_RL78G23\_NonOS.c と RL78\_RRH46410.c をコピーしてプロジェクトを作成してください。

また、LLVM ツールチェインを使用した際、セクション配置に起因するビルドエラーが発生する可能性があります。その場合、リンクスクリプトを変更する必要があります。

以下に、例としてサンプルプロジェクト”RRH46410\_RL78G23\_NonOS\_LLVM”で実施している linker\_script.ld ファイルの変更を記載します。

1. .text セクションを.frodata セクション以降に配置しています。



2. .rodata セクションの配置アドレスをミラー領域の先頭アドレスに固定しています。

```

.fini :
{
    KEEP(*(.fini))
} >ROM

PROVIDE(__rodata_limit = CONSTANT(MIRRORAREASTART)+ 0x3000 + LENGTH(MIRROR));

/* The rodata section is placed in MIRROR area in order to access as near
addressing. */
.rodata MAX(., (CONSTANT(MIRRORAREASTART)+ 0x3000)):
.rodata 0x3000 : AT(0x3000)
{
    . = ALIGN(2);
    __rodata = .;
    *(.rodata)
    *(.rodata.*)
    . = ALIGN(2);
}

```

.rodata セクションのアドレスを 0x3000 に変更

3. .data セクションを.oed\_traceram セクション以降のアドレスに配置しています。

```

.eh_frame :
{
    KEEP(*(.eh_frame))
} > ROM

.oed_traceram 0xf4300 (NOLOAD) : AT(0xf4300)
{
    KEEP(*(.oed_traceram))
} >RAM

.data 0xF4700 : AT(__mdata)
{
    . = ALIGN(2);
    PROVIDE (__datastart = .);
}

.bssf (NOLOAD):
{
    PROVIDE(__bssfstart = .);
    . = ALIGN(2);
    *(.bssf)
    *(.bssf.*)
    /*INPUT_SECTION_FLAGS(!SHF_EXECINSTR, SHF_WRITE, SHF_ALLOC) *(*_f)*/
    . = ALIGN(128);
    __end = .;
} >RAM AT>RAM
PROVIDE(__bssfsize = SIZEOF(.bssf));

PROVIDE(__stack_size = 0x100);
.oed_traceram 0xf4300 (NOLOAD) : AT(0xf4300)
{
    KEEP(*(.oed_traceram))
} >RAM
.stack 0xFFE20 (NOLOAD) : AT(0xFFE20)
{
    PROVIDE(__stack = .);
}

```

.data セクションのアドレスを 0xF4700 に変更

.oed\_traceram セクションの記述を .data セクションの前に移動

注 セクションの配置アドレスの変更によって、使用可能な ROM / RAM 領域が減少します。

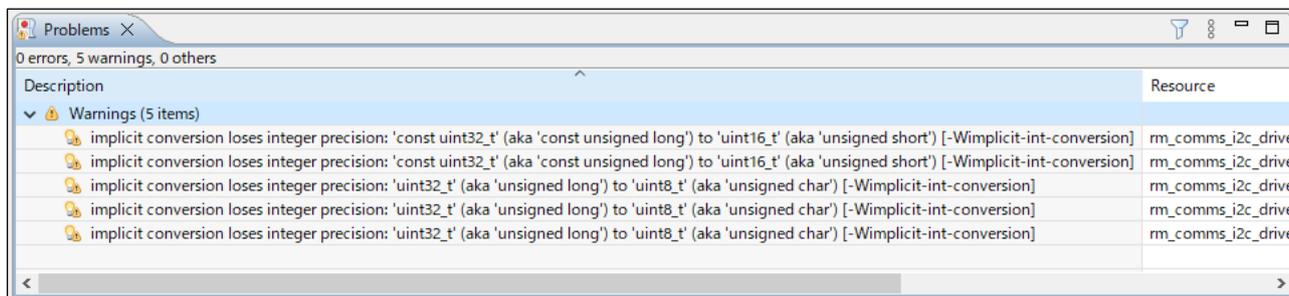
### 6.4.5 LLVM ビルド時の注意

LLVM プロジェクトをビルド時、以下の Warning が発生します。

これらの Warning は、COMMS\_I2C で使用される”slave\_address”と”bytes”を 32-bit 型で扱っているため発生しています。

”slave\_address”は 7-bit データ、”bytes”は 16-bit データのため、変換によるロスが発生しません。

そのため Warning が発生していますが、動作に問題はありません。



## 6.5 割り込み信号回路に関する注意

組み込み対象回路に合わせて Configuration 設定する必要があります。

センサの割り込み要求信号(SensorINT)を使用する場合、トリガレベルを正しく設定してください。

### (1) RRH46410 Sensor Pmod Board の回路構成

RRH46410 Sensor Pmod Board の割り込み信号の回路構成を以下に示します。詳細はボードのデータシートを参照してください。

表 6-2 RRH46410 Sensor Pmod Board の割り込み信号の回路構成

Sensor Pmod Board	回路構成（プルアップ抵抗有無設定可）
#1: IRQ# L 出力時：データ利用可能	SensorINT の反転出力（Open-Drain 出力）、かつ IRQ#に固定接続（SensorINT 反転信号を出力禁止にできない）
#7: BUSY#	－（SensorINT 信号の出力回路無し）

### (2) 複数の I2C デバイスをデジ チェーン接続させた場合の割り込みトリガ制御方法

各 I2C デバイスの IRQ#が Open-Drain 出力の場合、デジ チェーン接続が可能です。（注）

この接続では、割り込み要求信号が発生した場合、割り込み要求を出力した I2C デバイスの特定が困難です。

**回路構成上、複数の割り込み出力信号が Pmod #1 の IRQ#に接続される場合、割り込みを使用せず、全ての I2C デバイスを制御してください。（注）**

注：Open-Drain 以外の出力（CMOS や TTL の Push-Pull 出力）の I2C デバイスを接続しないでください。接続した場合、割り込み信号が繋がる I2C デバイスや周辺回路 IC が壊れる可能性があります。

### (3) 割り込み要求出力先変更が可能な Sensor Pmod Board を利用する場合の割り込みトリガ制御方法

Sensor Pmod Board によっては、ボード上の回路を変更することで、割り込み要求信号を Pmod #7 の BUSY#に出力可能なボードがあります。

この割り込み要求信号を BUSY#出力に切り替えたボードと標準の IRQ#出力ボードをデジ チェーン接続させることで、同時に 2 デバイスの割り込みトリガ動作が可能になります。

なお、MCU 側の Pmod #1 と #7 が割り込み信号入力機能を持つことも動作条件になります。

表 6-3 Sensor Pmod Board の割り込み信号の回路構成（ボード依存）

Sensor Pmod Board	回路構成（プルアップ抵抗有無設定可）
#1: IRQ#	以下の回路構成が可能（ボード依存）
#7: BUSY#	

- #1 もしくは#7 のどちらかに SensorINT の非反転出力、反転出力（Open-Drain 出力もしくは Push-Pull 出力）が可能（注 1）
- #1 もしくは#7 のどちらも SensorINT 出力無効かつオープン状態

注 1：非反転出力もしくは反転出力、Open-Drain 出力もしくは Push-Pull 出力は、ボード依存です。

## 6.6 RESET 信号回路に関する注意

組み込み対象回路に合わせて設定する必要があります。

### (1) RRH46410 Sensor Pmod Board の回路構成

RRH46410 Sensor Pmod Board のボードの RESET 信号の回路構成を以下に示します。詳細はボードのデータシートを参照してください。

表 6-4 RRH46410 Sensor Pmod Board の RESET 信号の回路構成

Sensor Pmod Board	回路構成（プルアップ抵抗有無設定可）
#2: RESET# L 入力時：デバイス Reset	センサデバイスの RESET 信号に直接接続

### (2) 複数の I2C デバイスをデジ チェーン接続させた場合の RESET 出力制御方法

MCU の 1 つの RESET 出力信号で、複数 I2C デバイスのリセット制御を行うことができます。その場合、複数 I2C デバイスのリセット制御論理が合う RESET 信号回路の構成と複数 I2C デバイスの要求を満たすリセットシーケンスの実現が必要です。

また、複数 I2C デバイスを接続する場合、端子負荷容量が大きくなるため、十分なりセットパルス期間を発生させてください。

## 6.7 Renesas Pmod Sensor Board のデジ・チェーン接続時のプルアップ抵抗処理

デジ・チェーン接続時のプルアップ抵抗の推奨処理を以下に示します。また、他ボードのプルアップ処理を無効にしてください。

多数の Sensor Board のプルアップ抵抗を同時に有効にした場合、正常に動作しない場合があります。

表 6-5 デジ・チェーン接続時のプルアップ抵抗を有効化する対象ボード

Pmod Sensor Board Type 6A 信号名	プルアップ抵抗の推奨処理
#1: IRQ# （注 1）	プルアップ抵抗回路付ボードで、最も MCU ボードに近いボードのみ有効
#2: RESET# （注 1）	プルアップ抵抗回路付ボードで、最も MCU ボードに近いボードのみ有効
#3: SCL	最も MCU ボードに近いボードのみ有効
#4: SDA	最も MCU ボードに近いボードのみ有効
#7: BUSY# （注 1, 2）	プルアップ抵抗回路付ボードで、最も MCU ボードに近いボードのみ有効

注 1：プルアップ抵抗回路無しボードがあります。

注 2：割り込み要求信号として利用時に処理してください。





## 改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
Rev 1.00	Dec.19.24	-	初版発行

## 製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本ドキュメントおよびテクニカルアップデートを参照してください。

### 1. 静電気対策

CMOS 製品の取り扱いの際は静電気防止を心がけてください。CMOS 製品は強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、当社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、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、 $V_{IL}(\text{Max.})$  から  $V_{IH}(\text{Min.})$  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、 $V_{IL}(\text{Max.})$  から  $V_{IH}(\text{Min.})$  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

- 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。
  - 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。
  - 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。
  - 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。
  - 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。
  - 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。  
標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット等  
高品質水準： 輸送機器（自動車、電車、船舶等）、交通制御（信号）、大規模通信機器、金融端末基幹システム、各種安全制御装置等  
当社製品は、データシート等により高信頼性、Harsh environment 向け製品と定義しているものを除き、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じても、当社は一切その責任を負いません。
  - 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。
  - 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。
  - 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。
  - 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。
  - 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。
  - 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。
  - 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。
  - 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](http://www.renesas.com)

## 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。

## 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/)