

RL78/G23

DALI-2 Input Device Occupancy Sensor(303) サンプルアプリケーション

要旨

本アプリケーションノートは、RL78/G23 マイクロコントローラを用いた DALI (Digital Addressable Lighting Interface)の通信を行うサンプルアプリケーションについて説明します。

サンプルアプリケーションは Input Device として動作します。対応している DALI 規格は以下です。

- ・ IEC 62386-101 Edition2.1 (以下、101ed.2.1)
- ・ IEC 62386-103 Edition1.0 (以下、103ed.1.0)
- ・ IEC 62386-303 Edition1.0 (以下、303ed.1.0)

規格に対応した通信を実現するために、RL78/G23 に搭載された周辺機能を用いてマンチェスタ符号化された DALI 信号の波形を処理しています。

本アプリケーションノートは DALI についての知識を既に持っていることを前提としています。DALI 規格の詳細については 6. 参考ドキュメントを参照してください。

動作確認デバイス

RL78/G23

注意：サンプルアプリケーションを他のマイコンへ適用する場合、そのマイコンの仕様に合わせて変更し、十分評価してください。

DALI-2 認証は半導体やソフトウェアのテストではありません。お客様で DALI-2 認証対応が可能な Input Device として作りこみを行ってください。

目次

1. 仕様概要	4
2. 動作確認条件	5
3. ハードウェア説明	6
3.1 システム構成	6
3.2 使用端子一覧	8
4. ソフトウェア説明	9
4.1 環境構築方法(CC-RL).....	9
4.1.1 ソフトウェアのダウンロード	9
4.1.2 e2 studio のインストール	9
4.1.3 RL78/G23 用 Renesas Flash Driver RL78 Type 01 のインストール	9
4.1.4 RL78/G23 用 EEPROM エミュレーション・ソフトウェア RL78 Type01 のインストール	9
4.1.5 DALI マスタコントローラ GUI のインストール	10
4.1.6 EZ-0012+EZ-0012 expansion board 向けファームウェアの書き込み	10
4.1.7 サンプルアプリケーションのインポート	10
4.1.8 Renesas Flash Driver / EEPROM エミュレーション・ソフトウェアの配置	10
4.1.9 サンプルアプリケーションのビルド方法	12
4.1.10 サンプルアプリケーションのハードウェア接続方法と設定スイッチ	12
4.1.10.1 入力装置の接続方法	12
4.1.10.2 ジャンパ/スイッチ設定	13
4.1.11 サンプルアプリケーションのデバッグ方法	14
4.2 環境構築(IAR).....	15
4.2.1 ソフトウェアのダウンロード	15
4.2.2 IAR Embedded Workbench for Renesas RL78.....	15
4.2.3 DALI マスタコントローラ GUI のインストール	15
4.2.4 EZ-0012+EZ-0012 expansion board 向けファームウェアの書き込み	15
4.2.5 サンプルアプリケーションのインポート	15
4.2.6 Renesas Flash Driver / EEPROM エミュレーション・ソフトウェアの配置	15
4.2.7 サンプルアプリケーションのビルド方法	17
4.2.8 サンプルアプリケーションのハードウェア接続方法と設定スイッチ	17
4.2.8.1 入力装置の接続方法	17
4.2.8.2 ジャンパ/スイッチ設定	18
4.2.9 サンプルアプリケーションのデバッグ方法	19
4.3 動作概要	20
4.4 動作手順	24
4.5 機能概要	26
4.5.1 機能ブロック図	26
4.5.2 下位ドライバ層	27
4.5.2.1 CGC.....	27
4.5.2.2 PORT.....	28
4.5.2.3 TAU	30
4.5.2.4 TRNG	30

4.5.2.5	SMS.....	31
4.5.2.6	TML32	32
4.5.2.7	SAU	32
4.5.2.8	WDT	32
4.5.2.9	RFD	32
4.5.3	上位ドライバ層	33
4.5.3.1	DALI101 通信ドライバ.....	33
4.5.3.2	EES	36
4.5.3.3	BSW	36
4.5.3.4	PIR.....	36
4.5.4	ライブラリ層	36
4.5.5	アプリケーション層	37
4.5.5.1	Input Device 実体確保	37
4.5.5.2	Input Device アプリケーション	37
4.5.6	閾値とキャリブレーション	39
4.6	ソフトウェア構成.....	40
4.6.1	フォルダ構成.....	40
4.6.2	オプション・バイトの設定一覧.....	41
4.6.3	フローチャート	42
4.6.3.1	メインフロー	42
4.6.3.2	初期化.....	43
4.6.3.3	NVM 読み込み	44
4.6.3.4	処理開始	45
4.6.3.5	Input Device タスク処理.....	46
5.	注意事項	47
6.	参考ドキュメント	47
	改訂記録	48

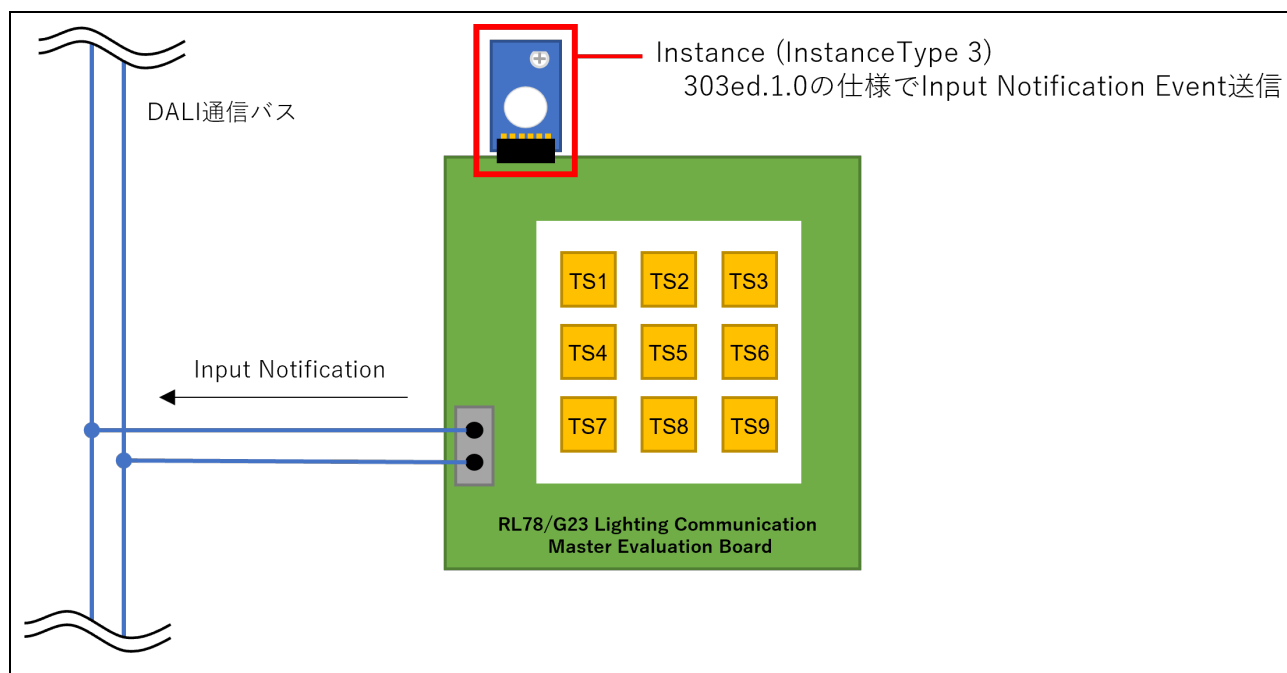
1. 仕様概要

本アプリケーションノートは、Input Device 機能を実現するサンプルアプリケーションについて記載しています。本サンプルアプリケーションの Input Device は Instance Type 3 (Occupancy Sensor)の Instance を1つ保有しており、物理的な信号処理装置として動作検知を行う人感センサ(Movement Sensor)を使用します。人感センサの動作検出信号を入力とし、303ed.1.0 の仕様に適合した Input Notification Event を発行します。

Input Notification Event は信号処理装置の状態を通知するための Forward Frame ですので、DALI subnet 上の Control Gear の調光等を行いたい場合は、DALI subnet 上の Application Controller で Input Notification Event と対応した 16bit Forward Frame を送信するアプリケーションを実装してください。

※Occupancy Sensor には Movement Sensor と Presence Sensor の 2 種類のタイプがありますが、本アプリケーションノートでは Movement Sensor を対象としたサンプルアプリケーションについて記載しています。

図 1-1 Input Device サンプルアプリケーション動作概要図



2. 動作確認条件

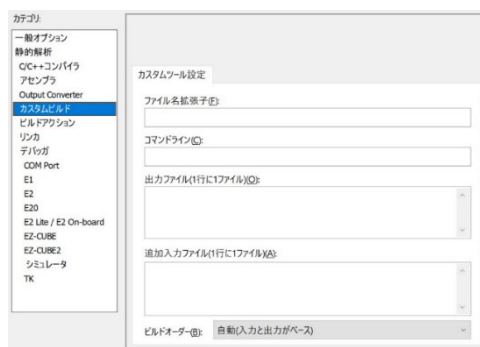
サンプルアプリケーションは以下の環境で動作確認をしております。

表 2.1 動作確認環境

項目	内容
使用マイコン	RL78/G23 (R7F100GGG2DFB)
使用ボード	RL78/G23 Lighting Communication Master Evaluation Board (RTK7RL23LMP00000BJ)
入力装置	Pmod PIR (410-389) 検出方式：パッシブ赤外線
動作周波数	高速オンチップ・オシレータ・クロック：32MHz
動作電圧	3.3V
最大消費電流 (DALI コマンド発行時)	31.30mA
統合開発環境 (e2 studio)	ルネサス エレクトロニクス製 e2 studio 2022-04
C コンパイラ (e2 studio)	ルネサス エレクトロニクス製 CC-RL V1.11.00
統合開発環境 (IAR)	IAR Systems 製 IAR Embedded Workbench IDE V4.21.4(注)
C コンパイラ (IAR)	IAR Systems 製 IAR C/C++ Compiler for Renesas RL78 V4.21.4
ライブラリ	ルネサス エレクトロニクス製 Renesas Flash Driver RL78 Type 01 V1.00 ルネサス エレクトロニクス製 EEPROM エミュレーション・ソフトウェア RL78 Type01 V1.00 ルネサス エレクトロニクス製 DALI103i ライブラリ Gen2 V1.00 ルネサス エレクトロニクス製 DALI303 ライブラリ Gen2 V1.00
スマート・コンフィグレータ (SC)	V1.3.0 [使用コンポーネント] ボードサポートパッケージ (BSP) V1.20 SNOOZE モード・シーケンサ (SMS) V1.0.1 電源検出回路 (LVD) V1.1.0

注：IAR Embedded Workbench IDE V5 系ではカスタムツールの設定を空欄に変更してください。

プロジェクト(P) - オプション(O) - カスタムビルドでカスタムツールの設定を開き、
ファイル名拡張子(F), コマンドライン(C), 出力ファイル(O), 追加入力ファイル(A)を消去



3. ハードウェア説明

3.1 システム構成

システムの構成例を以下に示します。

サンプルアプリケーションは RL78/G23 Lighting Communication Master Evaluation Board に対応した Input Device 動作を行います。Input Device は DALI システムのマスタ機器の一種に当たり、Input Device で取得した信号処理装置（スイッチ、センサ等）の情報を DALI システムに通知する機器です。

Input Device を使用する DALI システムの構成として、別途 Application Controller と Control Gear が必要となります。

本アプリケーションノートでは Application Controller として RX65N Cloud Kit+DALI-2 オプションボードもしくは RL78/G23 Lighting Communication Master Evaluation Board、Control Gear として EZ-0012+EZ-0012 expansion board を使用します。

図 3-1 システム構成例(RX65N Cloud Kit + DALI-2 オプションボード使用)

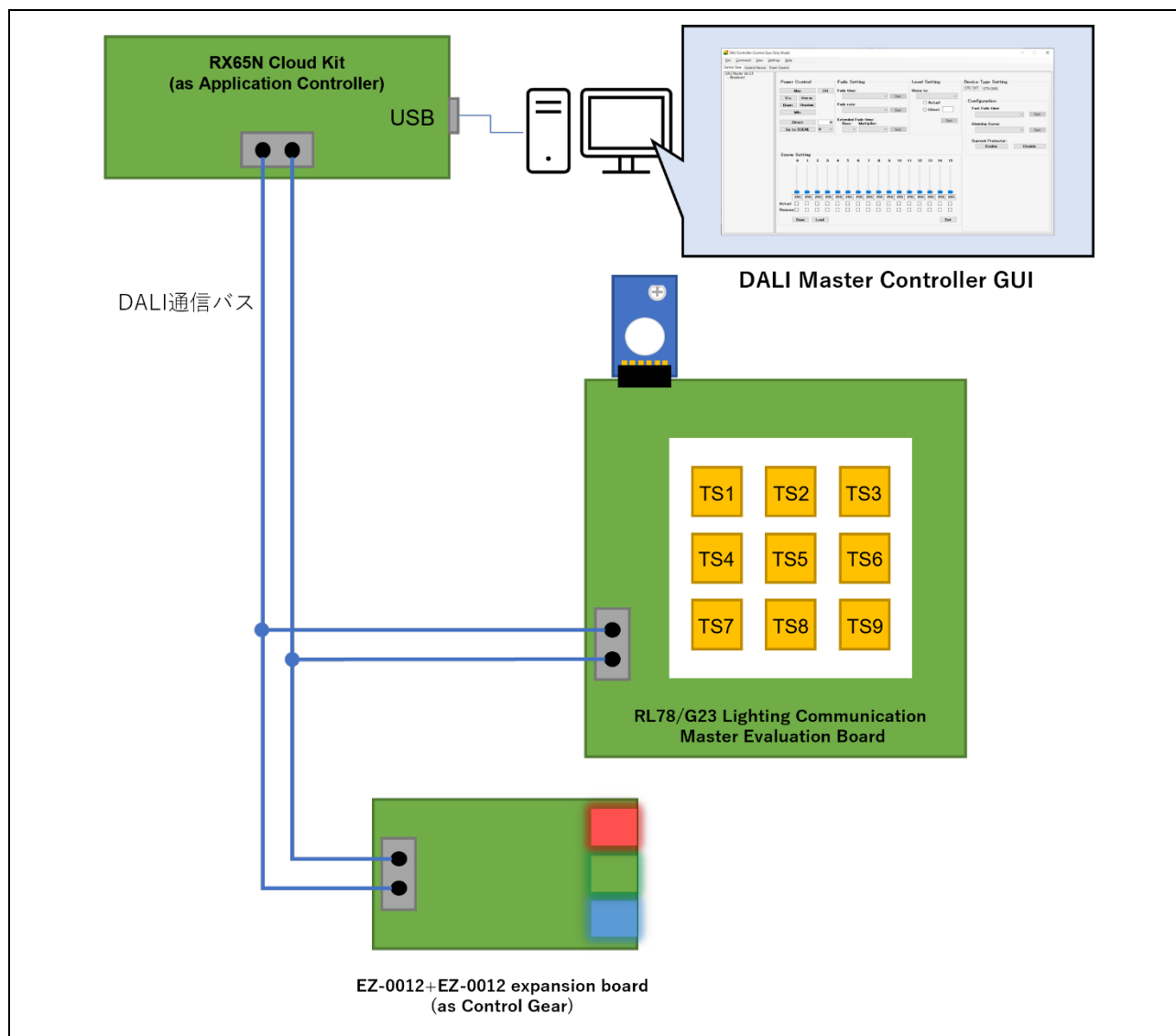
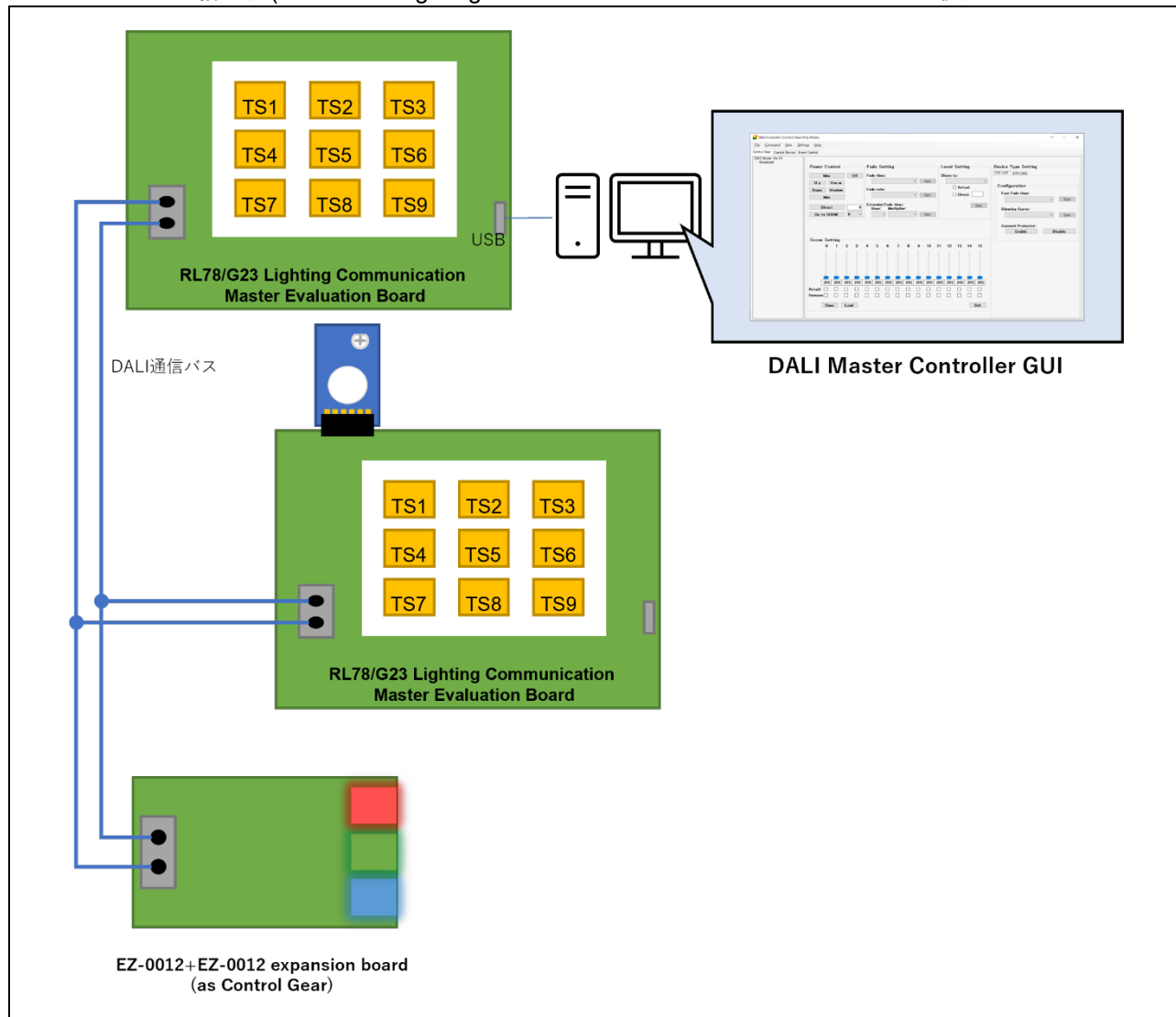


図 3-2 システム構成例(RL78/G23 Lighting Communication Master Evaluation Board 使用)



3.2 使用端子一覧

サンプルアプリケーションで使用する端子と機能を以下に示します。

表 3.1 使用端子と機能

端子名	入出力	内容
P10	出力	DALI 通信 (送信)
P16/TI01	入力	DALI 通信 (受信)
P21	入力	人感センサ入力

その他、RL78/G23 Lighting Communication Master Evaluation Board に関する機能については以下を参照してください。

- RL78/G23 照明通信マスタボード 初期ファームウェア SMS による DALI 通信変調 (R01AN6460JJ0100)
- RL78/G23 照明通信マスタ評価ボード ユーザーズ・マニュアル (R20UT5072JJ0102)

4. ソフトウェア説明

4.1 環境構築方法(CC-RL)

4.1.1 ソフトウェアのダウンロード

サンプルアプリケーションを動作させるにあたって、以下のソフトウェアが必要になります。

ルネサスエレクトロニクスのホームページからダウンロードしてください。

- e2 studio
- RL78/G23 用 Renesas Flash Driver RL78 Type 01
- RL78/G23 用 EEPROM エミュレーション・ソフトウェア RL78 Type01
- DALI マスタコントローラ GUI V3.00 以降

4.1.2 e2 studio のインストール

サンプルアプリケーションのプログラム開発およびオンチップ・デバッグを行う場合は、e2 studio が必要になります。

インストール、その他基本操作についての詳細は以下ユーザーズマニュアルを参照してください。

- 統合開発環境 e2 studio 2020-04、e2 studio v7.8 ユーザーズマニュアル 入門ガイド (R20UT4819)

4.1.3 RL78/G23 用 Renesas Flash Driver RL78 Type 01 のインストール

サンプルアプリケーションは Renesas Flash Driver を含んでいません。そのため、ルネサスエレクトロニクスのホームページからダウンロードして、ファイルを登録していただく必要があります。

インストールについては以下ユーザーズマニュアルを参照してください。ファイルの登録については、4.1.8 章を参照してください。

- RL78/G23 用 Renesas Flash Driver RL78 Type01 ユーザーズマニュアル (R20UT4830)

4.1.4 RL78/G23 用 EEPROM エミュレーション・ソフトウェア RL78 Type01 のインストール

サンプルアプリケーションは EEPROM エミュレーション・ソフトウェアを含んでいません。そのため、ルネサスエレクトロニクスのホームページからダウンロードして、ファイルを登録していただく必要があります。

インストールについては以下ユーザーズマニュアルを参照してください。ファイルの登録については、4.1.8 章を参照してください。

- RL78/G23 用 EEPROM エミュレーション・ソフトウェア RL78 Type01 ユーザーズマニュアル (R20UT5008)

4.1.5 DALI マスタコントローラ GUI のインストール

DALI マスタコントローラ GUI はサンプルアプリケーションの評価を行う際に必要となります。

インストール手順は以下ユーザーズマニュアルを参照してください。

- DALI マスタコントローラ GUI ユーザーズマニュアル (R20UT0715)

また、DALI マスタコントローラ GUI との通信を行うアプリケーションコントローラにファームウェアを書き込む必要があります。対象のファームウェアについては、DALI マスタコントローラ GUI インストーラをダウンロードいただく際に提供される ZIP ファイルに hex ファイルおよび mot ファイルが同封されておりますので、そちらをご使用ください。

4.1.6 EZ-0012+EZ-0012 expansion board 向けファームウェアの書き込み

サンプルアプリケーションの動作確認を行うための Control Gear として、ファームウェアの書き込みが必要となります。手順については以下を参照してください。

- RL78/I1A DALI-2 Control Gear 基本(102) 調光(207) 調色(209Tc) サンプルアプリケーション (R01AN6177)

4.1.7 サンプルアプリケーションのインポート

1. 提供プロジェクトファイルを解凍、任意の場所 (フォルダ) へ配置します。
2. e2 studio を起動し、プロジェクト・エクスプローラで右クリックまたは「ファイル」タブ→「インポート」を選択します。
3. 「一般」から「既存プロジェクトをワークスペースへ」を選択し、「次へ」をクリックします。
4. 「ルート・ディレクトリーの選択」を選択し、「参照」から 1.で配置したプロジェクトファイルを選択します。
5. サンプルアプリケーションを選択します。
6. 「終了」をクリックします。

4.1.8 Renesas Flash Driver / EEPROM エミュレーション・ソフトウェアの配置

サンプルアプリケーションをインポート後、以下のディレクトリにルネサスエレクトロニクスからダウンロードした Renesas Flash Driver / EEPROM エミュレーション・ソフトウェアを格納してください。

表 4.1 Renesas Flash Driver / EEPROM エミュレーション・ソフトウェアの配置場所

\DALI103i_303_sample<DIR>		
\Library<DIR>		
\RFD<DIR>		Renesas Flash Driver を格納するフォルダです。
\userown		本サンプルアプリケーションに合わせて設定
r_rfd_common_userown.c		
\include		RFD からコピーしてください。
\rfd		
r_rfd.h		
r_rfd_compiler.h		
r_rfd_device.h		
r_rfd_memmap.h		
r_rfd_types.h		
r_typedefs.h		
r_rfd_common_api.h		
r_rfd_common_control_api.h		
r_rfd_common_userown.h		
r_rfd_data_flash_api.h		
\source		
\commom		
r_rfd_common_api.c		
r_rfd_common_control_api.c		
\dataflash		
r_rfd_data_flash_api.c		
\EES<DIR>		EEPROM エミュレーション・ソフトウェアを格納するフォルダです。
\userown		本サンプルアプリケーションに合わせて設定
\include		
r_ees_descriptor.h		
r_ees_user_types.h		
r_ees_descriptor.c		
\include		EES からコピーしてください。
\ees		
r_ees.h		
r_ees_compiler.h		
r_ees_defines.h		
r_ees_device.h		
r_ees_memmap.h		
r_ees_types.h		
r_typedefs.h		
r_ees_api.h		
r_ees_exrfd_api.h		
r_ees_sub_api.h		
\source		
\ees		
r_ees_api.c		
r_ees_exrfd_api.c		
r_ees_sub_api.c		

4.1.9 サンプルアプリケーションのビルド方法

1. プロジェクト・エクスプローラでプロジェクトを右クリックし、「プロジェクトのビルド」を選択します。
2. ビルドが開始され、「コンソール」にビルドの状況が表示されるので "Build Finished" というメッセージが表示されたらビルド完了です。

4.1.10 サンプルアプリケーションのハードウェア接続方法と設定スイッチ

4.1.10.1 入力装置の接続方法

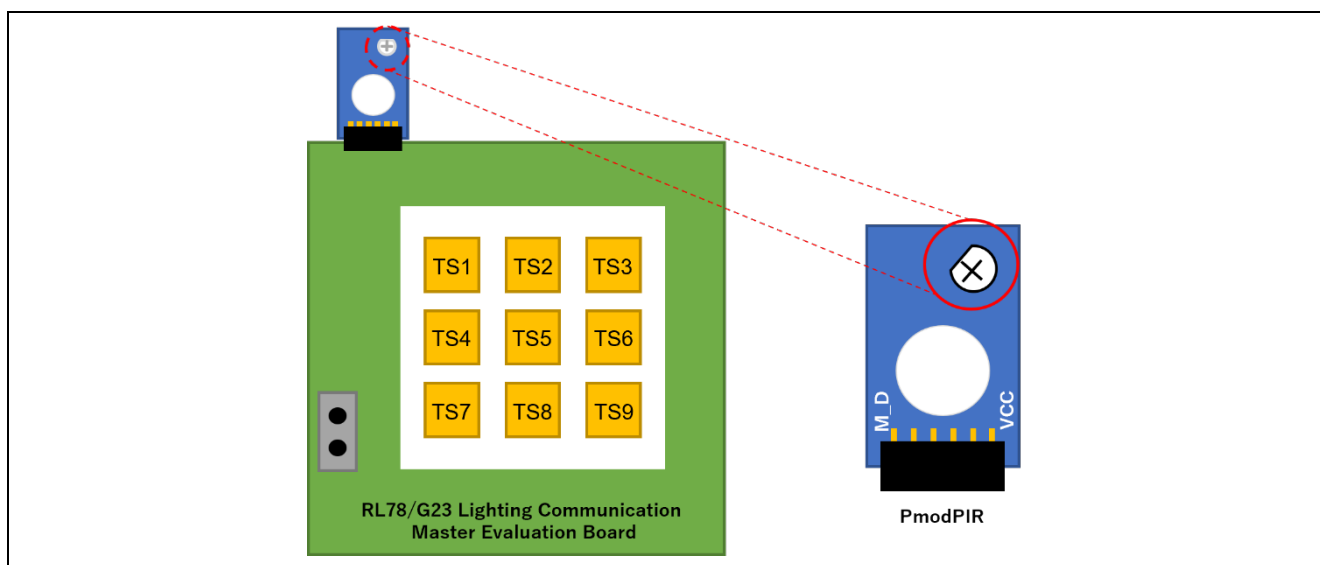
入力装置となる Occupancy Sensor(PmodPIR)と RL78/G23 Lighting Communication Master Evaluation Board の接続を以下に示します。

表 4.2 Occupancy Sensor 接続

RL78/G23 Lighting Communication Master Evaluation Board	Occupancy Sensor (PmodPIR)
CN1-7 (P21/GPIO)	J1 M_D
CN1-8 (P22/GPIO)	-
CN1-9 (P23/GPIO)	-
CN1-10 (P24/GPIO)	-
CN1-11 (GND)	J1 GND
CN1-12 (3.3V)	J1 VCC

Occupancy Sensor に付随する可変抵抗は、動作検知信号の出力時間を可変するものです。可変抵抗は以下の設定とします。下図の赤丸で示す部位が可変抵抗であり、手動で回転させることで設定を行います。

図 4-1 PmodPIR 可変抵抗設定位置



4.1.10.2 ジャンパ/スイッチ設定

本サンプルアプリケーション動作時には、RL78/G23 Lighting Communication Master Evaluation Board のジャンパ等を以下に設定する必要があります。

図 4-2 RL78/G23 Lighting Communication Master Evaluation Board コンポーネント

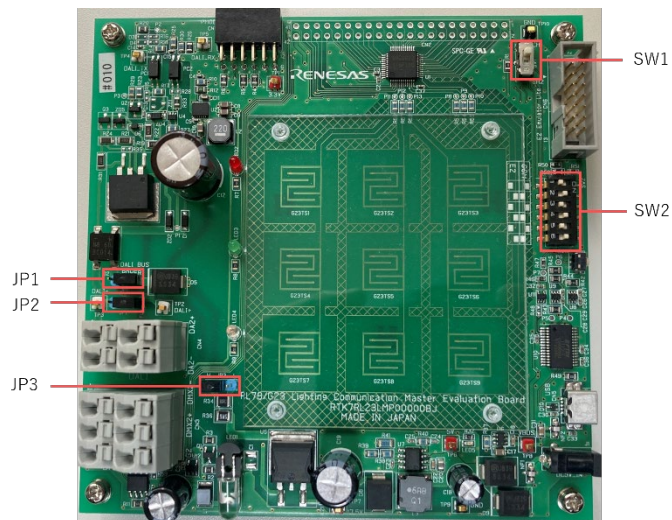



表 4.3 スイッチ設定一覧

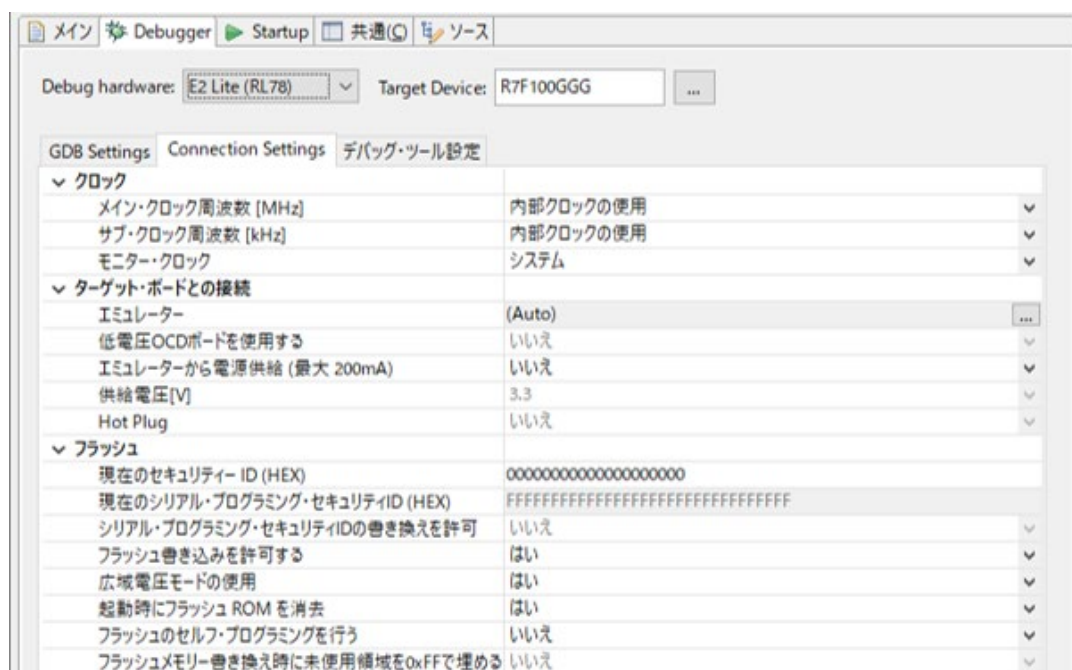
JP/SW 番号	設定
JP1	オープン
JP2	オープン
JP3	H 側とショート
SW1	don't care
SW2-1	OFF
SW2-2	ON
SW2-3	OFF
SW2-4	ON
SW2-5	OFF
SW2-6	ON

4.1.11 サンプルアプリケーションのデバッグ方法

1. e2 studio でサンプルアプリケーションをインポート後、 ボタンをクリックしてマイクロコントローラへプログラムをダウンロードします。
2. 「実行」→「デバッグの構成…」を選択し、「デバッグ構成」ウィンドウを開きます。
3. 「デバッグ構成」ウィンドウで“Renesas GDB Hardware Debugging” デバッグ構成の表示を展開し、既存のデバッグ構成をクリックします。
4. 「debugger」→「Connection Settings」タブに切り替え、下図の設定となっていることを確認してください。

※E2 エミュレータ Lite を使用の場合は Debug hardware : E2 Lite (RL78)を選択してください。

図 4-3 デバッグ画面設定内容



5. 「デバッグの開始」を選択し、「デバッグ」ビュー画面が表示されたらデバッグの準備が完了です。
デバック方法デバック方法、その他基本操作についての詳細は以下ユーザーズマニュアルを参照してください。

- 統合開発環境 e2 studio 2020-04、e2 studio v7.8 ユーザーズマニュアル 入門ガイド (R20UT4819)

4.2 環境構築(IAR)

4.2.1 ソフトウェアのダウンロード

サンプルアプリケーションを動作させるにあたって、以下のソフトウェアが必要になります。

IAR Systems のホームページからダウンロードしてください。

4.2.2 IAR Embedded Workbench for Renesas RL78

- サンプルアプリケーションのプログラム開発およびオンチップ・デバッグを行う場合は、IAR Embedded Workbench for Renesas RL78 が必要になります。

インストールについては下記動画を参照ください。

[IAR Embedded Workbench for RL78 でのスマート・コンフィグレータの使い方\(1/2\) - インストール](#)

4.2.3 DALI マスタコントローラ GUI のインストール

DALI マスタコントローラ GUI はサンプルアプリケーションの評価を行う際に必要となります。

インストール手順は以下ユーザズマニュアルを参照してください。

- DALI マスタコントローラ GUI ユーザズマニュアル (R20UT0715)

また、DALI マスタコントローラ GUI との通信を行うアプリケーションコントローラにファームウェアを書き込む必要があります。対象のファームウェアについては、DALI マスタコントローラ GUI インストーラをダウンロードいただく際に提供される ZIP ファイルに hex ファイルおよび mot ファイルが同封されておりますので、そちらをご使用ください。

4.2.4 EZ-0012+EZ-0012 expansion board 向けファームウェアの書き込み

サンプルアプリケーションの動作確認を行うための Control Gear として、ファームウェアの書き込みが必要となります。手順については以下を参照してください。

- RL78/I1A DALI-2 Control Gear 基本(102) 調光(207) 調色(209Tc) サンプルアプリケーション (R01AN6177)

4.2.5 サンプルアプリケーションのインポート

1. IAR Embedded Workbench for Renesas RL78 を起動します。
2. 「ファイル」タブ→「新規ワークスペース」を選択します。
3. 「ファイル」から「名前を付けてワークスペース保存」を選択し任意の名前、場所(フォルダ)にワークスペースを配置します。
4. 提供プロジェクトファイルを解凍、作成したワークスペース (フォルダ)へ配置します。
5. 「プロジェクト」から「既存プロジェクトの追加」を選択し、9.で配置したプロジェクトファイル(EWP ファイル)を選択します。
6. 「ファイル」から「終了」をクリックします。

4.2.6 Renesas Flash Driver / EEPROM エミュレーション・ソフトウェアの配置

サンプルアプリケーションをインポート後、以下のディレクトリにルネサスエレクトロニクスからダウンロードした Renesas Flash Driver / EEPROM エミュレーション・ソフトウェアを格納してください。

表 4.4 Renesas Flash Driver / EEPROM エミュレーション・ソフトウェアの配置場所

\DALI103i_303_sample<DIR>		
\Library<DIR>		
\RFD<DIR>		Renesas Flash Driver を格納するフォルダです。
\userown		本サンプルアプリケーションに合わせて設定
r_rfd_common_userown.c		
\include		RFD からコピーしてください。
\rfd		
r_rfd.h		
r_rfd_compiler.h		
r_rfd_device.h		
r_rfd_memmap.h		
r_rfd_types.h		
r_typedefs.h		
r_rfd_common_api.h		
r_rfd_common_control_api.h		
r_rfd_common_userown.h		
r_rfd_data_flash_api.h		
\source		
\commom		
r_rfd_common_api.c		
r_rfd_common_control_api.c		
\dataflash		
r_rfd_data_flash_api.c		
\EES<DIR>		EEPROM エミュレーション・ソフトウェアを格納するフォルダです。
\userown		本サンプルアプリケーションに合わせて設定
\include		
r_ees_descriptor.h		
r_ees_user_types.h		
r_ees_descriptor.c		
\include		EES からコピーしてください。
\ees		
r_ees.h		
r_ees_compiler.h		
r_ees_defines.h		
r_ees_device.h		
r_ees_memmap.h		
r_ees_types.h		
r_typedefs.h		
r_ees_api.h		
r_ees_exrfd_api.h		
r_ees_sub_api.h		
\source		
\ees		
r_ees_api.c		
r_ees_exrfd_api.c		
r_ees_sub_api.c		

4.2.7 サンプルアプリケーションのビルド方法

1. ワークスペースでプロジェクトを右クリックし、「メイク」を選択します。
2. ビルドが開始され、「コンソール」にビルドの状況が表示されるので "エラー合計数: 0" というメッセージが表示されたらビルド完了です。

4.2.8 サンプルアプリケーションのハードウェア接続方法と設定スイッチ

4.2.8.1 入力装置の接続方法

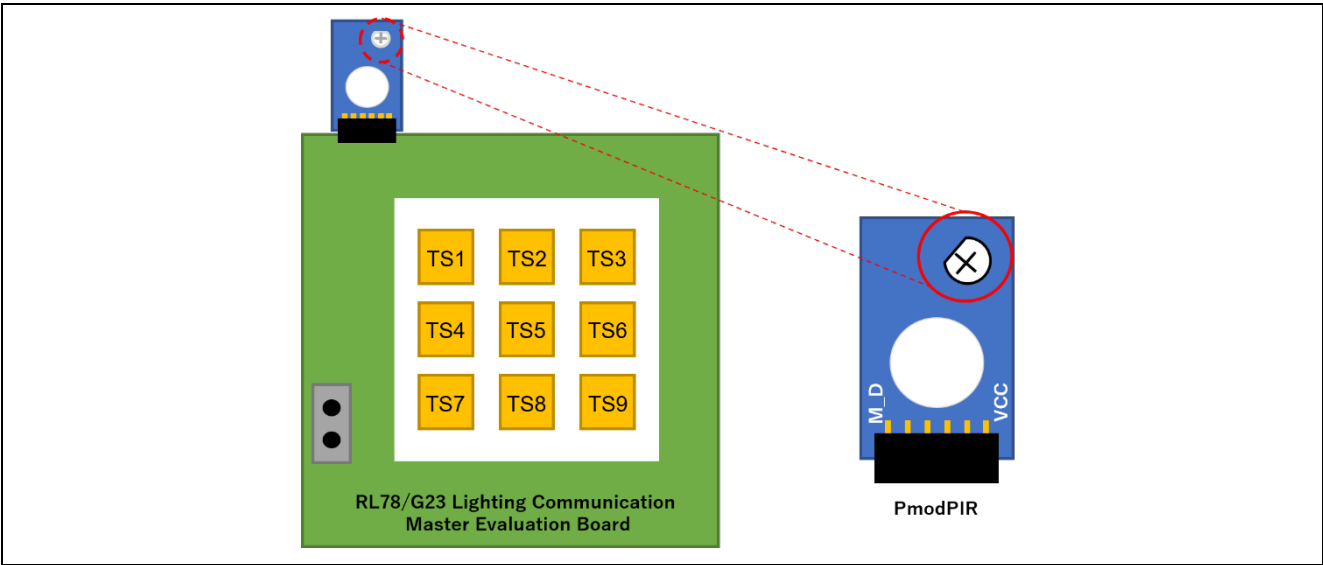
入力装置となる Occupancy Sensor(PmodPIR)と RL78/G23 Lighting Communication Master Evaluation Board の接続を以下に示します。

表 4.5 Occupancy Sensor 接続

RL78/G23 Lighting Communication Master Evaluation Board	Occupancy Sensor (PmodPIR)
CN1-7 (P21/GPIO)	J1 M_D
CN1-8 (P22/GPIO)	-
CN1-9 (P23/GPIO)	-
CN1-10 (P24/GPIO)	-
CN1-11 (GND)	J1 GND
CN1-12 (3.3V)	J1 VCC

Occupancy Sensor に付随する可変抵抗は、動作検知信号の出力時間を可変するものです。可変抵抗は以下の設定とします。下図の赤丸で示す部位が可変抵抗であり、手動で回転させることで設定を行います。

図 4-4 PmodPIR 可変抵抗設定位置



4.2.8.2 ジャンパ/スイッチ設定

本サンプルアプリケーション動作時には、RL78/G23 Lighting Communication Master Evaluation Board のジャンパ等を以下に設定する必要があります。

図 4-5 RL78/G23 Lighting Communication Master Evaluation Board コンポーネント

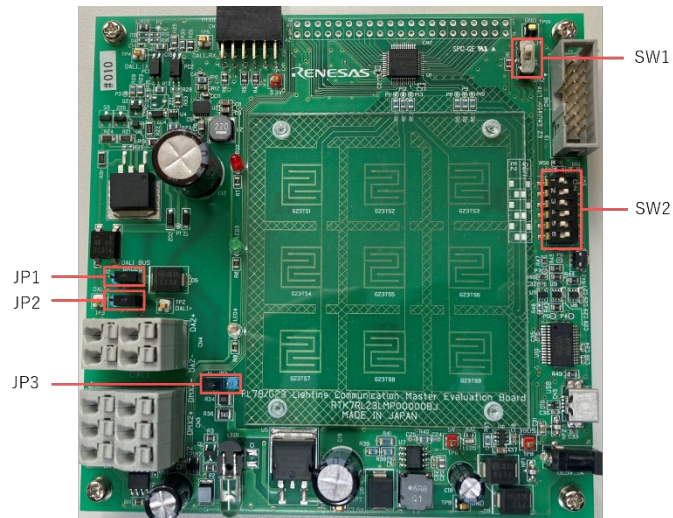



表 4.6 スイッチ設定一覧

JP/SW 番号	設定
JP1	オープン
JP2	オープン
JP3	H 側とショート
SW1	don't care
SW2-1	OFF
SW2-2	ON
SW2-3	OFF
SW2-4	ON
SW2-5	OFF
SW2-6	ON

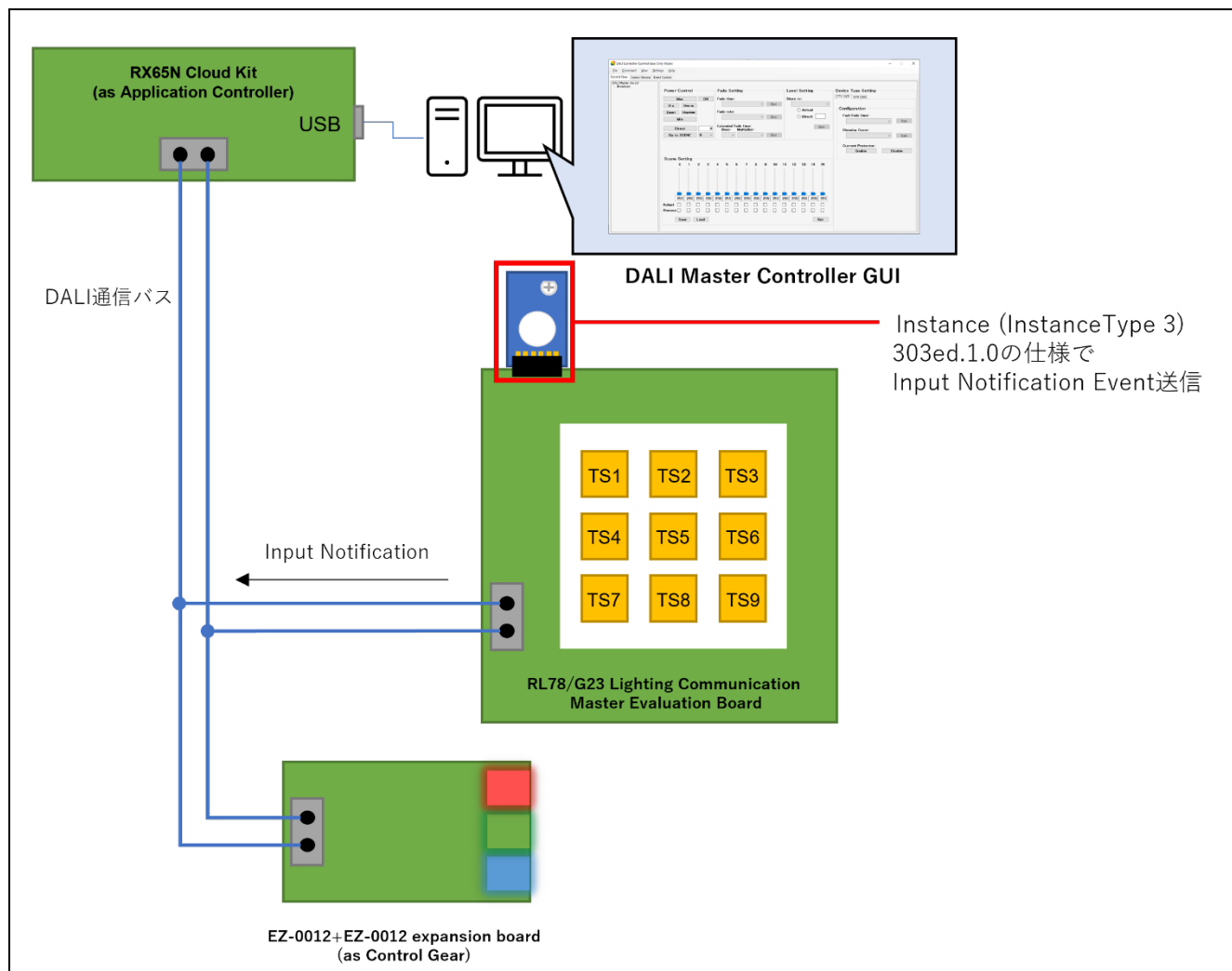
4.2.9 サンプルアプリケーションのデバッグ方法

1. IAR Embedded Workbench for Renesas RL78 でサンプルアプリケーションをインポート後、 ボタンをクリックしてマイクロコントローラへプログラムをダウンロードし、デバックを行います。

4.3 動作概要

本サンプルアプリケーションの動作概要を記載します。

図 4-6 システム構成例(RX65N Cloud kit + DALI-2 オプションボード使用)



本サンプルアプリケーションは DALI 規格における Input Device 機器として動作します。本サンプルアプリケーションにおける Input Device は Instance Type 3 の Instance を 1 つ保有しており、Instance は人感センサ(Movement Sensor)と対応付いています。Instance – 入力信号対応表を表 4.7 に記載します。

Instance はセンサの検知信号に応じて Input Notification Event を DALI 通信バスへ送信します。

表 4.7 Instance – 入力信号対応表

Instance Number	resolution (bit)	入力装置	Input Signal
0	2	Presence Sensor 使用時	空室状態を通知
			占有状態を通知
			動作無しを通知
			動作有りを通知
		Movement Sensor 使用時	動作有りを通知

※ 本サンプルアプリケーションでは Movement Sensor 使用時の動作内容となります。

また、Instance Type 3 の Instance は以下のように Event Information が定義されており、それぞれ条件を満たした際に Input Notification Event が発生します。各 Event の発生条件は IEC62386-303ed1.0 規格書を参照してください。

表 4.8 Event Information – 一覧

Event Information		説明
イベント内容	Event Info 値 (2 進値)	
No movement	00 0000 - - - 0	動作無し
Movement	00 0000 - - - 1	動作有り
Vacant	00 0000 - 00 -	空室状態
Still vacant	00 0000 - 10 -	空室状態の継続
Occupied	00 0000 - 01 -	占有状態
Still occupied	00 0000 - 10 -	占有状態の継続
Presence sensor	00 0000 0 - - -	Presence sensor である
Movement sensor	00 0000 1 - - -	Movement sensor である

※ “-” 部分は 0/1 不問となります。

なお、本サンプルアプリケーションの Input Device が発信した Event Message に対応する処理 (Control Gear に対する調光指示コマンド送信等)は同 DALI subnet 上の Application Controller に実装してください。

4.4 動作手順では、Occupied、Vacantの2イベントを発行させ、発行されたイベントに対して Control Gear の調光動作を関連付ける動作を実現します。

各イベントの発生条件を以下に示します。

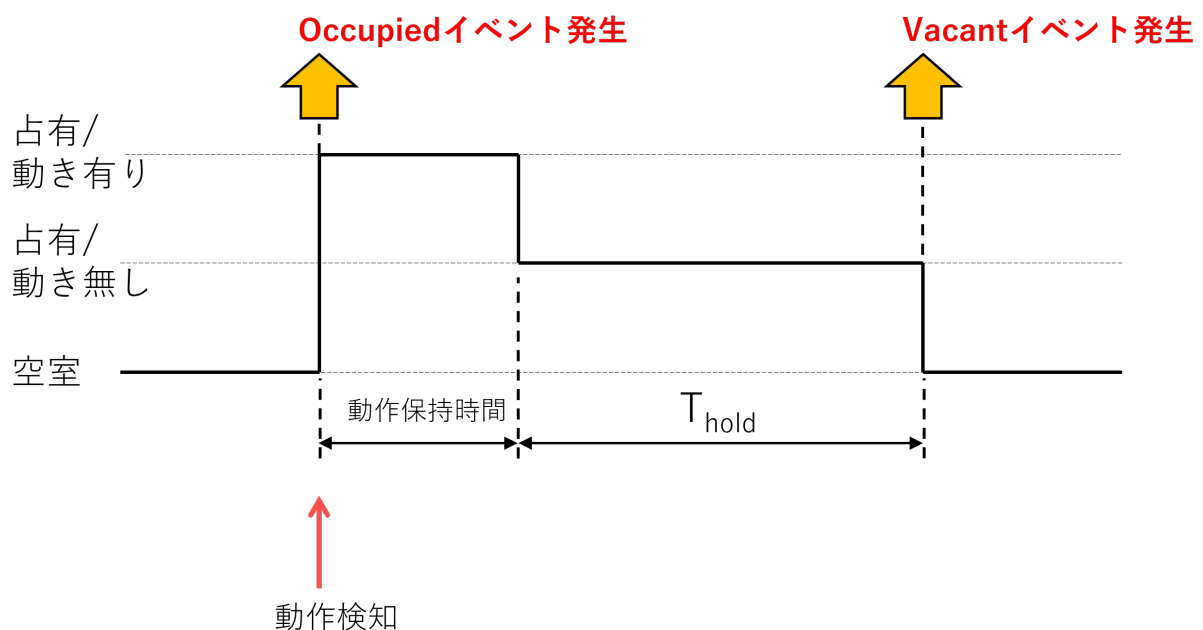
■ Occupied イベント

動作検出時にイベント発生

■ Vacant イベント

動作検出後、新たな動作検出が一定時間 (動作保持時間 + T_{hold}) なければ、イベント発生

図 4-7 Occupied イベントおよび Vacant イベント発生条件



このとき、動作保持時間と T_{hold} の時間は以下設定とします。

表 4.9 各設定時間

項目	変数設定値	T_{incr}	設定時間
T_{hold}	3	10 [s]	30 [s]
動作保持時間	3000	1[ms]	3 [s]

Occupied イベント、Vacant イベントに対し、Control Gear の調光コマンドを下表のように対応付けます。

表 4.10 発生イベント – 関連コマンド 対応表

発生イベント	Event information	関連付けするコマンド (16bit Forward Frame)
Occupied	0b 00 0000 1010	RECALL MAX LEVEL (0xFF05)
Vacant	0b 00 0000 1000	OFF (0xFF00)

これにより、ユーザ動作を行った際の Control Gear 調光は以下の動作を行います。

- 動作検出：max level で点灯
- 動作検出より一定時間経過：消灯

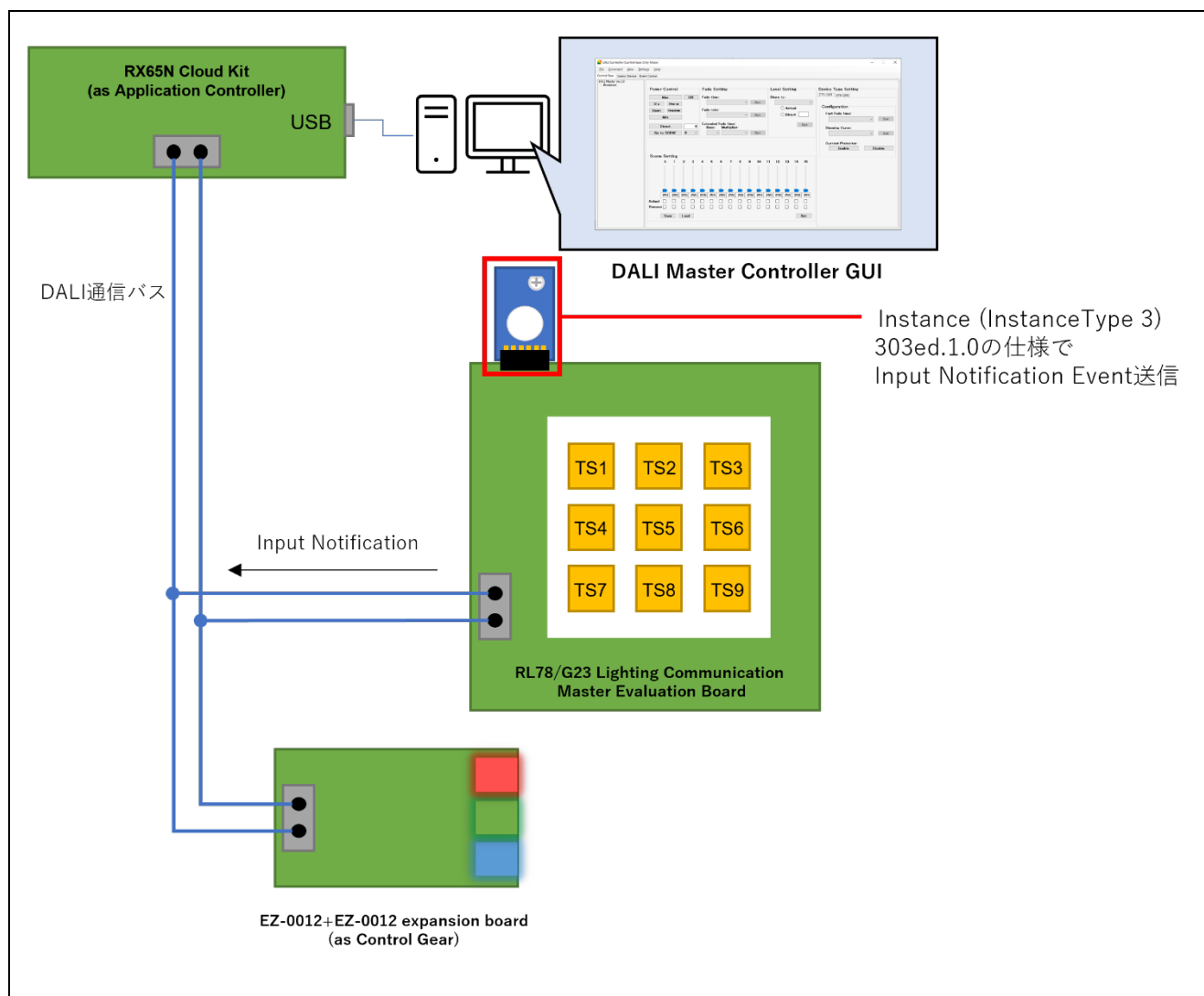
なお、発生させるイベントや関連付けるコマンドに関しては一例です。お客様が実現したい動作に合わせてイベント設定やコマンドの関連付けを行う必要があります。

4.4 動作手順

本アプリケーションノートでは、DALI マスタコントローラ GUI を使用してサンプルアプリケーションを動作させる手順を以下に示します。

Application Controller として RX65N Cloud kit + DALI-2 オプションボードを使用する例を以下に示します。RL78/G23 Lighting Communication Master Evaluation Board を Application Controller として使用する場合は 3.1 を参照して機器を接続してください。RX65N Cloud kit + DALI-2 オプションボードには AC アダプタを接続し、DALI バスに対する電源供給が十分できるように設定を行ってください。

図 4-8 システム構成例(RX65N Cloud kit + DALI-2 オプションボード使用)



[手順]

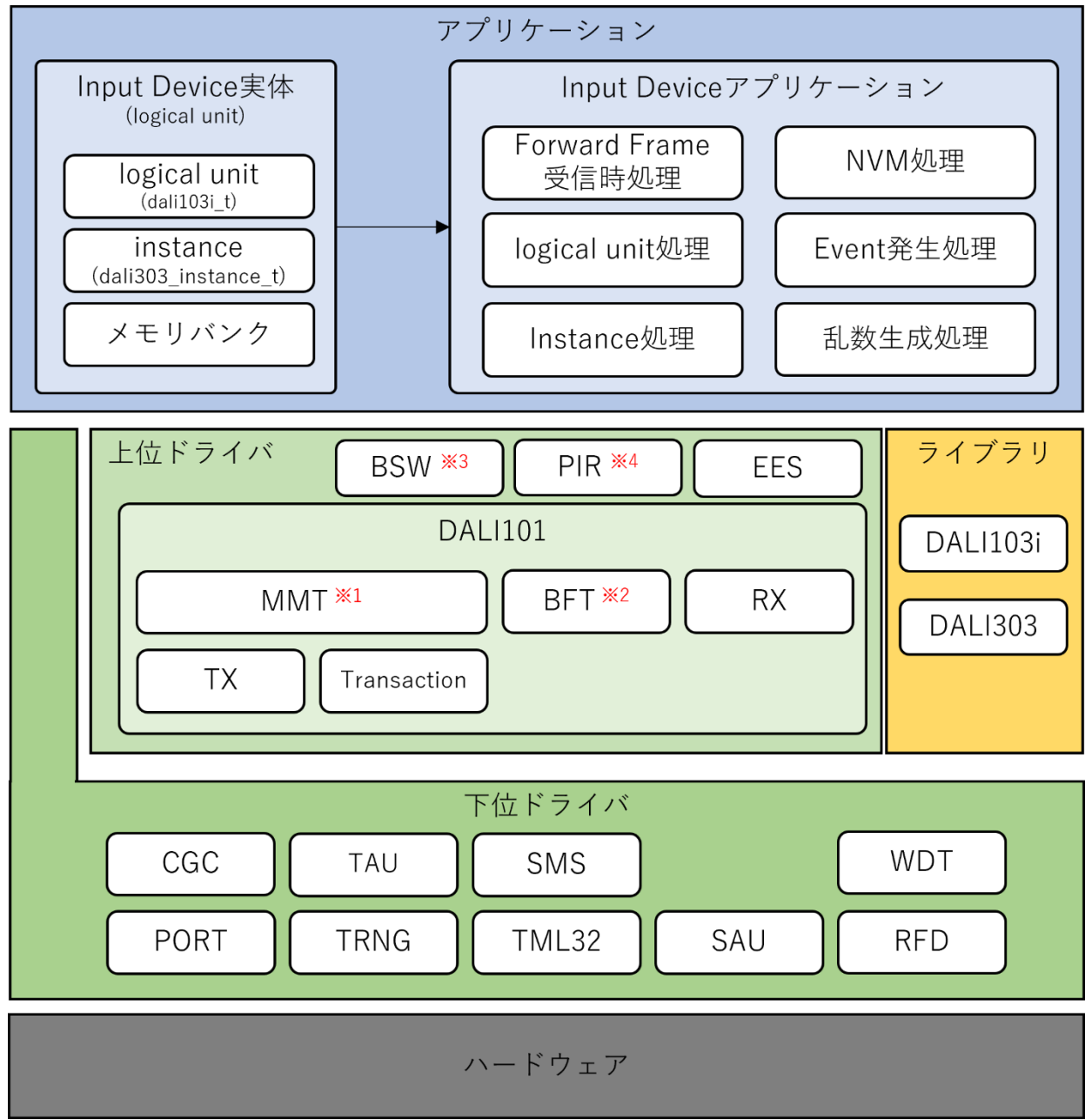
1. Application Controller として RX65N Cloud kit + DALI-2 オプションボード、RL78/G23 Lighting Communication Master Evaluation Board、EZ-0012+EZ-0012 expansion board を接続後、各装置に電源を供給します。
2. サンプルアプリケーションをビルド後、RL78/G23 Lighting Communication Master Evaluation Board へデバッグからプログラムをダウンロードし、デバッグ実行します。
3. DALI マスタコントローラ GUI を起動後、RX65N Cloud kit + DALI-2 オプションボードとシリアル接続します。
4. DALI マスタコントローラ GUI の Control Device タブ から RL78/G23 Lighting Communication Master Evaluation Board にショートアドレス、グループ番号、インスタンスグループ番号を付与します。
5. Manual Command (By Code)画面を開き、Control Device タブを選択します。
6. Manual Command (By Code)で RESET (0xFFFE10 を Twice 送信)を送信します。
7. Manual Command (By Code)で DTR0 (0xC13003 を Once 送信)を送信します。
8. Manual Command (By Code)で SET HOLD TIMER (0xFFFF21 を Twice 送信)を送信します。
9. DALI マスタコントローラ GUI の Event Control タブを選択します。
10. 人感センサに手をかざし、動作検出を行うことで Occupied イベントが発生させます。
このとき、DALI マスタコントローラ GUI の Event Control タブ、"Receive Event"に自動でイベントメッセージが登録されます。
11. 人感センサに動作検出をさせず、Vacant イベントが発生するまで待機します。
このとき、DALI マスタコントローラ GUI の Event Control タブ、"Receive Event"に自動でイベントメッセージが登録されます。
12. DALI マスタコントローラ GUI の Event Control タブにて Occupied イベントに対し、「Setting」ボタンを押下し、受信したイベントメッセージに対する送信フレームとして Broadcast で RECALL MAX LEVEL (0xFF05)を設定します。
13. 同様に Vacant イベントに対し、「Setting」ボタンを押下し、受信したイベントメッセージに対する送信フレームとして Broadcast で OFF (0xFF00)を設定します。
14. 人感センサに手をかざした際に EZ-0012+EZ-0012 expansion board 上の LED が点灯することを確認します。
15. 一定時間、動作検知がなければ EZ-0012+EZ-0012 expansion board 上の LED が消灯することを確認します。

4.5 機能概要

4.5.1 機能ブロック図

サンプルアプリケーションの機能ブロック図を以下に示します。

図 4-9 機能ブロック図



- ※1 MMT . . . Multi Master Transmitter
- ※2 BFT . . . Backward Frame Transmitter
- ※3 BSW . . . Binary Switch
- ※4 PIR . . . PmodPIR

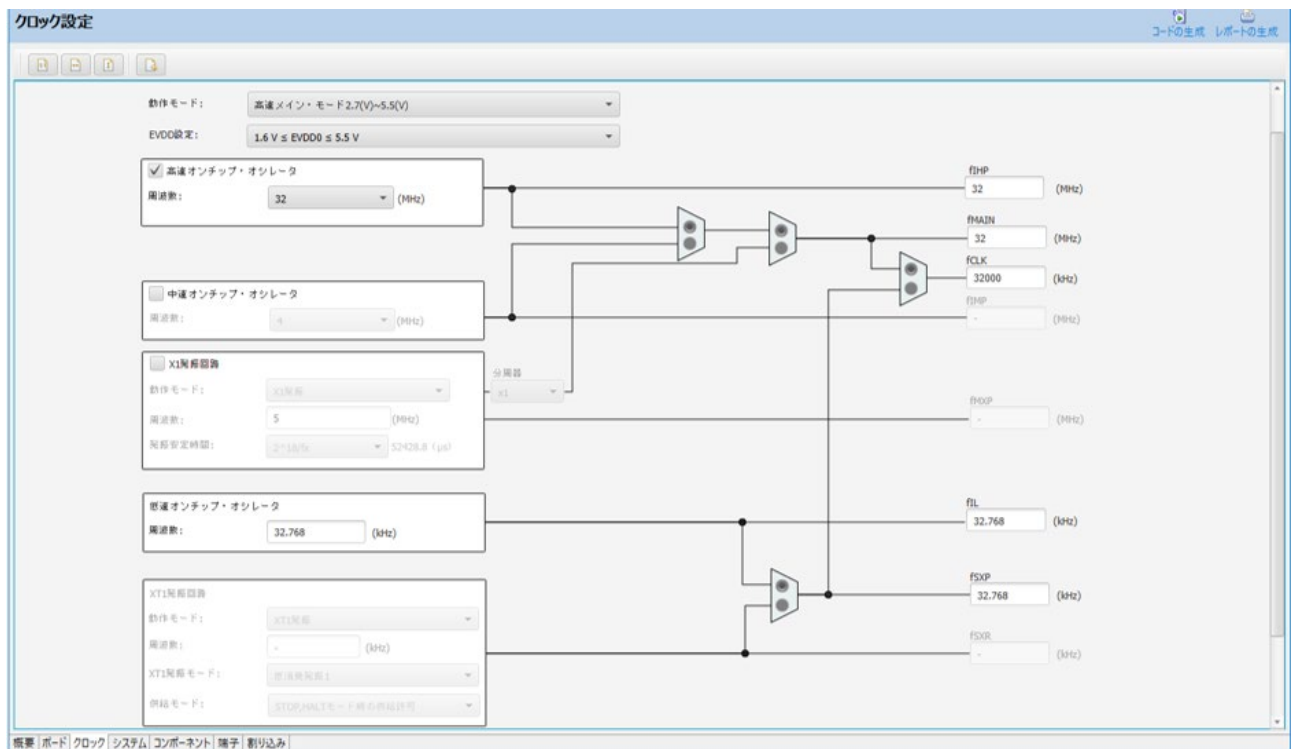
4.5.2 下位ドライバ層

RL78/G23 の周辺機能を使ったシンプルなドライバ層です。

4.5.2.1 CGC

クロック生成を行うドライバです。本サンプルアプリケーションではスマートコンフィギュレータ(SC)を使用し、以下の設定を行っています。

図 4-10 クロック設定



4.5.2.2 PORT

RL78/G23 マイコンのポート制御を行うドライバです。本サンプルアプリケーションでは、搭載されているマイコンの全ポートの制御を可能にするドライバを実装して必要に応じて呼び出しを行っています。

表 4.11 ポート設定(1/2)

内容	入出力	用途
P01/RxD1	入力	DMX512 用
P00/TxD1	出力	DMX512 用
P10	出力	DALI 送信用
P11	入力	デバッグ接続用
P12	出力	デバッグ接続用
P13	出力	未使用
P14	出力	未使用
P15	出力	未使用
P16/TI01	入力	DALI 受信(入力パルス幅測定)用
P17	出力	未使用
P20	出力	DMX512
P21	入力	PMOD(Movement Sensor 入力)
P22	出力	PMOD
P23	出力	PMOD
P24	出力	PMOD
P25	入力	未使用
P26	出力	未使用
P27	出力	未使用
P30	入力	未使用
P31/TS01	出力	タッチキースキャン出力 1
P40/TOOL0	入力	デバッグ接続用
P41/TO07	出力	未使用
P50/TS00	入力	タッチキースキャン入力 3
P51	出力	未使用
P60/SCLA0	出力	PMOD
P61/SDAA0	出力	PMOD
P62	出力	LED (赤)
P63	出力	LED (緑)

表 4.12 ポート設定(2/2)

内容	入出力	用途
P70/TS02	入力	タッチキースキャン入力 2
P71/TS03	入力	タッチキースキャン入力 1
P72/TS04	出力	LED (白)
P73/TS05	出力	未使用
P74/TS06	出力	タッチキースキャン出力 3
P75/TS07	出力	タッチキースキャン出力 2
P120	出力	未使用
P121	出力	未使用
P122	出力	未使用
P123	入力(入力専用ポート)	未使用
P124	入力(入力専用ポート)	未使用
P130	出力(出力専用ポート)	未使用
P137/INTP0	入力(入力専用ポート)	未使用
P140	出力	未使用
P146 ^注	FOR_PROBIT_LAB マクロ無効時 : 出力 FOR_PROBIT_LAB マクロ有効時 : 入力(内部プルアップ抵抗有効)	FOR_PROBIT_LAB マクロ無効時 : 未使用 FOR_PROBIT_LAB マクロ有効時 : ProbitLab の digital switch 4 接続状態入力
P147	出力	未使用
RESET	-	未使用
REGC	-	未使用
VSS	-	未使用
VDD	-	未使用

注 : FOR_PROBIT_LAB マクロは DALI-2 認証を受けるためのテストシーケンスを実施する際のみ必要なマクロです。テストシーケンス実施の際は FOR_PROBIT_LAB マクロを有効にしたうえで、P146 と GND の各端子を ProbitLab の digital switch 4 に接続し、テストシーケンスを実施してください。テストシーケンスを実施しない場合は Occupancy Sensor の入力状態の誤検出を避けるため無効にしてください。FOR_PROBIT_LAB マクロは以下の場所にて定義しています。

e2studio : プロジェクトのプロパティから「C/C++ ビルド」→「設定」→「ツール設定」
→「Compiler」→「ソース」画面上にある「プリプロセッサ・マクロの定義」

IAR : プロジェクトのオプションから「C/C++コンパイラ」→「プリプロセッサ」画面上にある「シンボル定義」

4.5.2.3 TAU

タイマ・アレイ・ユニットを使用したタイマ制御を行うドライバです。本サンプルアプリケーションではアプリケーションに則し、各チャンネルの設定を以下の通り行っています。

表 4.13 タイマ・アレイ・ユニット設定

チャンネル	設定	割り込み優先度	用途
TAU0 CH0	インターバル・タイマ(1ms)	割り込み不使用	アプリケーションの 1ms 定期処理同期
TAU0 CH1	入力パルス幅測定	優先度 0	DALI RxD ピンのパルス幅測定
TAU0 CH2	インターバル・タイマ (50us)	優先度 1	DALI RxD ピンの Idle/Active ステート継続 時間計測
TAU0 CH3	インターバル・タイマ (1458us)	優先度 3	Corrupted Backward Frame の Active State 送信時間
TAU0 CH4	インターバル・タイマ(任意)	優先度 3	Collision 発生時の処理用 - tBreak の Active State 送信時間 (1200-1400us) - DALI TxD ピンから送信した波形を DALI RxD ピンに受信するまでの ループバック時間(50ms)
TAU0 CH5	未使用	-	-
TAU0 CH6	未使用	-	-
TAU0 CH7	未使用	-	-

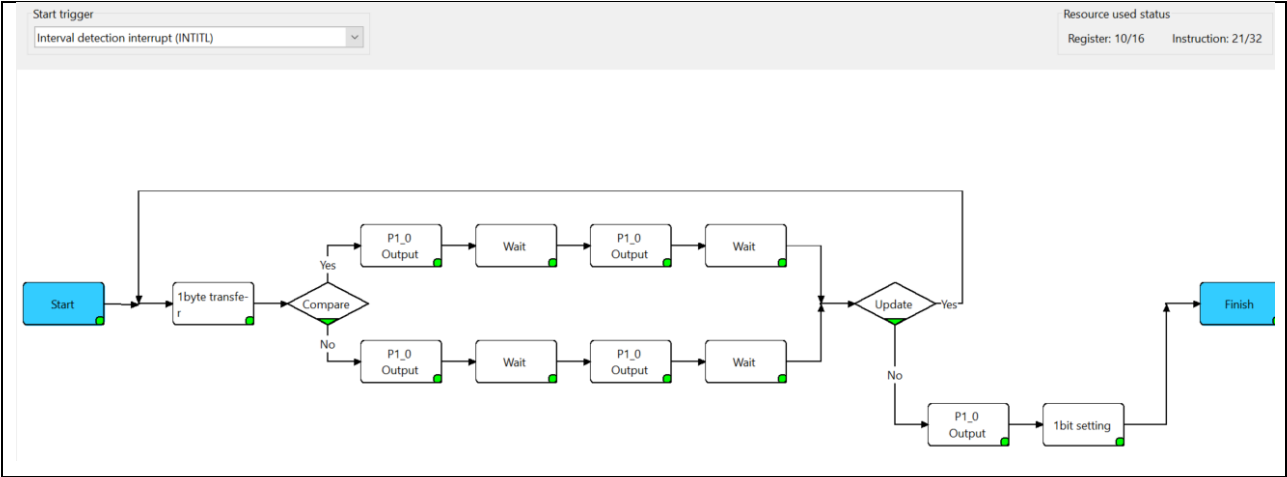
4.5.2.4 TRNG

真性乱数発生器を使用するドライバです。

4.5.2.5 SMS

SNOOSE モード・シーケンサ（以降、SMS）を使用するドライバです。本サンプルアプリケーションでは、SMS を DALI 送信の変調に使用します。スマートコンフィギュレータ（SC）を使用し、以下の設定を行っています。

図 4-11 SMS 設定



上記の設定で実行する SMS 処理の仕様を以下に示します。

表 4.14 SMS 設定

概要	SMS による DALI 波形出力処理
説明	<p>① INTITL 割り込みにより SMS が起動します。</p> <p>② 引数から 1bit データずつ判断しデータにあった波形の送信を行います。</p> <ul style="list-style-type: none">・ bit データ == 0 : P10 (DALI TxD ピン)に 1→0 の順にて Wait 時間幅で出力・ bit データ == 1 : P10 (DALI TxD ピン)に 0→1 の順にて Wait 時間幅で出力 <p>③ すべてのデータの送信が完了したら P10 ポートの出力を High に設定し、MK0H の bit4(SMSEMK)を 0 に変更し SMS 終了割り込み（INTSMSE）を発生させます。</p> <p>④ Finish 命令を実行して SMS の処理を終了します。。</p>
引数 ^{注 1}	address_d : 送信データ先頭アドレス address_e : 送信データ終端アドレス
戻り値	なし
備考	<p>DALI TxD ピンを基準に出力される波形のビット幅はマイコン外部の DALI 通信回路における立上りまたは立下り遅延で変化します。DALI 通信バスに出力される波形が規格に適合するように調整が必要です。</p> <p>詳細は RL78/G23 DALI-2 Input Device 基本(103)サンプルアプリケーションノートの 4.5.6.2 章を参照してください。</p>

4.5.2.6 TML32

32 ビット・インターバル・タイマを使用するドライバです。本サンプルアプリケーションではアプリケーションに則し、4ch の 8 ビット・カウンタ・モードとして各チャンネルの設定を以下の通り行っています。

表 4.15 32 ビット・インターバル・タイマ設定

チャンネル	設定	割り込み優先度	用途
TML32 CH0	インターバル・タイマ (16us)	割り込み不使用	SMS の起動トリガ
TML32 CH1	未使用		-
TML32 CH2	未使用		-
TML32 CH3	未使用		-

4.5.2.7 SAU

シリアル・アレイ・ユニットを使用するドライバです。

4.5.2.8 WDT

ウォッチドッグ・タイマを使用するドライバです。このドライバではウォッチドッグ・タイマのリスタートを行います。

ウォッチドッグ・タイマの設定に関しては 4.6.2 オプション・バイトの設定一覧を参照してください。

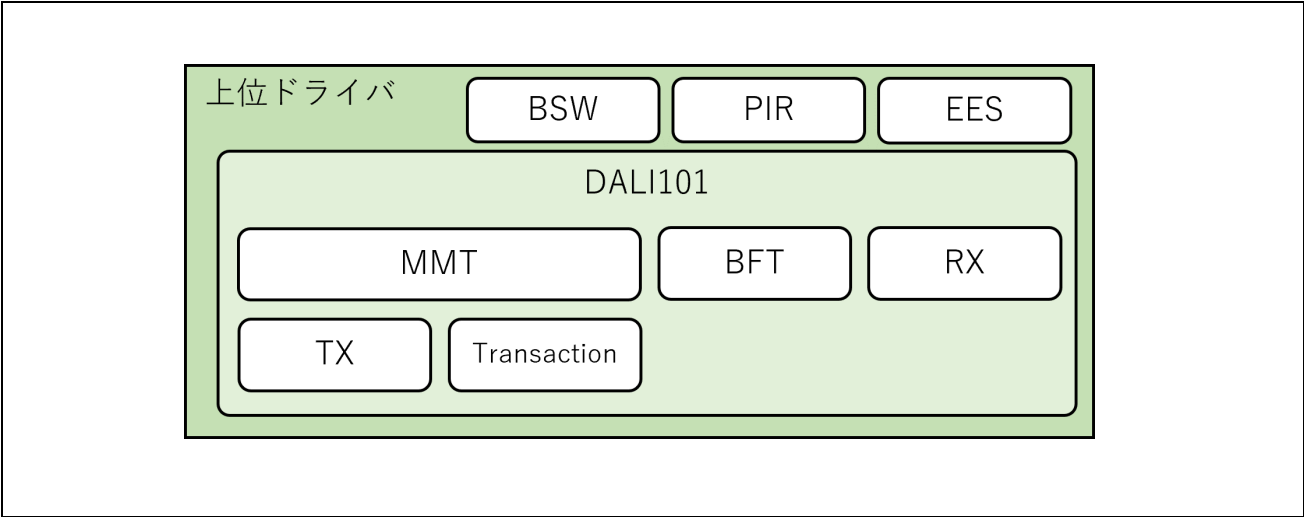
4.5.2.9 RFD

RL78/G23 のデータ・フラッシュに対して読み書きを行うドライバです。本サンプルアプリケーションではルネサスエレクトロニクス製 Renesas Flash Driver を使用しています。

詳細は RL78/G23 用 Renesas Flash Driver RL78 Type01 ユーザーズマニュアル (R20UT4830) を参照してください。

4.5.3 上位ドライバ層

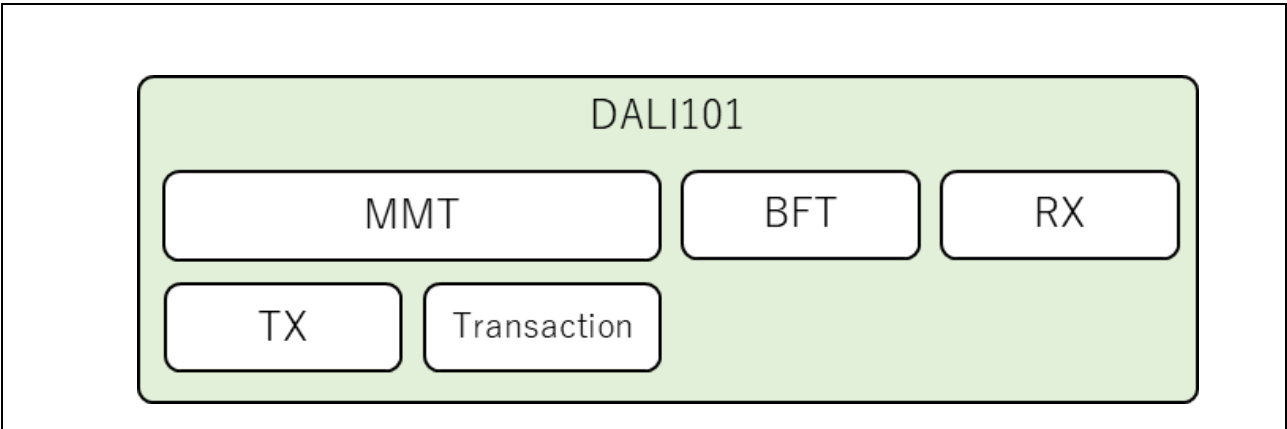
下位ドライバを組み合わせて実現する高機能なドライバです。



4.5.3.1 DALI101 通信ドライバ

本サンプルアプリケーションの DALI 通信ドライバは以下の構成となっています。

図 4-12 DALI101 通信ドライバ構成



(1) DALI101

DALI101 通信ドライバの API 関数群です。

本モジュール内では、DALI RxD ピンの状態（RxD レベル継続時間、受信ビット幅）を取得し、後述の MMT と RX へ渡す機能を実装します。

- DALI RxD レベル継続時間測定

DALI RxD ピンのエッジ検出をトリガに 50us インターバル・タイマ（タイマ・アレイ・ユニット 0 チャンネル 2）をスタート／リスタートします。同タイマの割り込みハンドラで継続時間に 100us を加えることで RxD レベル(High/Low)の継続時間を測定します。

- DALI RxD 受信ビット幅測定

DALI RxD ピンを対象にタイマ・アレイ・ユニットの入力パルス間隔測定機能を使用して受信ビット幅を測定します。

ただし、各計測時間はマイコン端子基準での時間となるためマイコン外部の DALI 通信回路における立上りまたは立下り遅延で変化するため、取得時間に誤差が生じます。そのため、DALI 通信回路の遅延を考慮した補正を行った上で、MMT 及び RX へ渡しています。

補正や閾値に関しては 4.5.6 閾値とキャリブレーションを参照してください。

その他、MMT、BFT、RX の関数群をラップする関数を実装しています。

(2) Multi-Master Transmitter (MMT)

DALI の Multi-Master Transmitter 規定を実現するモジュールです。

ユーザが登録した DALI Frame を priority に対応した適切な settling time で送信を開始します。また、送信中のループバック受信データをチェックし、collision が発生していないかを確認します。

collision 発生時には、送信を中止し適切な復帰処理を行います。

(3) Backward Frame Transmitter (BFT)

Backward Frame を送信するためのモジュールです。

8bit の有効な Backward Frame を送信する場合は後述の Transmitter (TX)を使用して送信し、Corrupted Backward Frame を送信する場合は、DALI TxD ピンを 1.458ms 間 Low 出力にしたあと High 出力にすることで実現をします。

なお、Corrupted Backward Frame とは複数の Backward Frame がぶつかることによって発生する破損した Backward Frame を指しており、DALI 機器内で複数の論理的な DALI デバイスがいた際に各デバイスで Backward Frame のデータ内容が異なる場合にそれぞれの Backward Frame を集約して Corrupted Backward Frame として送信されます。

(4) Receiver (RX)

DALI の Receiver 規定を実現するモジュールです。

上位ドライバ DALI101 から受け取る DALI RxD 受信ビット幅／ビットレベル(High/Low)は順次リングバッファに格納し、DALI RxD レベル継続時間で Stop Condition 及び System Failure の検出を行います。

Stop Condition 検出時に Frame 受信完了とみなし、それまでリングバッファに格納していた各ビット幅／ビットレベル(High/Low)から DALI 規格を満たしたフレームか否かをチェックします。

正常なフレームであった場合、直前 100ms 以内に同フレームを受信したか否かの情報も含めて提供します。

(5) Transaction (TRANSACTION)

Transaction 仕様（一連のフレームで意味を成すフレーム群）を含めた Frame 送信用バッファモジュールです。

(6) Transmitter (TX)

SMS と TML32 を使用して 32bit までの Frame を送信するモジュールです。

4.5.3.2 EES

RL78/G23 のデータ・フラッシュを使用して EEPROM エミュレーション機能を実行するためのソフトウェアです。本サンプルアプリケーションではルネサスエレクトロニクス製 EEPROM エミュレーション・ソフトウェアを使用しています。

詳細は RL78/G23 用 EEPROM エミュレーション・ソフトウェア RL78 Type01 ユーザーズマニュアル (R20UT5008) を参照してください。

4.5.3.3 BSW

Binary Switch(略称 : BSW)の入力状態を取得するドライバです。Binary Switch を接続しているピンを指定することで bouncing(設定の状態が変化する際、一時的に ON/OFF を繰り返す不安定な減少)対策を含めた入力状態を取得することができます。

4.5.3.4 PIR

PmodPIR の検知信号を取得するドライバです。センサの接続先となる PORT(下位ドライバ)を指定することで、センサから出力される端子レベルを監視し検知状態を取得することが出来ます。

4.5.4 ライブラリ層

DALI 通信における Input Device 用ライブラリとして、DALI103i ライブラリ及び DALI303 ライブラリを使用します。

4.5.5 アプリケーション層

4.5.5.1 Input Device 実体確保

Input Device の仕様に合わせて Instance や logical unit の実体を確保してください。本サンプルアプリケーションでは 1 logical unit で必要な要素を logical_unit_t 構造体にまとめてから実体を確保しています。

4.5.5.2 Input Device アプリケーション

(1) Forward Frame 受信時処理

受信した Forward Frame の解析、コマンドの実行、Backward Frame を送信します。

(2) logical unit 処理

本サンプルアプリケーションは単一の logical unit となっています。32 個を超える Instance が必要な場合は複数の logical unit を実装できます。

(3) Instance 処理

信号処理装置の処理を行います。使用する信号処理装置に応じて処理を実装してください。

本サンプルアプリケーションでは Occupancy Sensor の信号処理装置として Pmod PIR(Movement Sensor: 人感センサ)を使用しています。

(4) NVM 処理

データ・フラッシュの保存処理を行います。本サンプルアプリケーションでは EES ライブラリと RFD ライブラリで実装しています。

(a) 自動保存

DALI 規格の NVM 変数は少なくとも電源オフになる 30 秒前の状態を保存しておく必要があるため、本サンプルアプリケーションでは、NVM 変数に変更があった場合は 1 秒ごとに自動的に保存を行います。データ・フラッシュへの書き込み回数を減らしたい場合は保存頻度を伸ばすか、電源オフを検出したときのみ書き込むようにしてください。

(b) マニュアル保存

明示的に NVM 変数を保存したい場合、Application controller から"SAVE PERSISTENT VARIABLES"コマンドを送信します。

このコマンドを受信したら 300ms 以内にすべての NVM 変数を保存してください。

(5) Event 処理

(a) POWER NOTIFICATION

電源再投入を通知するためのイベントです。

Input Device への電源供給開始から 1.3s~5s のランダムなタイミングで送信してください。

(b) INPUT NOTIFICATION

信号処理装置の状態に応じて Event を発行することができます。Instance Type 1 から 31 を実装する場合は対応する規格(301 から 331)に従って実装、Instance Type 0 で実装する場合はユーザが仕様を定義してください。

(6) 乱数生成処理

DALI103i ライブラリで使用する乱数を生成します。生成する乱数の条件は下記の通りです。

- ・ [0x000000, 0xFFFFFE]の範囲で生成されること。
- ・ 少なくとも logical unit の数だけ生成を繰り返しても同じ値が生成されないこと。

※疑似乱数で実装する場合はシード値を真性乱数としてください。

4.5.6 閾値とキャリブレーション

本章の内容は、RL78/G23 DALI-2 Input Device 基本(103)サンプルアプリケーションノートの同章を参照してください。

4.6 ソフトウェア構成

サンプルアプリケーションのソフトウェア構成を記載します。

4.6.1 フォルダ構成

本サンプルアプリケーションのフォルダ構成を以下に示します。

表 4.16 フォルダ構成 (1/2)

フォルダ、ファイル名	説明
DALI103i_303_sample	サンプルアプリケーションのフォルダ
\App	アプリケーションプログラム格納フォルダ
r_common.h	共通定義ヘッダファイル
r_input_device.c	Input Device モジュールのソースファイル
r_input_device.h	Input Device モジュールのヘッダファイル
r_it3_instance.c	Instance Type 3 のソースファイル
r_it3_instance.h	Instance Type 3 のヘッダファイル
r_main.c	メインソースファイル
r_memory_bank.c	メモリバンクモジュールのソースファイル
r_memory_bank.h	メモリバンクモジュールのヘッダファイル
r_memory_banks.c	複数のメモリバンクを管理するモジュールのソースファイル
r_memory_banks.h	複数のメモリバンクを管理するモジュールのヘッダファイル
r_nvm.c	データ・フラッシュ・ライブラリ対応 NVM モジュールのソースファイル
r_nvm.h	データ・フラッシュ・ライブラリ対応 NVM モジュールのヘッダファイル
r_random.c	乱数生成モジュールのソースファイル
r_random.h	乱数生成モジュールのヘッダファイル
r_unit0_memory_bank.c	メモリバンクの unit0 定義用ソースファイル
r_unit0_memory_bank.h	メモリバンクの unit0 定義用ヘッダファイル
\Driver	ドライバプログラム格納フォルダ
r_bsw.c	Binary Switch ドライバのソースファイル
r_bsw.h	Binary Switch ドライバのヘッダファイル
r_dali101.c	DALI101ed.2.0 対応ドライバのソースファイル
r_dali101.h	DALI101ed.2.0 対応ドライバのヘッダファイル
r_dali101_bft.c	DALI101ed.2.0 対応ドライバ (Backward Frame 送信)のソースファイル
r_dali101_bft.h	DALI101ed.2.0 対応ドライバ (Backward Frame 送信)のヘッダファイル
r_dali101_common.h	DALI101ed.2.0 共通定義ヘッダファイル
r_dali101_mmt.c	DALI101ed.2.0 対応ドライバ(マルチマスター送信)のソースファイル
r_dali101_mmt.h	DALI101ed.2.0 対応ドライバ(マルチマスター送信)のヘッダファイル
r_dali101_rx.c	DALI101ed.2.0 対応ドライバ (受信)のソースファイル
r_dali101_rx.h	DALI101ed.2.0 対応ドライバ (受信)のヘッダファイル
r_dali101_transaction.c	DALI101ed.2.0 対応ドライバ(トランザクション)のソースファイル
r_dali101_transaction.h	DALI101ed.2.0 対応ドライバ(トランザクション)のヘッダファイル
r_dali101_tx.c	DALI101ed.2.0 対応ドライバ(送信)のソースファイル
r_dali101_tx.h	DALI101ed.2.0 対応ドライバ(送信)のソースファイル
r_pmod_pir.c	PmodPIR ドライバのソースファイル
r_pmod_pir.h	PmodPIR ドライバのヘッダファイル
r_port.c	PORT 用ソースファイル
r_port.h	PORT 用ヘッダファイル
r_sau.c	SAU 用ソースファイル
r_sau.h	SAU 用ヘッダファイル
r_tau.c	TAU 用ソースファイル
r_tau.h	TAU 用ヘッダファイル

表 4.17 フォルダ構成 (2/2)

フォルダ、ファイル名	説明
DALI103i_303_sample	サンプルアプリケーションのフォルダ
\Driver	ドライバプログラム格納フォルダ
r_tml32.c	TML32 用ソースファイル
r_tml32.h	TML32 用ヘッダファイル
r_trng.c	TRNG ソースファイル
r_trng.h	TRNG 用ヘッダファイル
r_wdt.c	WDT 用ソースファイル
r_wdt.h	WDT 用ヘッダファイル
\HardwareDebug	プロジェクト出力ファイル格納フォルダ
DALI103i_303_sample.mot	ファームウェアへの書き込みに使用するモトローラ・S タイプ・フォーマット形式のファイル
\Library	ライブラリ格納フォルダ
\DALI103i	DALI103ed1.0 対応ライブラリ格納フォルダ
\DALI303	DALI303ed1.0 対応ライブラリ格納フォルダ
\EES	EES 用ライブラリ格納フォルダ
\RFD	RFD 用ライブラリ格納フォルダ
\src	スマート・コンフィグレータ生成ファイル格納フォルダ
smc_gen	スマート・コンフィグレータ生成フォルダ
Utility	共通プログラム格納フォルダ
r_timer16.c	タイマモジュールのソースファイル
r_timer16.h	タイマモジュールのヘッダファイル
r_usertype.h	タイプ定義ヘッダファイル

4.6.2 オプション・バイトの設定一覧

サンプルアプリケーションのオプション・バイト設定を以下に示します。

表 4.18 オプション・バイト設定一覧

アドレス	設定値	内容
000C0H / 040C0H	11101111B	ウォッチドッグ・タイマ動作許可 ・リセット解除後、カウント停止 ・オーバーフロー時間 : $2^{17}/f_{IL}$ (3478.26 ms)
000C1H / 040C1H	11111101B	LVD リセット・モード (立ち下がり : 2.91V 立ち上がり : 2.97V)
000C2H / 040C2H	11101000B	HS モード、高速オンチップ・オシレータ : 32MHz
000C3H / 040C3H	10000100B	オンチップ・デバッグ許可

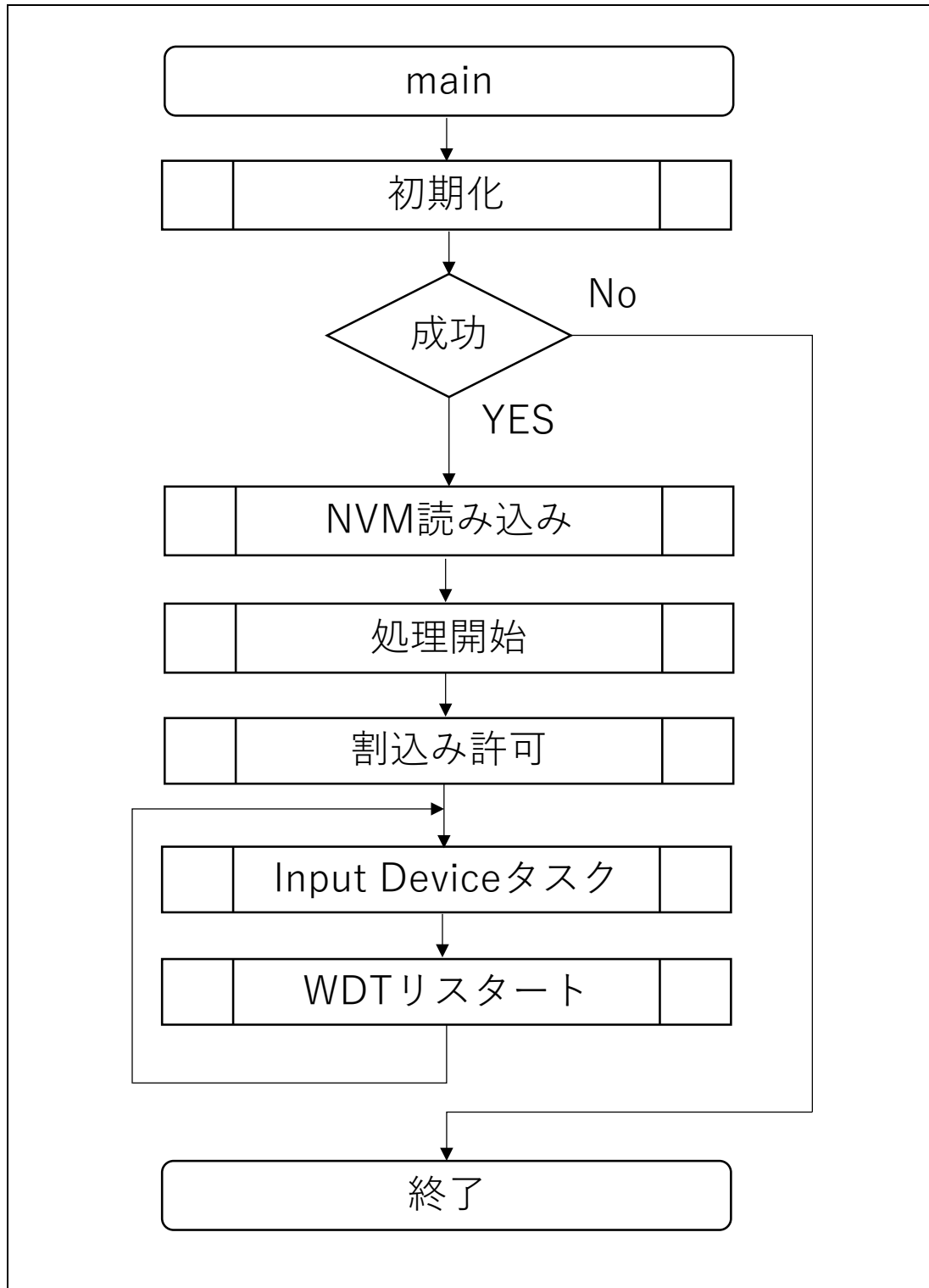
4.6.3 フローチャート

4.6.3.1 メインフロー

サンプルアプリケーションのメインフローを以下に示します。

初期化処理終了後、Input Device 処理を実行します。

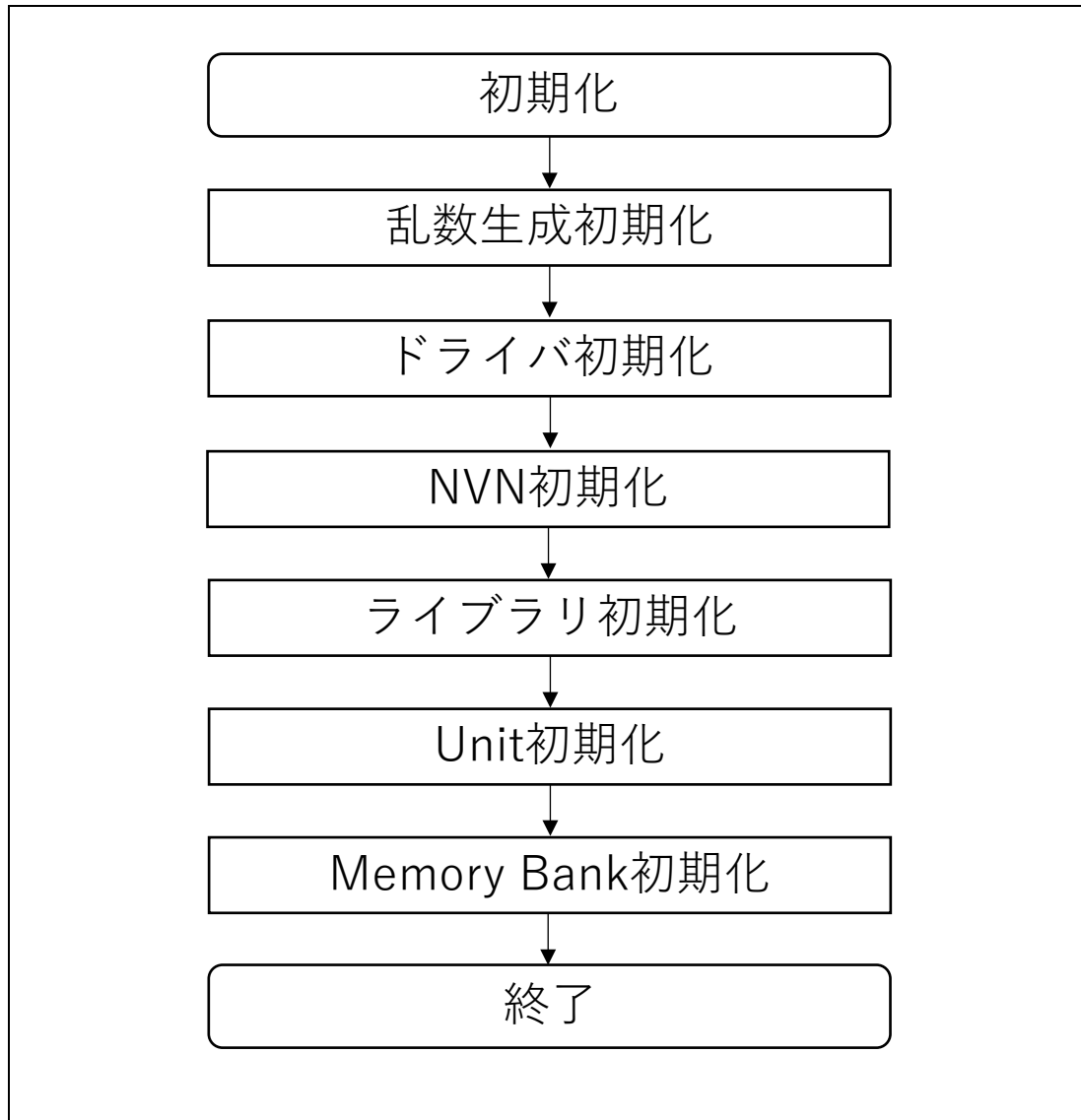
図 4-13 メインフロー図



4.6.3.2 初期化

初期化処理のフローを以下に示します。

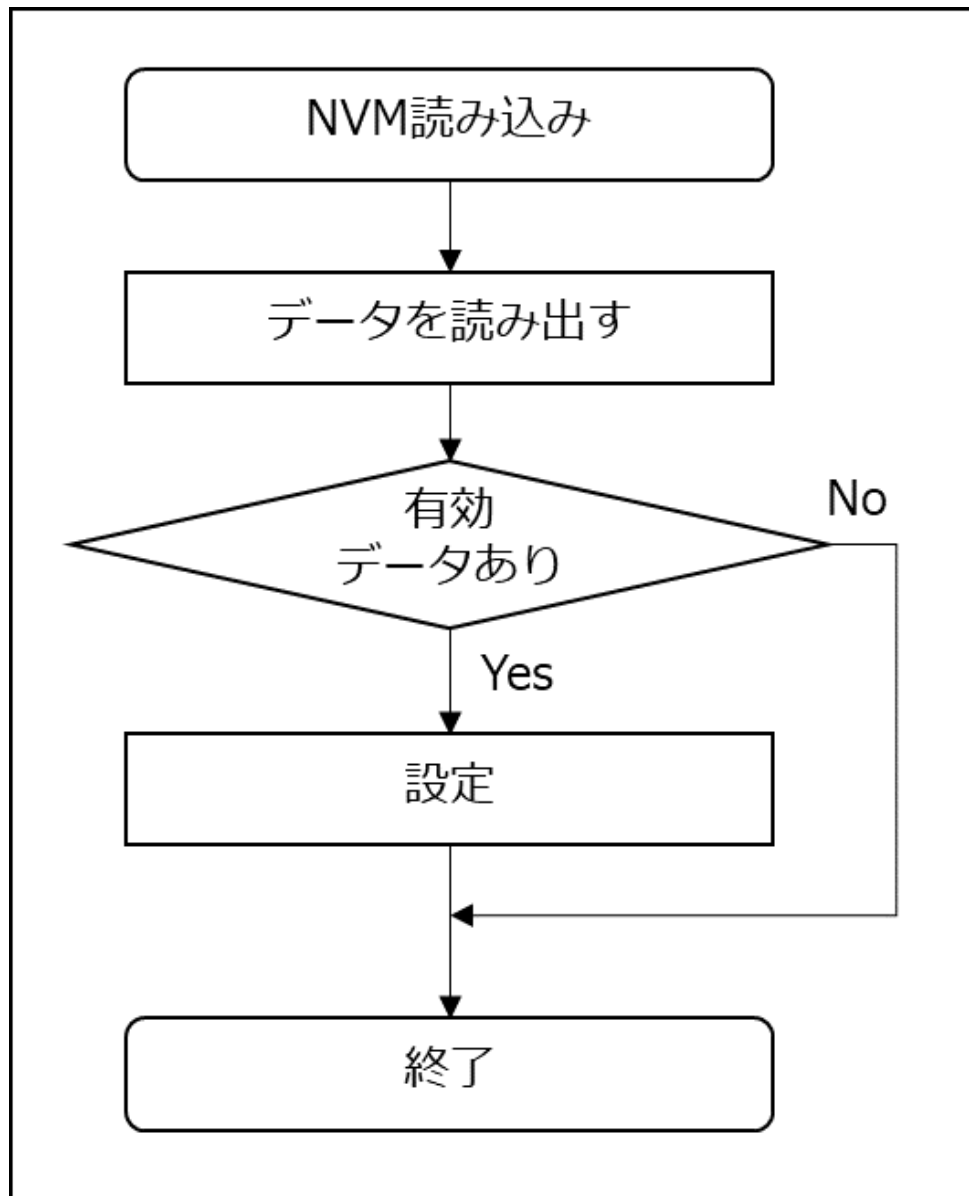
図 4-14 初期化処理フロー図



4.6.3.3 NVM 読み込み

初期化処理後に実施する NVM 読み込み処理のフローを以下に示します。

図 4-15 NVM 読み込み処理フロー図



4.6.3.4 処理開始

NVM 読み込み処理後に実施する処理開始のフローを以下に示します。

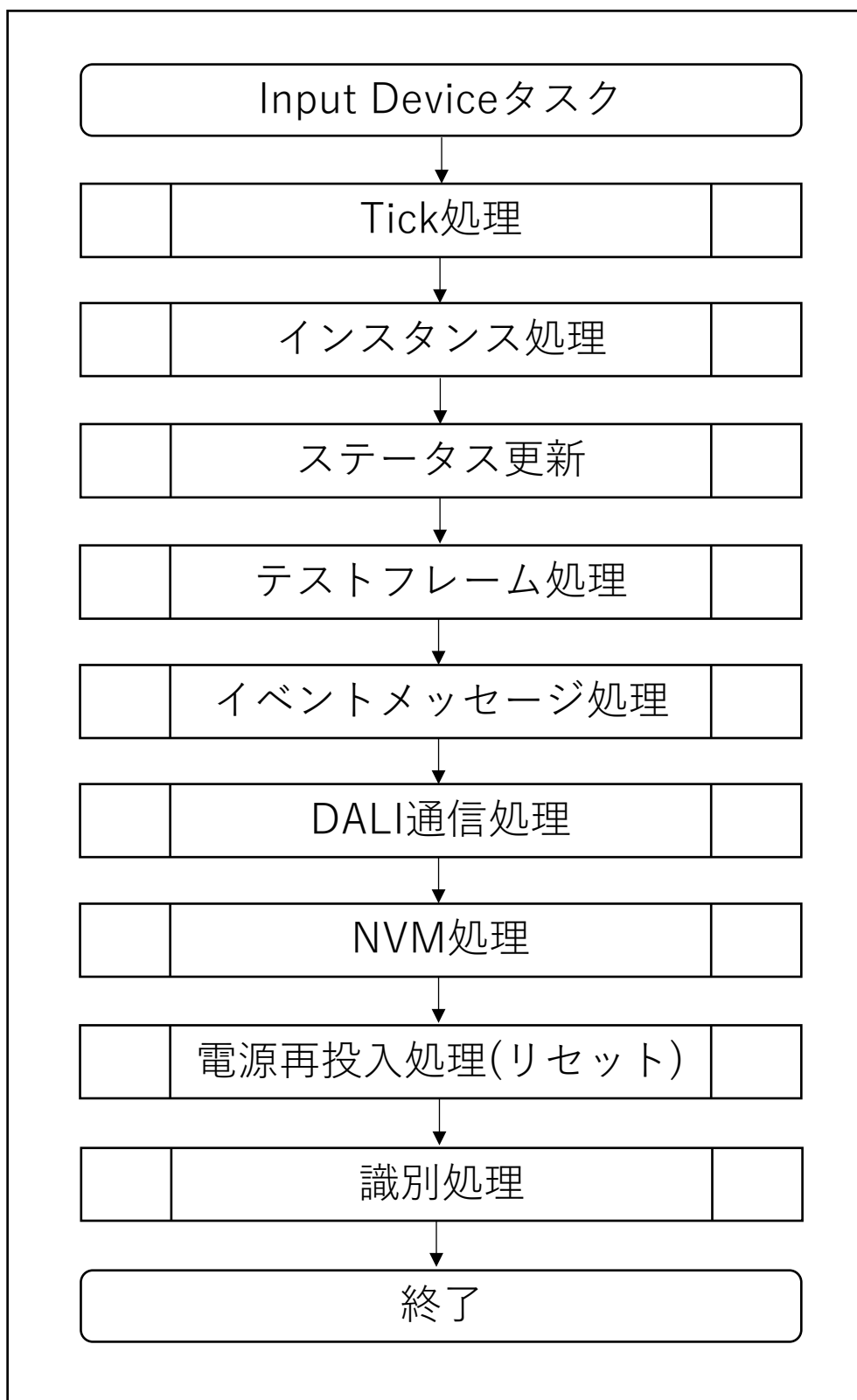
図 4-16 処理開始フロー図



4.6.3.5 Input Device タスク処理

Input Device タスク処理のフローを以下に示します。

図 4-17 Input Device タスク処理フロー図



5. 注意事項

- 本サンプルアプリケーションはボード RL78/G23 Lighting Communication Master Evaluation Board に合わせた構成・動作となっています。
- 本サンプルアプリケーションが提供する動作はあくまでもサンプルです。アプリケーション層は、使用目的に合わせた設計変更・評価を行ってください。

6. 参考ドキュメント

RL78/G23 ユーザーズマニュアル ハードウェア編 (R01UH0896JJ0110)

RL78 ファミリ ユーザーズマニュアル ソフトウェア編 (R01US0015J)

統合開発環境 e2 studio 2020-04、e2 studio v7.8 ユーザーズマニュアル 入門ガイド (R20UT4819)

DALI マスタコントローラ GUI ユーザーズマニュアル (R20UT0715)

EZ-0012 RL78/I1A DC/DC LED 制御評価ボード ユーザーズマニュアル (R01UH0363)

RL78/I1A DALI-2 Control Gear 基本(102) 調光(207) 調色(209Tc) サンプルアプリケーション(R01AN6177)

(最新版をルネサスエレクトロニクスホームページから入手してください。)

テクニカルアップデート／テクニカルニュース

(最新版の情報をルネサスエレクトロニクスホームページから入手してください。)

DALI 規格書

IEC 62386-101:2014+AMD1:2018 CSV Consolidated version (ed.2.1)

IEC 62386-103:2014+AMD1:2018 CSV Consolidated version (ed.1.1)

IEC 62386-303:2017 (ed.1.0)

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2023/11/08		
1.01	2025/11/28	31	SMS 設定

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うと、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

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。回路、ソフトウェアおよびこれらに関連する情報を使用する場合、お客様の責任において、お客様の機器・システムを設計ください。これらの使用に起因して生じた損害（お客様または第三者いずれに生じた損害も含みます。以下同じです。）に関し、当社は、一切その責任を負いません。
2. 当社製品または本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
3. 当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
4. 当社製品を組み込んだ製品の輸出入、製造、販売、利用、配布その他の行為を行うにあたり、第三者保有の技術の利用に関するライセンスが必要となる場合、当該ライセンス取得の判断および取得はお客様の責任において行ってください。
5. 当社製品を、全部または一部を問わず、改造、改変、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、改変、複製、リバースエンジニアリング等により生じた損害に関し、当社は、一切その責任を負いません。
6. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。

標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット等

高品質水準： 輸送機器（自動車、電車、船舶等）、交通管制（信号）、大規模通信機器、金融端末基幹システム、各種安全制御装置等

当社製品は、データシート等により高信頼性、Harsh environment 向け製品と定義しているものを除き、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じても、当社は一切その責任を負いません。

7. あらゆる半導体製品は、外部攻撃からの安全性を 100%保証されているわけではありません。当社ハードウェア／ソフトウェア製品にはセキュリティ対策が組み込まれているものもありますが、これによって、当社は、セキュリティ脆弱性または侵害（当社製品または当社製品が使用されているシステムに対する不正アクセス・不正使用を含みますが、これに限られません。）から生じる責任を負うものではありません。当社は、当社製品または当社製品が使用されたあらゆるシステムが、不正な改変、攻撃、ウイルス、干渉、ハッキング、データの破壊または窃盗その他の不正な侵入行為（「脆弱性問題」といいます。）によって影響を受けないことを保証しません。当社は、脆弱性問題に起因したまたはこれに関連して生じた損害について、一切責任を負いません。また、法令において認められる限りにおいて、本資料および当社ハードウェア／ソフトウェア製品について、商品性および特定目的との合致に関する保証ならびに第三者の権利を侵害しないことの保証を含め、明示または黙示のいかなる保証も行いません。
8. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は、データシート等において高信頼性、Harsh environment 向け製品と定義しているものを除き、耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
10. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。かかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
11. 当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
12. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものいたします。
13. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。
14. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。

注 1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社が直接的、間接的に支配する会社をいいます。

注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

(Rev.5.0-1 2020.10)

本社所在地

〒135-0061 東京都江東区豊洲 3-2-24（豊洲フォレシア）

www.renesas.com

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/