

## RZ/N2H グループ

### Encoder I/F FA-CODER sample program

#### 要旨

本書は、RZ/N2H Encoder I/F FA-CODER®のサンプルプログラム・パッケージについて記載します。

FA-CODER® 通信プロトコル仕様やエンコーダの仕様については、多摩川精機株式会社にお問い合わせの上入手してください。

#### 動作確認デバイス

RZ/N2H Evaluation Board (RTK9RZN2Hxxxxxxxx)

#### 目次

1. パッケージ内容 .....	2
1.1 ソフトウェア .....	2
1.2 ドキュメント .....	2
2. ファイル構成 .....	3
3. FA-CODER サンプルプログラムについて .....	5
3.1 ソフトウェア情報 .....	5
3.1.1 ベース OS .....	5
3.1.2 メモリサイズ .....	5
3.2 ハードウェア情報 .....	6
3.2.1 デバイス .....	6
3.2.2 ターゲットボード .....	6
3.3 各開発環境における動作手順 :CR52 版 .....	8
3.3.1 サンプルプログラム実行前の準備 .....	8
3.3.2 EWARM: IAR システムズ社製 .....	8
3.3.3 e <sup>2</sup> studio: RENESAS 社製 .....	10
3.4 各開発環境における動作手順 :CA55 版 .....	12
3.4.1 サンプルプログラム実行前の準備 .....	12
3.4.2 EWARM: IAR システムズ社製 .....	12
3.4.3 e <sup>2</sup> studio: RENESAS 社製 .....	14
改訂記録 .....	18

FA-CODER は多摩川精機株式会社の登録商標です。

## 1. パッケージ内容

本パッケージには次のものが含まれています。

RZ/N2H の FA-CODER I/F は 11 軸までに対応していますが、サンプルプログラムは、1 軸にのみ対応しています。2 軸以上を同時に使用する場合は、サンプルプログラムを必要な軸の数に合わせて変更してください。

### 1.1 ソフトウェア

- ソースコード

No.	名称	版数
1	RZ/N2H FA-CODER サンプルプログラム (CR52 版 <sup>注</sup> )	2.0
2	RZ/N2H FA-CODER サンプルプログラム (CA55 版 <sup>注</sup> )	2.0

【注】 サンプルプログラムには、CPU コア Cortex-R52 で動作する CR52 版と、CPU コア Cortex-A55 で動作する CA55 版があります。CR52 版、CA55 版はそれぞれの版に関する説明です。

### 1.2 ドキュメント

No.	文書名	版数	ファイル名
1	RZ/N2H グループ Encoder I/F FA-CODER sample program リリースノート	2.00	(日) r11an0928jj0200-rzn2h.pdf (本書) (英) r11an0928ej0200-rzn2h.pdf
2	RZ/N2H グループ FA-CODER サンプルプログラム アプリケーションノート	2.00	(日) r11an0927jj0200-rzn2h-fa-coder.pdf (英) r11an0927ej0200-rzn2h-fa-coder.pdf

## 2. ファイル構成

本パッケージのファイル構成と内容物の詳細を以下に示します。

Top

```

├─ r11an0928jj0200-rzn2h.pdf
├─ r11an0928ej0200-rzn2h.pdf
├─ workspace
│   └─ Software
│       ├── iccarm
│       │   ├── RZ_N2H_CR52_fac.zip      : RZ/N2H FA-CODER サンプルプログラム一式
│       │   │   └─ CR52 版 (IAR)
│       │   └─ RZ_N2H_CA55_fac.zip
│       │       ├── RZ_N2H_CA55_0_fac    : RZ/N2H FA-CODER サンプルプログラム一式
│       │       │   └─ CA55 版 (IAR)
│       │       └─ RZ_N2H_CR52_0_primary
│       │           └─ CPU_CTRL レジスタ設定プログラム (IAR)
│       └─ gcc
│           ├── RZ_N2H_CR52_fac.zip      : RZ/N2H FA-CODER サンプルプログラム一式
│           │   └─ CR52 版 (e2 studio)
│           └─ RZ_N2H_CA55_fac.zip
│               ├── RZ_N2H_CA55_0_fac    : RZ/N2H FA-CODER サンプルプログラム一式
│               │   └─ CA55 版 (e2 studio)
│               └─ RZ_N2H_CR52_0_primary
│                   └─ CPU_CTRL レジスタ設定プログラム (e2 studio)
└─ Documents
    ├── r11an0927jj0200-rzn2h-fa-coder.pdf
    └─ r11an0927ej0200-rzn2h-fa-coder.pdf

```

RZ\_N2H\_CR52\_fac.zip および RZ\_N2H\_CA55\_0\_fac フォルダのファイル構成を以下に示します。

Top folder

```

├─ configuration.xml      : FSP Configuration data
├─ (ビルドツール依存 環境ファイル)
├─ src
│   ├── hal_entry.c      : FA-CODER サンプルプログラム
│   ├── fac_main.c       : FA-CODER サンプルプログラム
│   ├── siochar.c        : SCI_UART サンプルプログラム
│   ├── siorw.c          : SCI_UART サンプルプログラム
│   ├── sio_char.h       : SCI_UART サンプルプログラム
│   └─ drv
│       └─ fac
│           ├── iodefinc_fac.h    : FAC レジスタ定義ファイル
│           ├── r_fac_rzt2.c      : FAC ドライバファイル
│           ├── r_fac_rzt2_config.h : FAC ドライバファイル
│           ├── r_fac_rzt2_dat.h  : FAC ドライバファイル
│           ├── r_fac_rzt2_if.h   : FAC ドライバファイル
│           ├── r_scie.c          : FAC ドライバファイル
│           └─ r_scie.h           : FAC ドライバファイル

```

RZ\_N2H\_CR52\_0\_primary フォルダのファイル構成を以下に示します。

Top folder

- |— configuration.xml : FSP Configuration data
- |— ( ビルドツール依存 環境ファイル )
- └— src
  - └— hal\_entry.c : CA55 起動用プログラム

### 3. FA-CODER サンプルプログラムについて

FA-CODER サンプルプログラム一式を使用するために必要な情報を記載します。

#### 3.1 ソフトウェア情報

##### 3.1.1 ベース OS

本サンプルプログラムは OS に依存しません。

##### 3.1.2 メモリサイズ

本サンプルプログラムおよび、FA-CODER ドライバが使用するメモリサイズは次表に示した通りです。表には、FA-CODER のエンコーダ I/F として使われている DMAC ドライバの分を除いて、Flexible Software Package や、コンパイラの C 言語ライブラリで使われるメモリサイズは含まれません。

##### (1) CR52 版

領域名		開発環境別メモリサイズ	
		EWARM [kBytes]	e <sup>2</sup> studio [kBytes]
FA-CODER ドライバ(CR52 版)	コード領域	6.9	6.3
	データ領域 (初期値あり)	0.0	0.0
	データ領域 (初期値なし)	2.7	2.7
	定数領域	0.7	0.7
エンコーダ I/F 用 DMAC ドライバ 注(CR52 版)	コード領域	2.5	2.1
	データ領域 (初期値あり)	0.0	0.0
	データ領域 (初期値なし)	0.2	0.2
	定数領域	0.0	0.0
サンプルプログラム(CR52 版)	コード領域	4.3	4.9
	データ領域 (初期値あり)	0.0	0.0
	データ領域 (初期値なし)	0.6	0.6
	定数領域	2.2	2.2

【注】 FA-CODER のエンコーダ I/F で使われている DMAC ドライバが使用するメモリサイズです。DMAC ドライバを使用する機能は、CR52 版でのみ有効になっています。

##### (2) CA55 版

領域名		開発環境別メモリサイズ	
		EWARM [kBytes]	e <sup>2</sup> studio [kBytes]
FA-CODER ドライバ(CA55 版)	コード領域	10.4	8.8
	データ領域 (初期値あり)	0.0	0.0
	データ領域 (初期値なし)	3.3	3.4
	定数領域	1.1	1.4
サンプルプログラム(CA55 版)	コード領域	6.9	8.8
	データ領域 (初期値あり)	0.1	0.0
	データ領域 (初期値なし)	0.6	0.6
	定数領域	2.4	2.5

## 3.2 ハードウェア情報

### 3.2.1 デバイス

RZ/N2H

### 3.2.2 ターゲットボード

#### (1) ボード名

RZ/N2H Evaluation Board (RTK9RZN2Hxxxxxxxx)

#### (2) ターゲットボードの設定

ターゲットボードの設定は以下の通りです。

DSW2-6: OFF

DSW5-1: OFF, DSW5-2: OFF, DSW5-3: OFF, DSW5-7: OFF, DSW5-8: OFF

DSW7-3: OFF, DSW7-4: ON

DSW12\_5: ON, DSW12\_6: OFF, DSW12\_7: ON, DSW12\_8: OFF

DSW13-1: OFF, DSW13-2: ON, DSW13-3: OFF, DSW13-4: ON

DSW18-5: OFF, DSW18-6: ON

DSW20-1: OFF, DSW20-3: OFF, DSW20-5: OFF

DSW21-2: OFF, DSW21-3: ON, DSW21-4: OFF, DSW21-5: ON, DSW21-6: OFF

DSW3-1: ON, DSW3-2: OFF, DSW3-3: ON, DSW3-6: OFF (xSPI1 boot mode に設定)

JP8: 2-3pin 間をショート (VDD1833\_2 を 3.3V に設定)

JP9: 2-3pin 間をショート (VDD1833\_3 を 3.3V に設定)

JP23: 1-2pin 間をショート, 3-4pin 間と 5-6pin 間はオープン (VDD1833\_6 を 3.3V に設定)

#### (3) ターゲットボードの使用端子

エンコーダ I/F として使用する端子と、ターゲットボードのピンヘッダ端子との対応は以下の通りです。チャンネル FACODER8 は利用できません。

チャンネル	端子名	ピンヘッダ 端子	入出力	I/O 電源ドメイン	内容
FACODER0	RXDE00	CN44 #9	入力	VDD33	データ受信端子
	TXDE00	CN44 #7	出力	VDD33	リクエスト出力端子
	DEE00	CN44 #5	出力	VDD33	ドライブ/レシーブ制御端子
FACODER1	RXDE01	CN44 #8	入力	VDD33	データ受信端子
	TXDE01	CN44 #6	出力	VDD33	リクエスト出力端子
	DEE01	CN44 #4	出力	VDD1833_5	ドライブ/レシーブ制御端子
FACODER2	RXDE02	CN43 #23	入力	VDD33	データ受信端子
	TXDE02	CN43 #21	出力	VDD33	リクエスト出力端子
	DEE02	CN43 #27	出力	VDD33	ドライブ/レシーブ制御端子
FACODER3	RXDE03	CN43 #24	入力	VDD1833_6	データ受信端子
	TXDE03	CN43 #22	出力	VDD1833_6	リクエスト出力端子
	DEE03	CN43 #18	出力	VDD1833_6	ドライブ/レシーブ制御端子
FACODER4	RXDE04	CN44 #27	入力	VDD33	データ受信端子
	TXDE04	CN44 #25	出力	VDD33	リクエスト出力端子
	DEE04	CN44 #23	出力	VDD33	ドライブ/レシーブ制御端子

チャンネル	端子名	ピンヘッダ 端子	入出力	I/O 電源ドメイン	内容
FACODER5	RXDE05	CN43 #14	入力	VDD1833_6	データ受信端子
	TXDE05	CN43 #12	出力	VDD1833_6	リクエスト出力端子
	DEE05	CN43 #10	出力	VDD1833_6	ドライブ/レシーブ制御端子
FACODER6	RXDE06	CN42 #18	入力	VDD1833_3	データ受信端子
	TXDE06	CN42 #16	出力	VDD1833_3	リクエスト出力端子
	DEE06	CN42 #14	出力	VDD1833_3	ドライブ/レシーブ制御端子
FACODER7	RXDE07	CN42 #36	入力	VDD1833_3	データ受信端子
	TXDE07	CN42 #34	出力	VDD1833_3	リクエスト出力端子
	DEE07	CN42 #24	出力	VDD1833_3	ドライブ/レシーブ制御端子
FACODER8	RXDE08	-	-	-	N.A.
	TXDE08	-	-	-	N.A.
	DEE08	-	-	-	N.A.
FACODER9	RXDE09	CN51 #18	入力	VDD1833_2	データ受信端子
	TXDE09	CN51 #16	出力	VDD1833_2	リクエスト出力端子
	DEE09	CN51 #14	出力	VDD1833_2	ドライブ/レシーブ制御端子
FACODER10	RXDE10	CN51 #27	入力	VDD1833_2	データ受信端子
	TXDE10	CN51 #25	出力	VDD1833_2	リクエスト出力端子
	DEE10	CN51 #23	出力	VDD1833_2	ドライブ/レシーブ制御端子
FACODER11	RXDE11	CN51 #26	入力	VDD1833_2	データ受信端子
	TXDE11	CN51 #24	出力	VDD1833_2	リクエスト出力端子
	DEE11	CN51 #22	出力	VDD1833_2	ドライブ/レシーブ制御端子

### 3.3 各開発環境における動作手順 :CR52 版

#### 3.3.1 サンプルプログラム実行前の準備

本サンプルプログラムでは、PC との通信を行います。ターゲットボードの USB 接続端子は、CN27 です。ホスト PC と接続したときに現れる COM ポートのうち、大きい番号のついたポートを選択します。ホスト PC のターミナルソフトは、次表のように設定します。

機能	設定
通信方式	調歩同期式 シリアル送信 / 受信
送信 / 受信順	LSB ファースト
転送速度	19200 bps
キャラクタ長	8 ビット
ストップビット長	1 ビット
パリティ機能	なし
ハードウェアフロー制御	なし

#### 3.3.2 EWARM: IAR システムズ社製

##### (1) ビルド環境

IAR Embedded Workbench for Arm (EWARM)

Version 9.60.2 + patch (EWARM\_Patch\_for\_RZT2H\_N2H\_rev1.0)

RENESAS FSP Smart Configurator (FSP SC) 2024-10

RENESAS Flexible Software Package (FSP) for RZ/N2 v2.1.0

##### (2) 実行環境 ICE

IAR I-jet

##### (3) サンプルプログラムのビルド手順

サンプルプログラムのビルド手順は以下の通り。

- 1 RZ\_N2H\_CR52\_fac.zip を展開し、展開したソースファイルを任意の場所にコピーする。
- 2 EWARM を起動する。
- 3 [ファイル]メニュー→[ワークスペースを開く]を選択する。
- 4 展開したソースファイルの RZ\_N2H\_fac.eww を開く。
- 5 EWARM IDE の[ツール]メニューから FSP Smart Configurator を起動する。 注

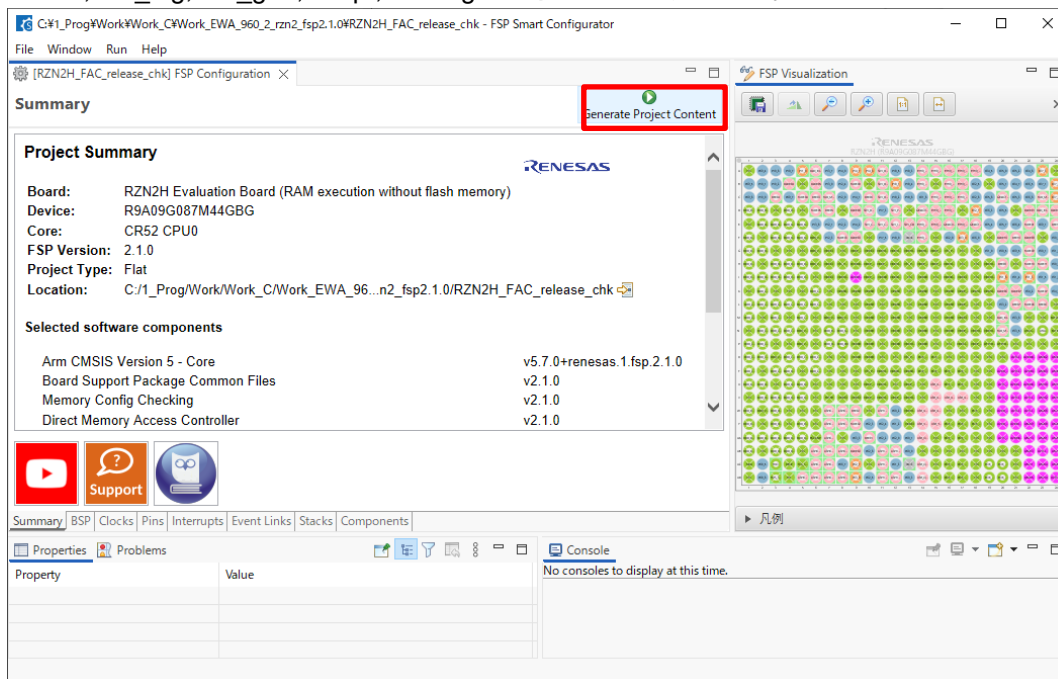
**【注】** 次の手順で、EWARM IDE の[ツール]メニューに FSP Smart Configurator の起動が加えられます。EWARM IDE の[ツール]メニュー→[ツールの設定]を選択する。  
[新規作成]ボタンを選択し、各フィールドに表の文字列を指定して、[OK]を押す。

フィールド	文字列
メニューテキスト	FSP Smart Configurator
コマンド	\$RASC_EXE_PATH\$
引数	--compiler IAR configuration.xml
初期ディレクトリ	\$PROJ_DIR\$

コマンド欄の文字列は、Smart Configurator 実行ファイル rasc.exe のパスを保持する変数です。  
なお、コマンドプロンプトから FSP Smart Configurator のインストール先フォルダを直接指定して起動することもできます。



- 6 Smart Configurator の FSP Configuration ペインで、Generate Project Content をクリックする。  
rzn, rzn\_cfg, rzn\_gen, script, .settings フォルダが生成される。



- 7 プロジェクト生成が完了したら、Smart Configurator を閉じる。  
8 EWARM の[プロジェクト]メニュー→[すべてを再ビルド]を選択する。  
ファイル Debug¥Exe¥RZ\_N2H\_fac.out が生成される。

#### (4) サンプルプログラムの実行手順

「ビルド手順」を実行後、ターゲットボード、デバッガを正しく接続し、以下の操作を行う。

- 1 [プロジェクト]メニュー→[ダウンロードしてデバッグ]を選択する。
- 2 [デバッグ]メニュー→[実行]を選択する。

#### (5) サンプルプログラムの実行結果

サンプルプログラムを実行し、ターミナルソフトのウィンドウにコマンドを入力してください。コマンドについては、RZ/N2H グループ FA-CODER サンプルプログラム アプリケーションノートの 4.11.7 コンソールコマンドを参照してください。

```

COM8 - Tera Term VT
File Edit Setup Control Window Help
Tama sample program start
R_FAC_GetVersion = 2.0

tamagawa>single
single command
result:success
single turn data:      4576105
request id:           0H
parity bit(request id): 0H
information code:     0H
encoder alarm:        2H
communication alarm:  0H
crc data:             DDH
tamagawa>

```

### 3.3.3 e<sup>2</sup> studio: RENESAS 社製

#### (1) ビルド環境

RENESAS e<sup>2</sup> studio 2024-10

Toolchain version: GNU ARM Embedded 12.2.1.arm-12-24

RENESAS Flexible Software Package (FSP) for RZ/N2 v2.1.0

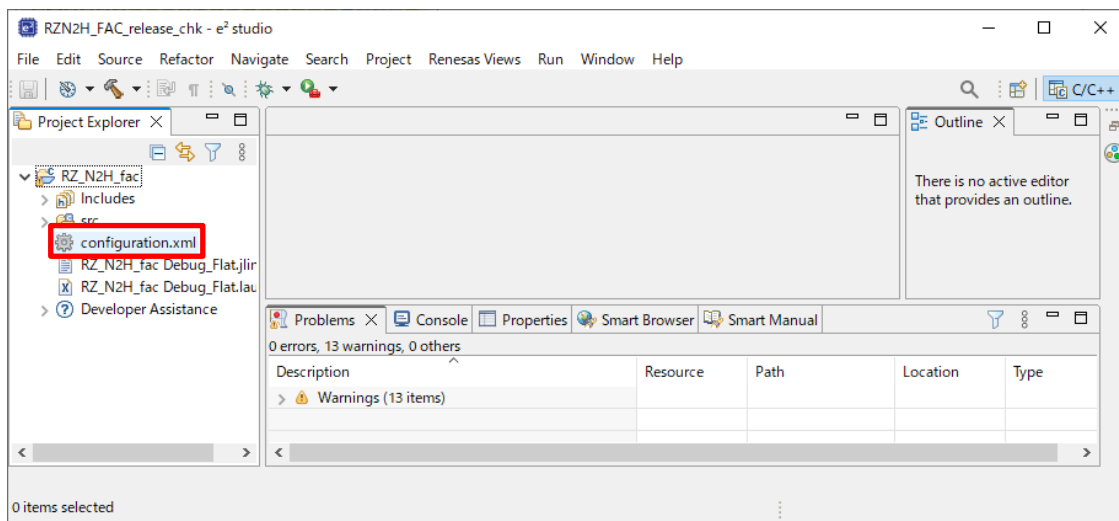
#### (2) 実行環境 ICE

SEGGER J-Link v7.98c

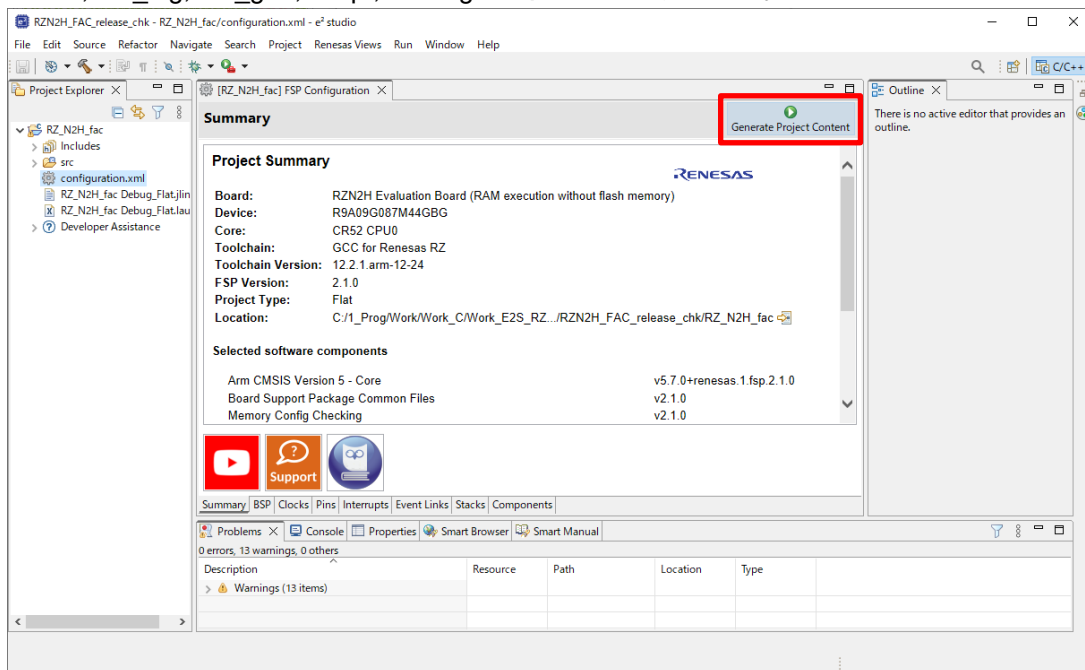
#### (3) サンプルプログラムのビルド手順

サンプルプログラムのビルド手順は以下の通り。

- 1 RZ\_N2H\_CR52\_fac.zip を展開し、展開したソースファイルを任意の場所にコピーする。
- 2 e<sup>2</sup> studio を起動し、ワークスペースへ移動後 [ファイル]メニュー→[インポート]をクリックし、一般 > 既存プロジェクトをワークスペースへ を選択して[次へ]をクリックする。
- 3 プロジェクトのインポート画面でルートディレクトリとしてサンプルプログラムを展開したフォルダを選択する。
- 4 プロジェクトを選択し、プロジェクトをワークスペースにコピー をチェックして [終了]をクリックする。
- 5 e<sup>2</sup> studio の Project Explorer ペインで configuration.xml をダブルクリックして開く。



- 6 e<sup>2</sup> studio の FSP Configuration ペインで Generate Project Content をクリックする。  
rzn, rzn\_cfg, rzn\_gen, script, .settings フォルダが生成される。



- 7 [プロジェクト]メニュー→[すべてビルド]を選択する。  
ファイル Debug¥RZ\_N2H\_fac.elf が生成される。

#### (4) サンプルプログラムの実行手順

「ビルド手順」を実行後、ターゲットボード、デバッガを正しく接続し、以下の操作を行う。

- 1 [実行]メニュー→[デバッグ]→[Renesas GDB Hardware Debugging]を選択する。
- 2 [デバッグ]をクリックすると内蔵 RAM へのダウンロードを開始する。
- 3 [実行]メニュー→[再開]をクリックすると、サンプルプログラムが実行される。

#### (5) サンプルプログラムの実行結果

サンプルプログラムを実行し、ターミナルソフトのウィンドウにコマンドを入力してください。コマンドについては、RZ/N2H グループ FA-CODER サンプルプログラム アプリケーションノートの 4.11.7 コンソールコマンドを参照してください。

```

COM8 - Tera Term VT
File Edit Setup Control Window Help
tama sample program start
R_FAC_GetVersion = 2.0

tamagawa>single
single command
result:success
single turn data:      4576105
request id:           0H
parity bit(request id): 0H
information code:     0H
encoder alarm:        2H
communication alarm:  0H
crc data:             0DH
tamagawa>

```

### 3.4 各開発環境における動作手順 :CA55 版

#### 3.4.1 サンプルプログラム実行前の準備

本サンプルプログラムでは、PC との通信を行います。ターゲットボードの USB 接続端子は、CN27 です。ホスト PC と接続したときに現れる COM ポートのうち、小さい番号のついたポートを選択します。ホスト PC のターミナルソフトは、次表のように設定します。

機能	設定
通信方式	調歩同期式 シリアル送信 / 受信
送信 / 受信順	LSB ファースト
転送速度	19200 bps
キャラクタ長	8 ビット
ストップビット長	1 ビット
パリティ機能	なし
ハードウェアフロー制御	なし

#### 3.4.2 EWARM: IAR システムズ社製

##### (1) ビルド環境

IAR Embedded Workbench for Arm (EWARM)

Version 9.60.2 + patch (EWARM\_Patch\_for\_RZT2H\_N2H\_rev1.0)

RENESAS FSP Smart Configurator (FSP SC) 2024-10

RENESAS Flexible Software Package (FSP) for RZ/N2 v2.1.0

##### (2) 実行環境 ICE

IAR I-jet

##### (3) サンプルプログラムのビルド手順

サンプルプログラムのビルド手順は以下の通り。

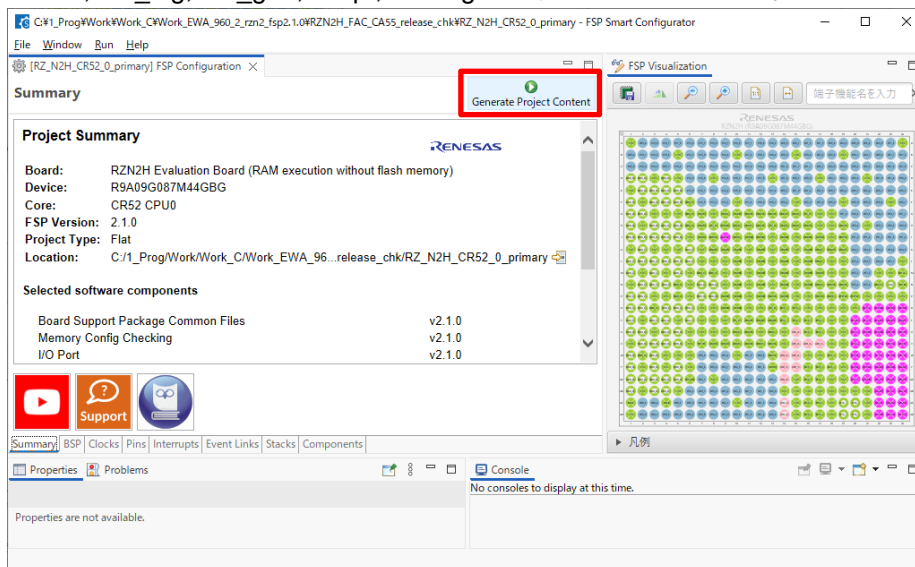
- 1 RZ\_N2H\_CA55\_fac.zip を展開し、展開したソースファイルを任意の場所にコピーする。
- 2 EWARM を起動する。
- 3 [ファイル]メニュー→[ワークスペースを開く]を選択する。
- 4 展開したソースファイルの RZ\_N2H\_CR52\_0\_primary → RZ\_N2H\_CR52\_0\_primary.eww を開く。
- 5 EWARM IDE の[ツール]メニューから FSP Smart Configurator を起動する。<sup>注</sup>

【注】 次の手順で、EWARM IDE の[ツール]メニューに FSP Smart Configurator の起動が加えられます。EWARM IDE の[ツール]メニュー→[ツールの設定]を選択する。  
[新規作成]ボタンを選択し、各フィールドに表の文字列を指定して、[OK]を押す。

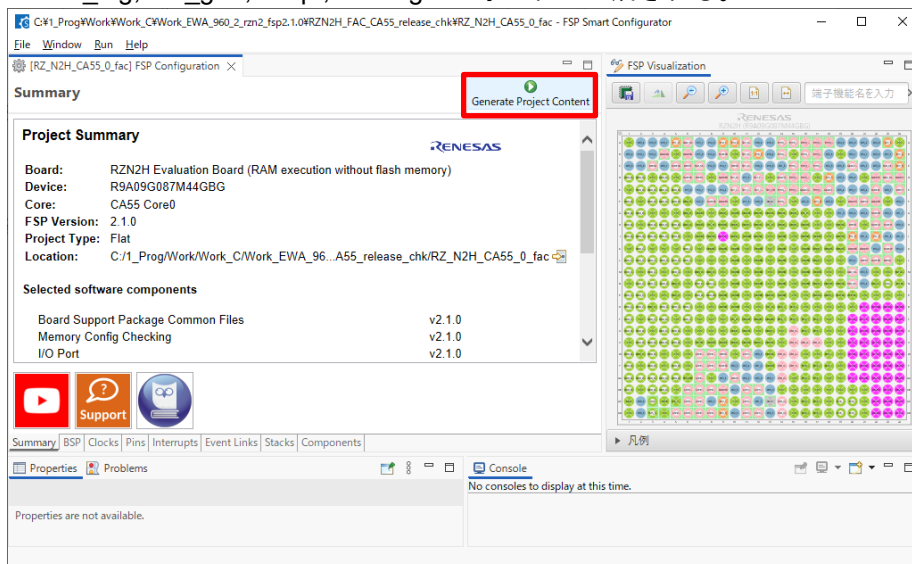
フィールド	文字列
メニューテキスト	FSP Smart Configurator
コマンド	\$RASC_EXE_PATH\$
引数	--compiler IAR configuration.xml
初期ディレクトリ	\$PROJ_DIR\$

コマンド欄の文字列は、Smart Configurator 実行ファイル rasc.exe のパスを保持する変数です。  
なお、コマンドプロンプトから FSP Smart Configurator のインストール先フォルダを直接指定して起動することもできます。

- 6 Smart Configurator の FSP Configuration ペインで、Generate Project Content をクリックする。  
rzn, rzn\_cfg, rzn\_gen, script, .settings フォルダが生成される。



- 7 プロジェクト生成が完了したら、Smart Configurator を閉じる。  
8 EWARM の[プロジェクト]メニュー→[すべてを再ビルド]を選択する。  
ファイル Debug¥Exe¥RZ\_N2H\_CR52\_0\_primary.sbd が生成される。  
9 [ファイル]メニュー→[ワークスペースを開く]を選択する。  
10 展開したソースファイルの RZ\_N2H\_CA55\_0\_fac → RZ\_N2H\_CA55\_0\_fac.eww を開く。(開く際に、プライマリプロジェクトの RZ\_N2H\_CR52\_0\_primary.sbd が参照されます。必ず先にプライマリプロジェクトをビルドしておいてください。)  
11 EWARM IDE の[ツール]メニューから FSP Smart Configurator を起動する。  
12 Smart Configurator の FSP Configuration ペインで、Generate Project Content をクリックする。rzn, rzn\_cfg, rzn\_gen, script, .settings フォルダが生成される。



- 13 プロジェクト生成が完了したら、Smart Configurator を閉じる。  
14 EWARM の[プロジェクト]メニュー→[すべてを再ビルド]を選択する。  
ファイル Debug¥Exe¥RZ\_N2H\_CA55\_0\_fac.out が生成される。

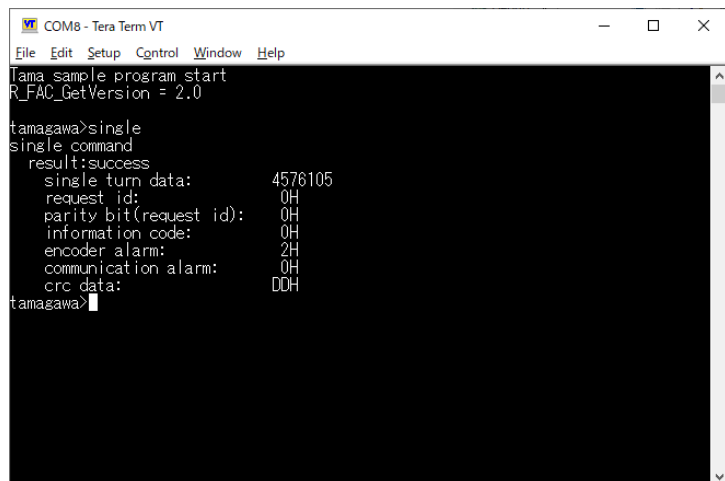
#### (4) サンプルプログラムの実行手順

「ビルド手順」を実行後、ターゲットボード、デバッガを正しく接続し、以下の操作を行う。

- 1 RZ\_N2H\_CR52\_0\_primary のワークスペースで[プロジェクト]メニュー→[ダウンロードしてデバッグ]を選択する。RZ\_N2H\_CA55\_0\_fac のプロジェクトが立ち上がる。
- 2 RZ\_N2H\_CR52\_0\_primary のワークスペースで[デバッグ]メニュー→[実行]を選択する。CA55 起動用プログラムが実行される。
- 3 RZ\_N2H\_CA55\_0\_fac のワークスペースで[デバッグ]メニュー→[実行]を選択する。サンプルプログラムが実行される。

#### (5) サンプルプログラムの実行結果

サンプルプログラムを実行し、ターミナルソフトのウィンドウにコマンドを入力してください。コマンドについては、RZ/N2H グループ FA-CODER サンプルプログラム アプリケーションノートの 4.11.7 コンソールコマンドを参照してください。



```
COM8 - Tera Term VT
File Edit Setup Control Window Help
Tama sample program start
R_FAC_GetVersion = 2.0

tamagawa>single
single command
result:success
single turn data:      4576105
request id:           0H
parity bit(request id): 0H
information code:     0H
encoder alarm:        2H
communication alarm:  0H
crc data:             00H
tamagawa>
```

### 3.4.3 e<sup>2</sup> studio: RENESAS 社製

#### (1) ビルド環境

RENESAS e<sup>2</sup> studio 2024-10

Toolchain version:

GNU ARM Embedded 12.2.1.arm-12-24 (RZ\_N2H\_CR52\_0\_primary で使用)

GCC ARM A-Profile (AArch64 bare-metal) 10.3.1.20210621 (RZ\_N2H\_CA55\_0\_fac で使用)

RENESAS Flexible Software Package (FSP) for RZ/N2 v2.1.0

#### (2) 実行環境 ICE

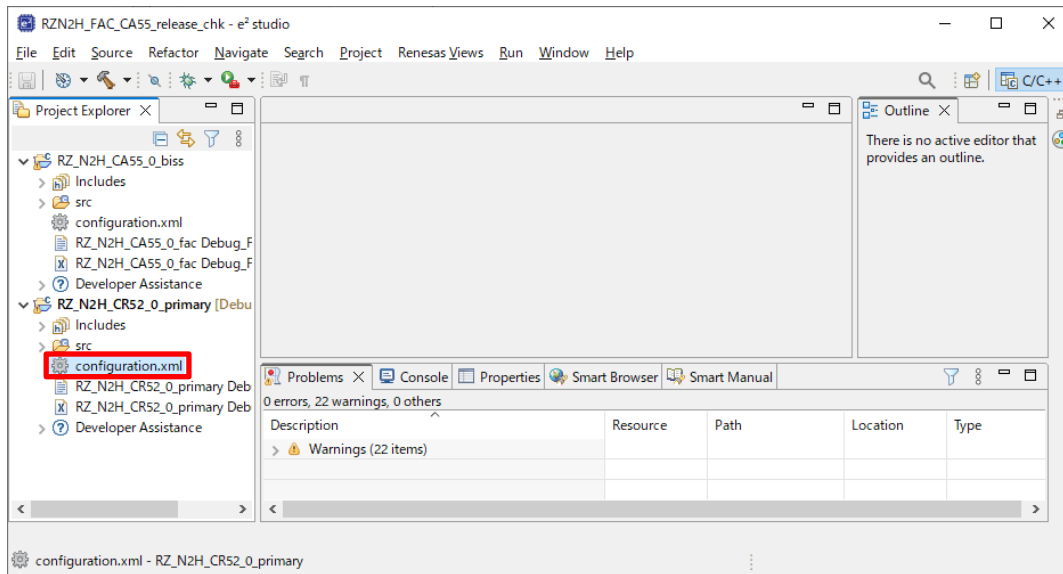
SEGGER J-Link v7.98c

#### (3) サンプルプログラムのビルド手順

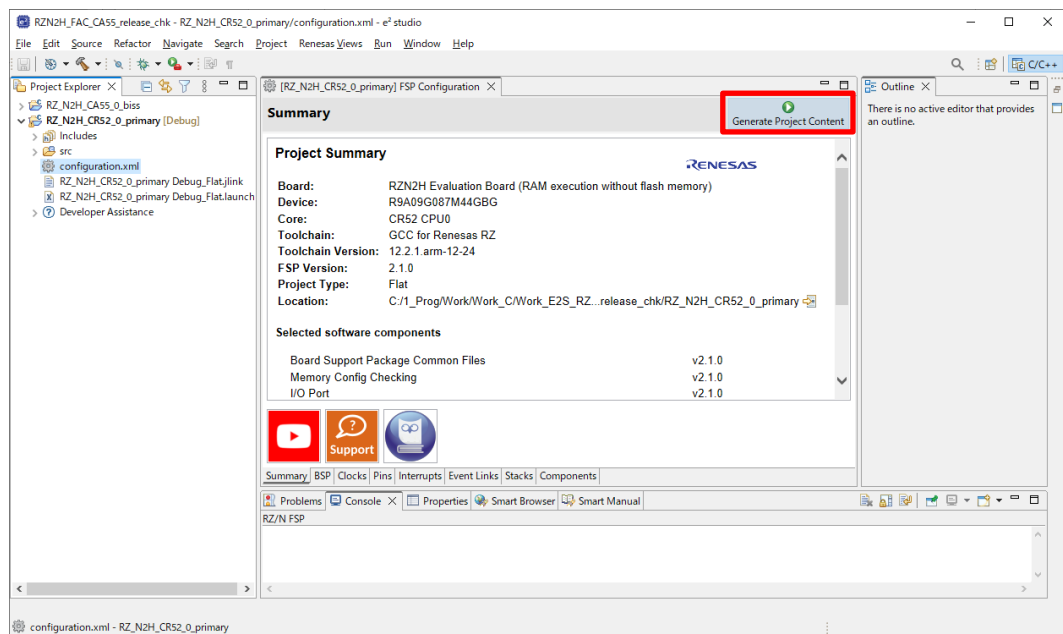
サンプルプログラムのビルド手順は以下の通り。

- 1 RZ\_N2H\_CA55\_fac.zip を展開し、展開したソースファイルを任意の場所にコピーする。
- 2 e<sup>2</sup> studio を起動し、ワークスペースへ移動後 [ファイル]メニュー→[インポート]をクリックし、一般 > 既存プロジェクトをワークスペースへ を選択して[次へ]をクリックする。
- 3 プロジェクトのインポート画面でルートディレクトリとしてサンプルプログラムを展開したフォルダを選択する。
- 4 プロジェクトを選択し、プロジェクトをワークスペースにコピー をチェックして [終了]をクリックする。

- e<sup>2</sup> studio の Project Explorer ペインで RZ\_N2H\_CR52\_0\_primary プロジェクトの configuration.xml をダブルクリックして開く。

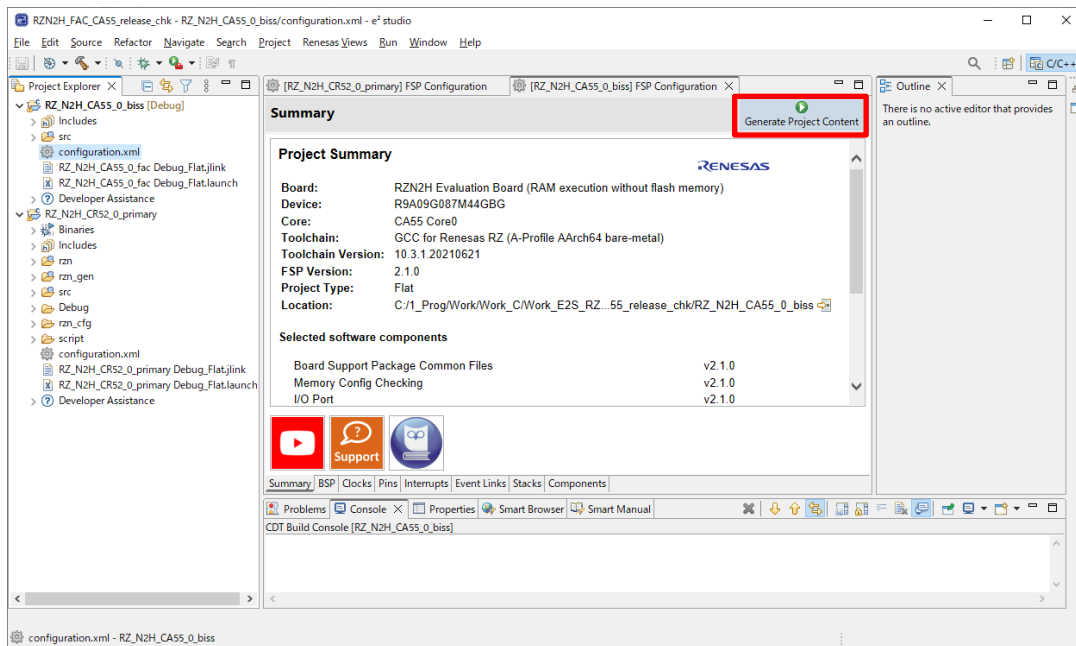


- e<sup>2</sup> studio の FSP Configuration ペインで Generate Project Content をクリックする。  
RZ\_N2H\_CR52\_0\_primary プロジェクトに rzn, rzn\_cfg, rzn\_gen, script, .settings フォルダが生成される。



- e<sup>2</sup> studio の Project Explorer ペインで RZ\_N2H\_CR52\_0\_primary を選択し、[プロジェクト]メニュー → [プロジェクトをビルド]を実行する。ファイル Debug\RZ\_N2H\_CR52\_0\_primary.sbd が生成される。
- e<sup>2</sup> studio の Project Explorer ペインで RZ\_N2H\_CA55\_0\_fac プロジェクトの configuration.xml をダブルクリックして開く。(開く際に、プライマリプロジェクトの RZ\_N2H\_CR52\_0\_primary.sbd が参照されます。必ず先にプライマリプロジェクトをビルドしておいてください。)

- 9 e2 studio の FSP Configuration ペインで Generate Project Content をクリックする。  
RZ\_N2H\_CA55\_0\_fac プロジェクトに rzn, rzn\_cfg, rzn\_gen, script, .settings フォルダが生成される。

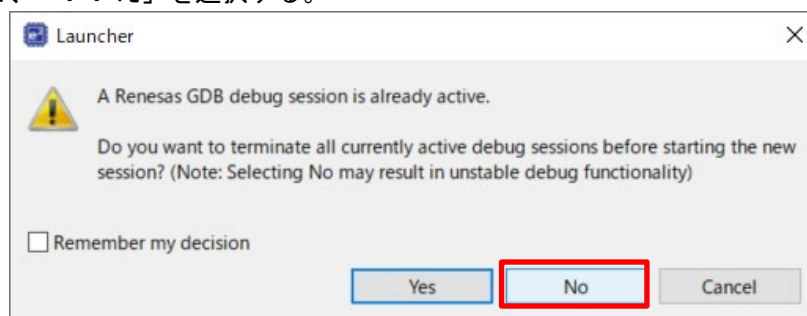


- 10 [プロジェクト]メニュー→[すべてビルド]を選択する。  
ファイル Debug¥RZ\_N2H\_CA55\_0\_fac.elf が生成される。

#### (4) サンプルプログラムの実行手順

「ビルド手順」を実行後、ターゲットボード、デバッガを正しく接続し、以下の操作を行う。

- 1 RZ\_N2H\_CR52\_0\_primary プロジェクトに対し [実行]メニュー→[デバッグ]→[Renesas GDB Hardware Debugging]を選択する。  
[デバッグ]をクリックすると内蔵 RAM へのダウンロードを開始する。
- 2 [実行]メニュー→[再開]をクリックすると、CA55 起動用プログラムが実行される。
- 3 RZ\_N2H\_CA55\_0\_fac プロジェクトに対し [実行]メニュー→[デバッグ]→[Renesas GDB Hardware Debugging]を選択する。
- 4 先に起動したデバッグセッションを停止するかを問い合わせるダイアログボックスが表示されるので、「いいえ」を選択する。

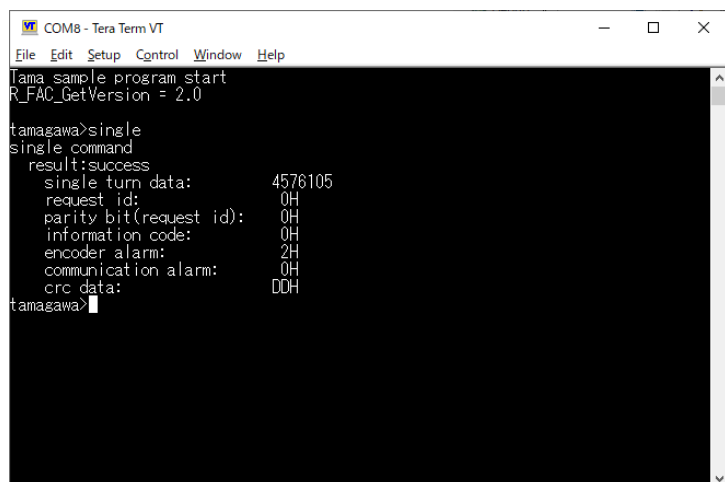


- 5 Proceed with launch が表示される場合「はい」を選択する。
- 6 [デバッグ]をクリックすると内蔵 RAM へのダウンロードを開始する。
- 7 [実行]メニュー→[再開]をクリックすると、サンプルプログラムが実行される。



## (5) サンプルプログラムの実行結果

サンプルプログラムを実行し、ターミナルソフトのウィンドウにコマンドを入力してください。コマンドについては、RZ/N2H グループ FA-CODER サンプルプログラム アプリケーションノートの 4.11.7 コンソールコマンドを参照してください。



```
COM8 - Tera Term VT
File Edit Setup Control Window Help
Tama sample program start
R_FAC_GetVersion = 2.0

tamagawa>single
single command
result:success
single turn data:      4576105
request id:           0H
parity bit(request id): 0H
information code:     0H
encoder alarm:        2H
communication alarm:  0H
crc data:             DDH
tamagawa>
```

## 改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
2.00	Dec 23.24	-	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くと、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。
  - 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。