

## RL78/G14 グループ

R01AN3340JJ0100

Rev.1.00

## SPI バス加速度センサ用制御ソフトウェア

2016.08.31

### 要旨

本アプリケーションノートでは、ルネサス エレクトロニクス製 MCU を使用した SPI バス加速度センサ用制御方法と制御ソフトウェアの使用方法を説明します。

本制御ソフトウェアは、加速度センサの各レジスタにアクセスし、システムに必要な制御が可能です。

本制御ソフトウェアでは、スレーブデバイスを制御するためのソフトウェアを上位層、クロック同期式シングルマスタ基本プロトコル制御を実現するソフトウェアを下位層とし扱います。上位層は下位層で用意されているプロトコルを組み合わせることでスレーブデバイスを制御します。

本制御ソフトウェアは、上位層の加速度センサを制御するソフトウェアと下位層のクロック同期式シングルマスタ制御するソフトウェアを提供します。

### 動作確認デバイス

#### 加速度センサ

アナログ・デバイセズ社製 ADXL362 (以下の URL を参照してください)

[http://www.analog.com/jp/adxl362?adacid=AF\\_JP\\_P1108](http://www.analog.com/jp/adxl362?adacid=AF_JP_P1108)

#### 動作確認に使用した MCU

RL78/G14 グループ : RL78/G14 (SAU を使用)

本アプリケーションノートを他の MCU や他メモリへ適用する場合、その MCU やメモリの仕様に合わせて変更し、十分評価してください。

また、本制御ソフトウェアはセンサデバイスへのアクセス制御方法のサンプルコードであり、動作を保証するものではありません。

## 目次

1. 概要.....	3
1.1 機能.....	3
1.2 API の概要とメモリサイズ.....	4
1.2.1 API の概要.....	4
1.2.2 動作環境とメモリサイズ.....	5
1.3 関連アプリケーションノート.....	6
1.3.1 クロック同期式シングルマスタ制御ソフトウェア.....	6
1.3.2 センサノード応用例.....	6
1.4 ハードウェア接続.....	7
1.5 ソフトウェア制御.....	8
1.5.1 概要.....	8
1.5.2 ソフトウェア構成.....	9
1.5.3 SPI モード制御.....	11
1.5.4 スレーブデバイスセレクト制御.....	11
1.5.5 データバッファと送信／受信データの関係.....	12
1.6 DTC 使用の制限.....	12
1.7 ファイル構成.....	13
2. API 情報.....	15
2.1 ハードウェアの要求.....	15
2.2 ソフトウェアの要求.....	15
2.3 サポートされているツールチェーン.....	15
2.4 ヘッドファイル.....	15
2.5 整数型.....	15
2.6 コンパイル時の設定.....	16
2.7 引数.....	21
2.8 戻り値.....	21
3. API 関数.....	22
3.1 R_ACCLNSR_SPI_Open.....	22
3.2 R_ACCLNSR_SPI_Close.....	23
3.3 R_ACCLNSR_SPI_WriteReg.....	24
3.4 R_ACCLNSR_SPI_ReadReg.....	25
3.5 R_ACCLNSR_SPI_ReadFIFO.....	26
3.6 R_ACCLNSR_SPI_GetIntStatus.....	27
3.7 R_ACCLNSR_SPI_StartSampling.....	28
3.8 R_ACCLNSR_SPI_StopSampling.....	29
3.9 R_ACCLNSR_SPI_GetVersion.....	30
4. 注意事項.....	31
4.1 RAM 初期化に関する注意事項.....	31
4.2 加速度センサ制御ソフトウェア共通の注意事項.....	31
4.3 アナログ・デバイsez社製 ADXL362 制御例.....	32

## 1. 概要

### 1.1 機能

ルネサス エレクトロニクス製 MCU を使用した SPI バス加速度センサ用制御ソフトウェアを提供します。

別途、MCU 個別のクロック同期式シングルマスタ制御ソフトウェアが必要です。

表 1-1 に使用する周辺機器と用途を、図 1-1 に使用例を示します。

以下に、SPI バス加速度センサの制御ソフトウェアの概略を示します。

- SPI マスタデバイスをルネサス エレクトロニクス製 MCU、SPI スレーブデバイスを加速度センサとするデバイスドライバです。
- 加速度センサ内のレジスタ空間へのアクセス API を提供します。
- 最大 2 個の同一型名の加速度センサ制御が可能です。（デバイス数を増やす場合、制御ソフトウェアの修正が必要）

さらに、加速度センサ個別の機能を以下に示します。

#### (1) アナログ・デバイゼズ社製 ADXL362 制御時

- SPI モード 0 で制御し、レジスタ読み出し／レジスタ書き込み／FIFO 読み出しの 3 コマンドプロトコルをサポートします。
- 割り込みステータス取得関数をサポートします。
- 測定開始／停止関数をサポートします。
- INT1 ピンを FIFO トリガレベル割り込み要求端子として使用します。
- INT2 ピンを同期サンプリングのトリガとして使用します。

表 1-1 使用する周辺機器と用途

周辺機器	用途
MCU 内蔵のシリアル通信 (クロック同期式モード)	シリアル通信機能（クロック同期式モード）による SPI スレーブデバイスとの通信：1 もしくは複数チャンネル（必須）
Port	スレーブデバイスセレクト制御信号用：使用デバイス数分のポートが必要（必須）
割り込み	スレーブデバイスからの割り込み検出（オプション）
タイマ出力	タイマ出力（オプション）

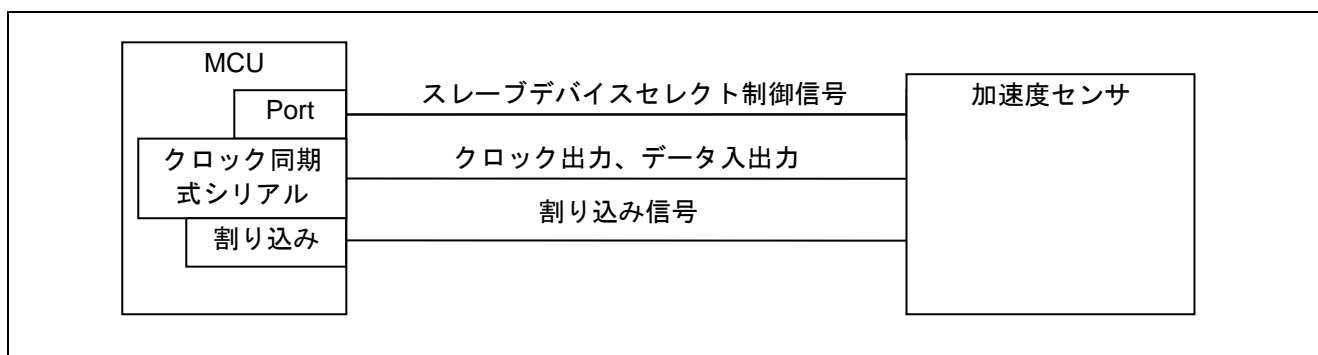


図 1-1 使用例

## 1.2 API の概要とメモリサイズ

### 1.2.1 API の概要

表 1-2に加速度センサ制御ソフトウェアに含まれる API 関数を示します。

表 1-2 API 関数

関数名	説明
R_ACCLSNSR_SPI_Open()	本制御ソフトウェアの初期化処理
R_ACCLSNSR_SPI_Close()	本制御ソフトウェアの終了処理
R_ACCLSNSR_SPI_WriteReg()	レジスタ書き込み処理
R_ACCLSNSR_SPI_ReadReg()	レジスタ読み出し処理
R_ACCLSNSR_SPI_ReadFIFO()	FIFO 読み出し処理
R_ACCLSNSR_SPI_GetIntStatus()	割り込みステータス取得処理
R_ACCLSNSR_SPI_StartSampling()	サンプリング開始処理
R_ACCLSNSR_SPI_StopSampling()	サンプリング停止処理
R_ACCLSNSR_SPI_GetVersion()	本制御ソフトウェアのバージョン情報取得処理

## 1.2.2 動作環境とメモリサイズ

## (1) RL78/G14 の場合

表 1-3に動作確認条件、表 1-4に加速度センサ制御ソフトウェアに必要なメモリサイズを示します。

メモリサイズは「2.6 コンパイル時の設定」のデフォルト設定を選択した場合の値です。選択する定義により、メモリサイズは異なります。

表 1-3 動作確認条件

項目	内容
使用 MCU	RL78/G14 グループ (プログラム ROM 512KB/RAM 48KB)
動作周波数	メイン・システム・クロック : 32MHz 周辺ハードウェア・クロック : 32MHz クロック同期シリアル・クロック : 4MHz
動作電圧	3.3V
統合開発環境	ルネサス エレクトロニクス製 CS+ for CC V4.00.00
C コンパイラ	ルネサス エレクトロニクス製 RL78 ファミリ用 C コンパイラパッケージ V1.02.00 コンパイルオプション : 統合開発環境のデフォルト設定を使用しています。
ソフトウェアのバージョン	Ver.1.00
使用ボード	サイレックス・テクノロジー社製 SX-ULPAN-DVK-S01

表 1-4 必要メモリサイズ

使用メモリ	サイズ	備考
ROM	1,551 バイト (加速度センサ用制御ソフトウェア) 621 バイト (クロック同期式シングルマスタ制御ソフトウェアと共通関数)	r_acclsnsr_spi.c r_acclsnsr_spi_type.c r_acclsnsr_spi_type_sub.c r_acclsnsr_spi_drvif_gpio.c r_acclsnsr_spi_drvif_gpio_dev0.c r_acclsnsr_spi_drvif_int.c r_acclsnsr_spi_drvif_int_dev0.c
RAM	9 バイト (加速度センサ用制御ソフトウェア) 0 バイト (クロック同期式シングルマスタ制御ソフトウェアと共通関数)	r_acclsnsr_spi_drvif_serial.c r_acclsnsr_spi_drvif_serial_dev0.c r_acclsnsr_spi_drvif_timer.c r_acclsnsr_spi_drvif_timer_dev0.c R_SIO2_csi.c 上記の動作確認条件による
最大使用ユーザスタック	122 バイト	
最大使用割り込みスタック	4 バイト	

必要メモリサイズは、C コンパイラのバージョンやコンパイルオプションにより異なります。

エンディアンにより、上記のメモリサイズは、異なります。

上記は、クロック同期式シングルマスタ制御ソフトウェアに SAU を使用し、データ転送に Software を使用する場合のメモリサイズです。

上記のクロック同期式シングルマスタ制御ソフトウェアの ROM/RAM サイズは使用ボードに合わせ込んだ場合のメモリサイズです。

最大使用ユーザスタックサイズは、下位層のクロック同期式シングルマスタ制御ソフトウェアのスタックサイズも含まれます。

### 1.3 関連アプリケーションノート

本アプリケーションノートに関連するアプリケーションノートを以下に示します。併せて参照してください。

#### 1.3.1 クロック同期式シングルマスタ制御ソフトウェア

以下のアプリケーションノートのソフトウェアは、SPI モード 3 で制御します。

本制御ソフトウェアに添付したクロック同期式シングルマスタ制御ソフトウェア (SPI モード 0) は、以下のソフトウェアを SPI モード 0 制御に変更したものです。

SPI モード 0 と SPI モード 3 の違いがありますが、その他の条件は同一であるため、クロック同期式シングルマスタ制御ソフトウェア (SPI モード 0) の設定方法は、以下のアプリケーションノートを参照してください。

- RL78/G14, RL78/G1C, RL78/L12, RL78/L13, RL78/L1C グループ シリアル・アレイ・ユニットの CSI モードを使ったクロック同期式シングルマスタ制御ソフトウェア(R01AN1195JJ)

#### 1.3.2 センサノード応用例

- RL78/G14 グループ IoT SDK WLAN センサーノード版ソリューション(R01AN3363JJ)

上記の応用例は、以下の制御ソフトウェアと組み合わせて使用します。

- RL78/G14 グループ SPI バス加速度センサ用制御ソフトウェア(R01AN3150JJ)

### 1.4 ハードウェア接続

MCU とシリアル・インタフェースにより端子名が異なります

なお、高速で動作させた場合を想定し、各信号ラインの回路的マッチングを取るためのダンピング抵抗やコンデンサの付加を検討してください。

(1) アナログ・デバイsez社製 ADXL362 – INT2 ピンを同期サンプリング用トリガとして使用

図 1-2に接続例を示します。

電源のデカップリング、電源 (VS、VDD I/O) の分離、電源のスタートアップ条件を満たすように回路を構成してください。

加速度センサの動作電源電圧を MCU の動作電圧以外の値に変更する場合、別途インタフェースレベル変換回路が必要です。

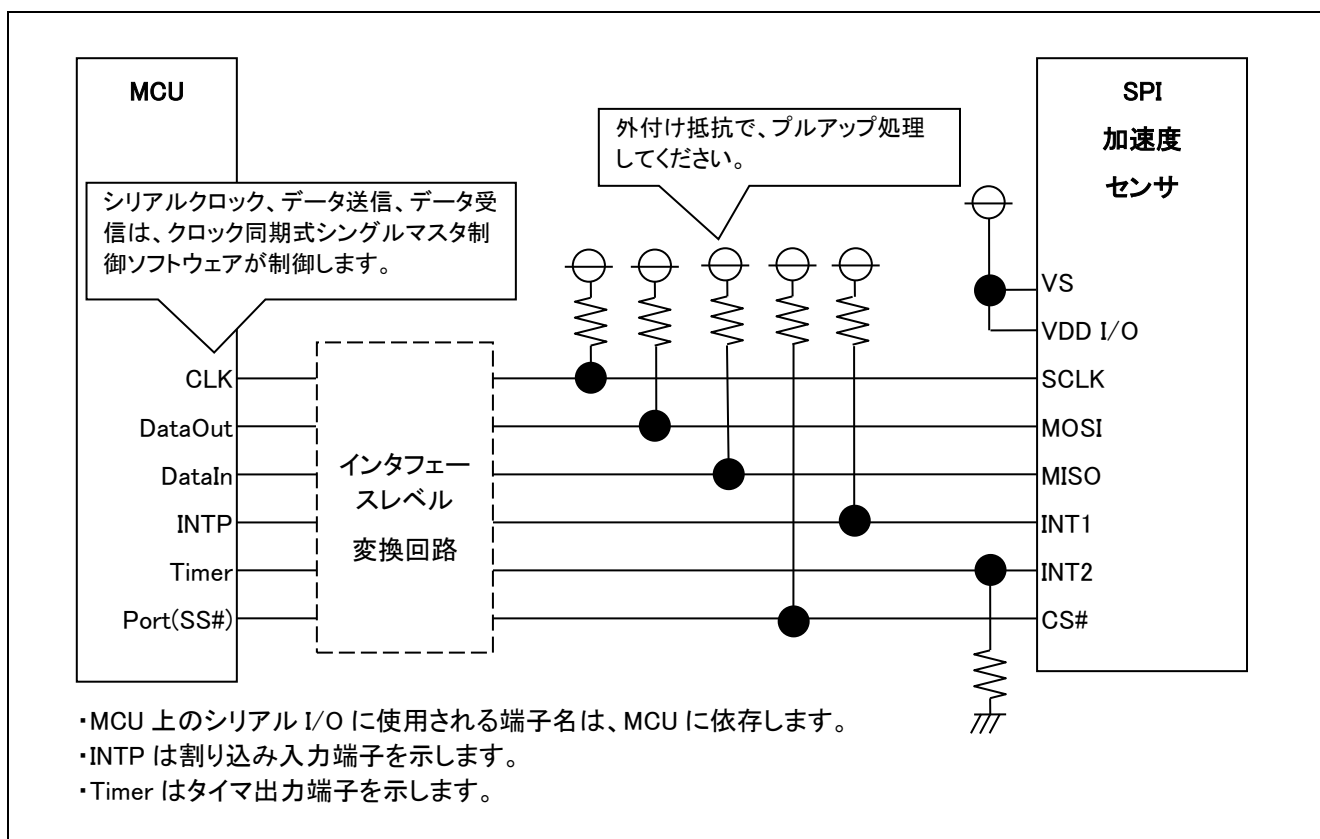


図 1-2 ADXL362 との接続例 – INT2 ピンを同期サンプリング用トリガとして使用

表 1-5 使用端子と機能

端子名	入出力	内容
CLK	出力	クロック出力
DataOut	出力	マスターデータ出力
DataIn	入力	マスターデータ入力
Port(SS#)	出力	スレーブデバイス選択制御信号出力
INTP	入力	割り込み入力
Timer	出力	同期サンプリング用のトリガ出力 (ハイアクティブ)

## 1.5 ソフトウェア制御

### 1.5.1 概要

MCU のクロック同期式シリアル通信機能を使って、内部クロックを使用したクロック同期式シングルマスタ制御を実現します。

なお、使用可能なシリアルクロック周波数は、MCU のユーザーズマニュアル ハードウェア編およびスレーブデバイスのデータシートで、確認してください。

#### (1) アナログ・デバイセズ社製 ADXL362 制御時

アナログ・デバイセズ社製 ADXL362 加速度センサのレジスタ読み出し/レジスタ書き込み/FIFO 読み出しの 3 コマンド制御機能を提供します。これらの 3 コマンドを使って、加速度センサ内のレジスタ空間にアクセスできます。

本制御ソフトウェアでは、SPI 制御用端子以外に、MCU からのタイマ出力機能と加速度センサからの割り込み検出機能を提供します。

表 1-6 に加速度センサ INT1, INT2 の機能割り当てを示します。

データシートを参照し、動作モード設定や各種機能を有効にして、センサデータを取得してください。

表 1-6 ADXL362 の INT1, INT2 の機能割り当て

端子名	用途
INT1	FIFO トリガレベル割り込み（アクティブハイもしくはアクティブロー）として使用 そのため、INTMAP1 レジスタの FIFO_WATERMARK ビットを”1”に設定してください。 FIFO トリガレベルは、FIFO_CONTROL レジスタの AH ビットと FIFO_SAMPLES レジスタを使って設定してください。 FIFO に格納されたサンプル数が FIFO トリガレベル以上になると、STATUS レジスタの FIFO_WATERMARK ビットがセットされ、割り込みを要求します。
INT2	同期トリガ入力（アクティブハイ）として使用 そのため、FILTER_CTL レジスタの EXT_SAMPLE ビットを”1”に設定してください。 同期トリガ信号のタイミング条件については、加速度センサのデータシートを参照してください。

注：レジスタへの書き込みは、R\_ACCLSNSR\_SPI\_WriteReg()関数を使用してください。



## 1.5.2 ソフトウェア構成

## (1) アナログ・デバイセズ社製 ADXL362 制御時

図 1-3にソフトウェア構成を示します。

本制御ソフトウェアを使用して、スレーブデバイスを制御するためのアプリケーションを作成してください。

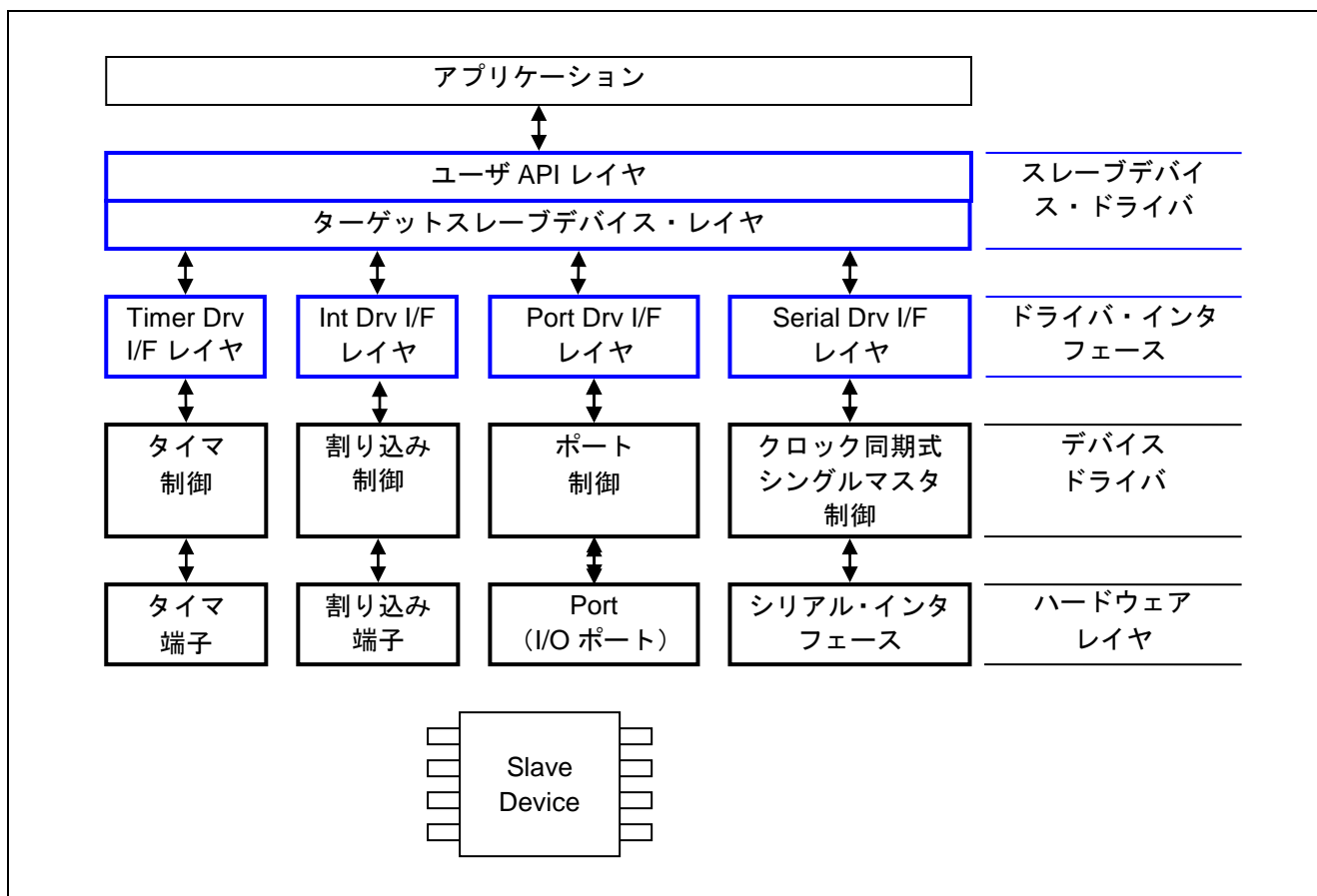


図 1-3 ソフトウェア構成

## (a) アプリケーション

加速度センサの制御例（デモプログラム）を同梱していますので、参照してください。

## (b) ユーザ API レイヤ (r\_acclsnsr\_spi.c)

加速度センサ制御ソフトウェアのユーザインタフェース部です。

## (c) ターゲットスレーブデバイス・レイヤ (r\_acclsnsr\_spi\_type.c)

加速度センサ制御ソフトウェアの加速度センサデバイス依存部分です。

## (d) Serial Drv I/F レイヤ (r\_acclsnsr\_spi\_drvif\_serial.c)

上位層の加速度センサ制御ソフトウェアと下位層のデバイスドライバとの接続部分です。

クロック同期式シングルマスタ制御ソフトウェア毎に合わせ込みが必要です。

**(e) Port Drv I/F レイヤ (r\_acclsnsr\_spi\_drvif\_gpio.c)**

上位層の加速度センサ制御ソフトウェアと下位層のデバイスドライバ（ポート制御ソフトウェア）との接続部分です。

下位層のポート制御ソフトウェア毎に合わせ込みが必要です。ただし、本制御ソフトウェアでは、下位層のポート制御ソフトウェアを使用せず、このレイヤでポート機能を制御するレジスタを直接制御します。

**(f) Int Drv I/F レイヤ (r\_acclsnsr\_spi\_drvif\_int.c)**

上位層の加速度センサ制御ソフトウェアと下位層のデバイスドライバ（割り込み制御ソフトウェア）との接続部分です。

割り込み制御ソフトウェア毎に合わせ込みが必要です。ただし、本制御ソフトウェアでは、下位層の割り込み制御ソフトウェアを使用せず、このレイヤで端子設定と割り込みを制御するレジスタを直接制御します。

**(g) Timer Drv I/F レイヤ (r\_acclsnsr\_spi\_drvif\_timer.c)**

上位層の加速度センサ制御ソフトウェアと下位層のデバイスドライバとの接続部分です。

タイマ制御ソフトウェア毎に合わせ込みが必要です。ただし、本制御ソフトウェアでは、下位層のタイマ制御ソフトウェアを使用せず、このレイヤで端子設定とタイマを制御するレジスタを直接制御します。

加速度センサ制御ソフトウェアでは、タイマは同期サンプリング用のタイマ出力目的で使用します。

**(h) クロック同期式シングルマスタ制御 (R\_SIO2\_csi.c)**

クロック同期式シングルマスタ制御ソフトウェアです。

SPI モード 0 専用のものを同梱済です。

**(i) ポート制御**

ポート制御のデバイスドライバです。

ただし、本制御ソフトウェアでは、ポート制御のデバイスドライバを使用せず、Port Drv I/F レイヤで制御しています。

**(j) 割り込み制御**

割り込み制御のデバイスドライバです。

ただし、本制御ソフトウェアでは、割り込み制御のデバイスドライバを使用せず、Int Drv I/F レイヤで制御しています。

**(k) タイマ制御**

タイマ制御のデバイスドライバです。

ただし、本制御ソフトウェアでは、タイマ制御のデバイスドライバを使用せず、Timer Drv I/F レイヤで制御しています。

### 1.5.3 SPI モード制御

#### (1) アナログ・デバイセズ社製 ADXL362 制御時

図 1-4に示す SPI モード 0 (CPOL=0、CPHA=0) で制御します。

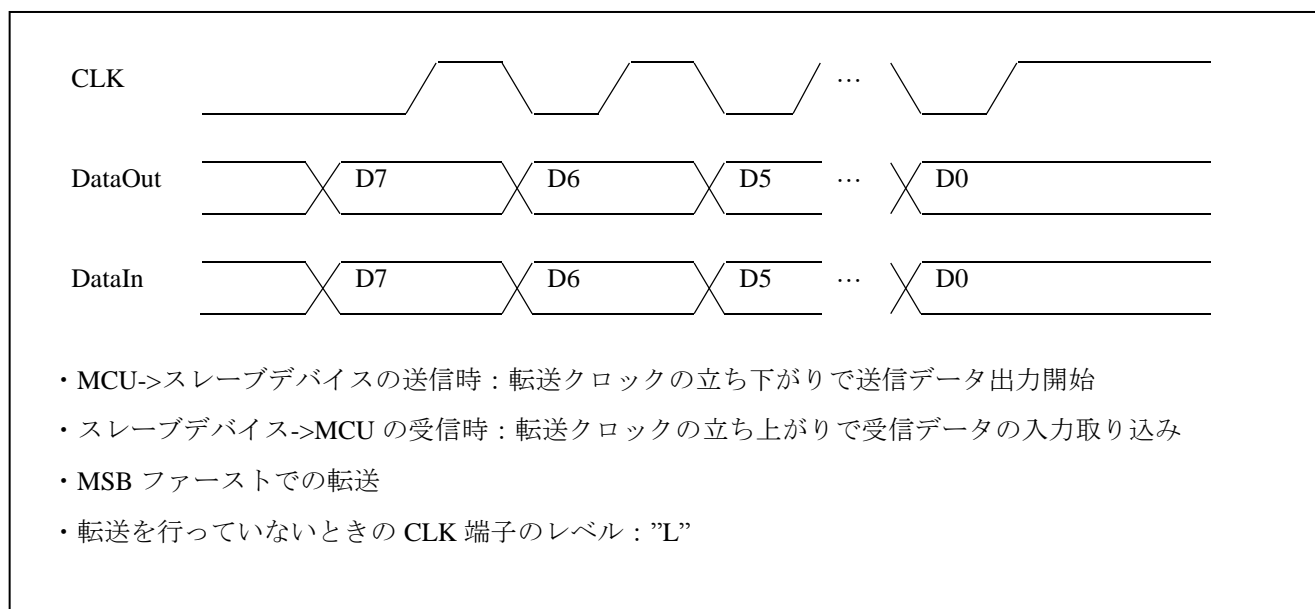


図 1-4 制御可能なスレーブデバイスのタイミング

### 1.5.4 スレーブデバイスセレクト制御

スレーブデバイスセレクト制御信号端子を MCU の Port に接続し、MCU 汎用ポート出力で制御します。

スレーブデバイスセレクト制御 (MCU の Port(SS#)) 信号の立ち下がりから、Clock (MCU の CLK) 信号の立ち下がりまでの時間は、スレーブデバイスセレクト制御のセットアップ時間待ちのために、ソフトウェア・ウェイトで制御しています。

Clock (MCU の CLK) 信号の立ち上がりから、スレーブデバイスセレクト制御 (MCU の Port(SS#)) 信号の立ち上がりまでの時間は、スレーブデバイスセレクト制御のホールド時間待ちのために、ソフトウェア・ウェイトで制御しています。

本制御ソフトウェアでは、スレーブデバイスセレクト制御のセットアップ時間待ち、およびスレーブデバイスセレクト制御のホールド時間待ちを約 1us としています。

### 1.5.5 データバッファと送信／受信データの関係

本制御ソフトウェアは、送信／受信データポインタを引数として設定します。RAM 上のデータバッファのデータ並びと送信／受信順番の関係は、以下のとおりで、エンディアンや使用するシリアル通信機能に関係なく、送信データバッファの並びの順に送信し、また、受信の順に受信データバッファに書き込みます。

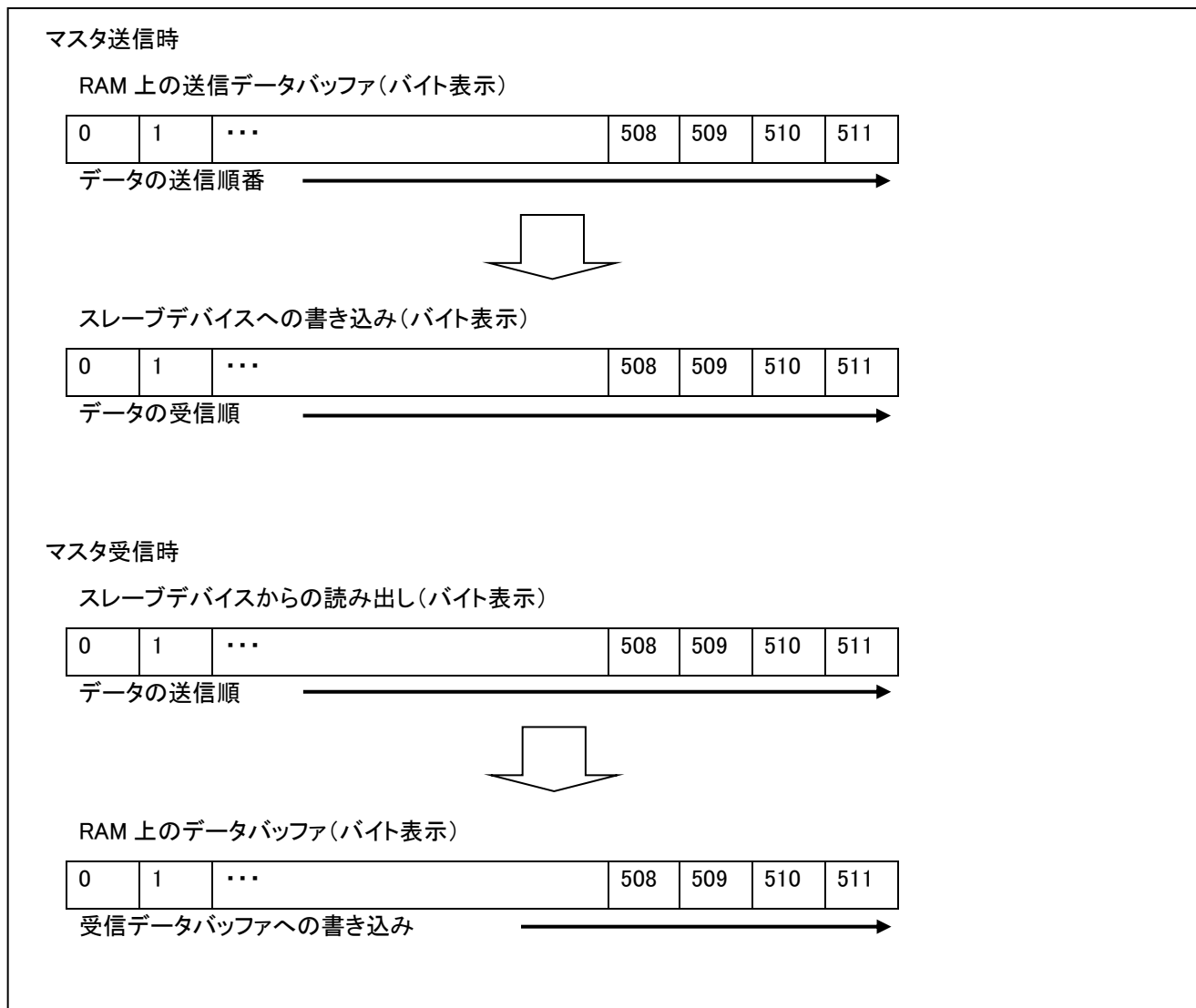


図 1-5 データバッファと送信／受信データの関係

## 1.6 DTC 使用の制限

DTC の使用に制限があります。制限事項を以下に示します。

### (1) RL78/G14 の場合

DTC : MCU が DTC 機能を持ちますが、本制御ソフトウェアは未サポートです。

## 1.7 ファイル構成

表 1-7に本制御ソフトウェアで使用するファイルを示します。

表 1-7 ファイル構成

¥an-r01an3340jj0100-rl78g14-serial	<DIR>	加速度センサ用制御ソフトウェアのフォルダ
r01an3340jj0100-rl78g14.pdf		アプリケーションノート
¥source	<DIR>	プログラム格納用フォルダ
¥com	<DIR>	共通関数格納用フォルダ 【注 1】
mtl_com.c		共通関数の各種定義
mtl_com.h.common		共通ヘッダファイル
mtl_com.h		共通関数のヘッダファイル 【注 3】
mtl_com.h.RL78		共通関数のヘッダファイル
mtl_endi.c		共通ファイル（エンディアン設定関連）
mtl_mem.c		共通ファイル（標準ライブラリ関数）
mtl_os.c		共通ファイル（標準ライブラリ関数）
mtl_os.h		共通ファイル（標準ライブラリ関数） ヘッダファイル
mtl_str.c		共通ファイル（標準ライブラリ関数）
mtl_tim.c		共通ファイル（ループタイマ関連）
mtl_tim.h		共通ファイル（ループタイマ関連） ヘッダファイル 【注 4】
mtl_tim.h.sample		ループタイマの設定値サンプル
¥r_acclsnsr_spi	<DIR>	加速度センサドライバ用フォルダ
r_acclsnsr_spi_if.h		ユーザ I/F モジュール ヘッダ
¥ref	<DIR>	コンフィギュレーションファイル用フォルダ（参照用）
r_acclsnsr_spi_config_reference.h		加速度センサ設定ヘッダファイル（参照用）
r_acclsnsr_spi_pin_config_reference.h		CS#端子設定ヘッダファイル（参照用）
¥src	<DIR>	プログラム用フォルダ
r_acclsnsr_spi.c		ユーザ I/F モジュール
r_acclsnsr_spi_private.h		ヘッダファイル
¥acclsnsr_types	<DIR>	加速度センサ個別用フォルダ
r_acclsnsr_spi_type.c		内部モジュール
¥acclsnsr_adxl362	<DIR>	ADXL362 用フォルダ
r_acclsnsr_spi_type_sub.c		内部モジュール
r_acclsnsr_spi_type_sub.h		内部モジュール ヘッダファイル
¥driver_interfaces	<DIR>	ドライバ I/F 用フォルダ
¥gpio	<DIR>	GPIO 用フォルダ
r_acclsnsr_spi_drvif_gpio.c		ドライバ I/F モジュール
¥rl78_gpio	<DIR>	RL78 用フォルダ
r_acclsnsr_spi_drvif_gpio.h		内部モジュール ヘッダファイル
r_acclsnsr_spi_drvif_gpio_dev0.c		内部モジュール（デバイス 0 用）
r_acclsnsr_spi_drvif_gpio_dev1.c		内部モジュール（デバイス 1 用）
¥interrupt	<DIR>	割り込み用フォルダ
r_acclsnsr_spi_drvif_int.c		ドライバ I/F モジュール
¥rl78_intp	<DIR>	RL78 用フォルダ
r_acclsnsr_spi_drvif_int_dev0.c		内部モジュール（デバイス 0 用）

r_acclsnsr_spi_drvif_int_dev1.c	内部モジュール (デバイス 1 用)
¥serial <DIR>	シリアル用フォルダ
r_acclsnsr_spi_drvif_serial.c	ドライバ I/F モジュール
¥rl78_sau_csi <DIR>	RL78 用フォルダ (SPI モード 3 用) 【注 2】
r_acclsnsr_spi_drvif_serial_dev0.c	内部モジュール (デバイス 0 用)
r_acclsnsr_spi_drvif_serial_dev1.c	内部モジュール (デバイス 1 用)
¥rl78_sau_csi2 <DIR>	RL78 用フォルダ (SPI モード 0 用)
r_acclsnsr_spi_drvif_serial_dev0.c	内部モジュール (デバイス 0 用)
r_acclsnsr_spi_drvif_serial_dev1.c	内部モジュール (デバイス 1 用)
¥timer <DIR>	タイマ用フォルダ
r_acclsnsr_spi_drvif_timer.c	ドライバ I/F モジュール
¥rl78_timer_rj <DIR>	RL78 用フォルダ (タイマ RJ 用)
r_acclsnsr_spi_drvif_timer_dev0.c	内部モジュール (デバイス 0 用)
r_acclsnsr_spi_drvif_timer_dev1.c	内部モジュール (デバイス 1 用)
¥r_bsp <DIR>	BSP 用フォルダ
platform.h	ヘッダファイル
¥r_config <DIR>	コンフィギュレーションファイル用フォルダ
r_acclsnsr_spi_config.h	加速度センサ設定ヘッダファイル (参照用)
r_acclsnsr_spi_pin_config.h	CS#端子設定ヘッダファイル (参照用)
¥r_sio2_csi_rl78 <DIR>	RL78 CSI 用クロック同期式シングルマスタ制御ソフトウェア (SPI モード 0) 用フォルダ
R_SIO2.h	ヘッダファイル (参照用)
R_SIO2_csi.c	ユーザ I/F モジュール
R_SIO2_csi.h	ユーザ I/F モジュール共通定義 【注 3】
R_SIO2_csi.h.rl78g14	ユーザ I/F モジュール共通定義 (RL78/G14 用)
¥sample <DIR>	動作確認プログラム格納用フォルダ
r_acclsnsr_spi_mcu_demo_adxl362.c	動作確認用のサンプルソースファイル
r_acclsnsr_spi_mcu_demo_adxl362.h	動作確認用のサンプルソースヘッダファイル

注 1 : RL78 CSI 用クロック同期式シングルマスタ制御ソフトウェアからコールされるものです。

注 2 : 「1.3 関連アプリケーションノート」に示すクロック同期式シングルマスタ制御ソフトウェア (SPI モード 3) を使用する場合に使用します。

使用するドライバ I/F モジュール選択に関しては、「2.6 コンパイル時の設定」の r\_acclsnsr\_spi\_config.h の ACCLSNSR\_SPI\_CFG\_DEVx\_MODE の設定に依存します。

注 3 : 使用ボードに合わせた RL78/G14 用のものです。

注 4 : 「1.2.2 動作環境とメモリサイズ」に合わせた設定値を入力済みです。動作条件を変更する場合、「1.3 関連アプリケーションノート」に示すクロック同期式シングルマスタ制御ソフトウェアを参照し、設定してください。

## 2. API 情報

### 2.1 ハードウェアの要求

ご使用になる MCU が以下の機能をサポートしている必要があります。

- I/O ポート
- 外部割り込み
- タイマ (タイマ RJ を使用)

### 2.2 ソフトウェアの要求

クロック同期式シングルマスタ制御ソフトウェアを用意してください。

### 2.3 サポートされているツールチェーン

本制御ソフトウェアは、「1.2.2 動作環境とメモリサイズ」に示すツールチェーンで動作確認を行っています。

### 2.4 ヘッドファイル

すべての API 呼び出しと使用されるインタフェース定義は `r_acclsnsr_spi_if.h` に記載しています。

ビルド毎の構成オプションは、`r_acclsnsr_spi_config.h` と `r_acclsnsr_spi_pin_config.h` で選択します。以下の順番でインクルードしてください。なお、`r_acclsnsr_spi_pin_config.h` のインクルードは不要です。

```
#include "r_acclsnsr_spi_if.h"  
#include "r_acclsnsr_spi_config.h"
```

### 2.5 整数型

このプロジェクトは ANSI C99 を使用しています。これらの型は `stdint.h` で定義されています。

## 2.6 コンパイル時の設定

### (1) アナログ・デバイセズ社製 ADXL362 制御時

#### (a) 制御ソフトウェアの設定

本制御ソフトウェアのコンフィグレーションオプションの設定は、`r_acclsnr_spi_config.h`で行います。  
オプション名および設定値に関する説明を下表に示します。

Configuration options in <code>r_acclsnr_spi_config.h</code>	
<code>#define ACCLNSNR_SPI_CFG_DEV0_INCLUDED</code> (1) ※デフォルト値は“1 (有効)”	デバイス番号 0 を必ず選択してください。(1: 有効) デバイス 1 個の場合、デバイス番号 0 のみを有効にしてください。
<code>#define ACCLNSNR_SPI_CFG_DEV1_INCLUDED</code> (0) ※デフォルト値は“0 (無効)”	デバイス番号 1 を使用するかを選択できます。(1: 有効、0: 無効)
<code>#define ACCLNSNR_SPI_CFG_DEVx_DRVIF_CH_NO</code> (0) ※デフォルトは“0”	デバイス番号 x で使用する下位層のクロック同期式シングルマスタ制御ソフトウェアのチャンネル番号を指定してください。
<code>#define ACCLNSNR_SPI_CFG_DEVx_ADXL362</code> (1) ※デバイス番号 0 のデフォルト値は、“1” ※デバイス番号 1 のデフォルト値は、“0”	デバイス番号 x で加速度センサ ADXL362 を制御する場合、制御対象に設定してください。(1: 制御対象、0: 制御非対象)
<code>#define ACCLNSNR_SPI_CFG_DEVx_MODE</code> <code>ACCLNSNR_SPI_MODE_RL78_SAU_CSI2</code> ※デフォルトは、 <code>ACCLNSNR_SPI_MODE_RL78_SAU_CSI2</code>	デバイス番号 x で使用する下位層のクロック同期式シングルマスタ制御ソフトウェアを設定してください。また、DMAC もしくは DTC と組み合わせて使用する場合は、合わせて定義してください。 デバイス分の設定が必要です。
<code>#define ACCLNSNR_SPI_CFG_DEVx_BR</code> (uint8_t)(0x03) デフォルト値は、“(uint8_t)(0x03)”	デバイス番号で通信する場合のビットレートレジスタに書き込む値を設定してください。

注：DEVx の“x”はデバイス番号 (x=0 or 1) を示します。

#### (b) スレーブデバイスセレクト端子の設定

本制御ソフトウェアのスレーブデバイスセレクト端子の設定は、`r_acclsnr_spi_pin_config.h`で行います。  
オプション名および設定値に関する説明を下表に示します。

Configuration options in <code>r_acclsnr_spi_pin_config.h</code>	
<code>#define ACCLNSNR_SPI_CS_DEVx_CFG_PORTNO</code> '1' ※デバイス 0 のデフォルト値は“'1'”	デバイス番号 x 用の SS# を割り付けるポート番号を設定してください。 設定値の前後にシングルコーテーション「'」をつけてください。
<code>#define ACCLNSNR_SPI_CS_DEVx_CFG_BITNO</code> '1' ※デバイス 0 のデフォルト値は“'1'”	デバイス番号 x 用の SS# を割り付けるビット番号を設定してください。 設定値の前後にシングルコーテーション「'」をつけてください。

注：DEVx の“x”はデバイス番号 (x=0 or 1) を示します。

#### (c) クロック同期式シリアル端子設定とソフトウェア設定

クロック同期式シングルマスタ制御ソフトウェアを参照し、端子を割り当て、かつソフトウェアを設定してください。

共通関数である「COM」フォルダ内のソフトウェアについてもクロック同期式シングルマスタ制御ソフトウェアを参照し、設定してください。



(d) 割り込み入力端子の設定

加速度センサの INT1 を FIFO トリガレベル割り込み（アクティブハイもしくはアクティブロー）として使用します。

割り込み入力端子は、ドライバ・インタフェース層の r\_acclsnsr\_spi\_drvif\_int\_devN.c（N はデバイス番号）で設定します。

本制御ソフトウェアは RL78/G14 の INTP5 にて、立ち下がりエッジの割り込みを検出する設定です。ユーザ環境に合わせて設定条件を見直してください。

表 2-1 に割り込み入力端子の設定方法を示します。また、変更箇所のソースコードを以下に示します。

表 2-1 割り込み入力端子の設定方法

#define 定義【注 1】	設定方法															
ACCLSNSR_SPI_INT_EGP_SFR_DEV0	外部割り込み立ち上がりエッジ許可レジスタ（EGPx）															
ACCLSNSR_SPI_INT_EGN_SFR_DEV0	外部割り込み立ち下がりエッジ許可レジスタ（EGNx）															
ACCLSNSR_SPI_INT_PMK_SFR_DEV0	割り込みマスク・フラグ・レジスタの INTPx 用ビット															
ACCLSNSR_SPI_INT_PIF_SFR_DEV0	割り込み要求フラグ・レジスタの INTPx 用ビット															
ACCLSNSR_SPI_INT_PPR0_SFR_DEV0	優先順位指定フラグ・レジスタ（PPR0x）															
ACCLSNSR_SPI_INT_PPR1_SFR_DEV0	優先順位指定フラグ・レジスタ（PPR1x）															
ACCLSNSR_SPI_INT_EGP_DOWN_TRIG_DEV0	EGPx レジスタ設定用の#define 定義。 本定義と EGPx レジスタで and 処理を行い、対象のビットを 0（立ち下がりエッジ）にします。 本定義の値は、使用する割り込み番号のビットを 0、その他を 1 としてください。 立ち上がりエッジで使用する場合、本定義値を見直し、かつ、本定義を使用している and 処理を or 処理に変更してください。															
ACCLSNSR_SPI_INT_EGN_DOWN_TRIG_DEV0	EGNx レジスタ設定用の#define 定義。 本定義と EGNx レジスタで or 処理を行い、対象のビットを 1（立ち下がりエッジ）にします。 本定義の値は、使用する割り込み番号のビットを 1、その他を 0 としてください。 立ち上がりエッジで使用する場合、本定義値を見直し、かつ、本定義を使用している or 処理を and 処理に変更してください。															
ACCLSNSR_SPI_INT_PPR0_LEVEL_DEV0、 ACCLSNSR_SPI_INT_PPR1_LEVEL_DEV0	PPR0x レジスタ、PPR1x レジスタ設定用の#define 定義 下表に従い設定してください。 <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>PPR0</th> <th>PPR1</th> <th>優先順位レベルの選択</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>レベル 0 を指定（高優先順位）</td> </tr> <tr> <td>0</td> <td>1</td> <td>レベル 1 を指定</td> </tr> <tr> <td>1</td> <td>0</td> <td>レベル 2 を指定</td> </tr> <tr> <td>1</td> <td>1</td> <td>レベル 3 を指定（低優先順位）</td> </tr> </tbody> </table>	PPR0	PPR1	優先順位レベルの選択	0	0	レベル 0 を指定（高優先順位）	0	1	レベル 1 を指定	1	0	レベル 2 を指定	1	1	レベル 3 を指定（低優先順位）
PPR0	PPR1	優先順位レベルの選択														
0	0	レベル 0 を指定（高優先順位）														
0	1	レベル 1 を指定														
1	0	レベル 2 を指定														
1	1	レベル 3 を指定（低優先順位）														

注 1： デバイス番号 1 の場合は、DEV0 を DEV1 に置き換えてください。

変更箇所のソースコード : r\_acclsnsr\_spi\_drvif\_int\_devN.c 54 行目付近

```
/* *****  
Macro definitions  
***** */  
#define ACCLSNSR_SPI_INT_DISABLE_DEV0 (1)  
#define ACCLSNSR_SPI_INT_ENABLE_DEV0 (0)  
#define ACCLSNSR_SPI_INT_CLEAR_DEV0 (0)  
#define ACCLSNSR_SPI_INT_SET_DEV0 (1)  
  
/* ==== Configuration ==== */  
#define ACCLSNSR_SPI_INT_EGP_SFR_DEV0 (EGP0)  
#define ACCLSNSR_SPI_INT_EGN_SFR_DEV0 (EGN0)  
#define ACCLSNSR_SPI_INT_PMK_SFR_DEV0 (PMK5)  
#define ACCLSNSR_SPI_INT_PIF_SFR_DEV0 (PIF5)  
#define ACCLSNSR_SPI_INT_PPR0_SFR_DEV0 (PPR05)  
#define ACCLSNSR_SPI_INT_PPR1_SFR_DEV0 (PPR15)  
#define ACCLSNSR_SPI_INT_EGP_DOWN_TRIG_DEV0 (0xdf) /* EGP5 = 0 Down trigger  
interrupt */  
#define ACCLSNSR_SPI_INT_EGN_DOWN_TRIG_DEV0 (0x20) /* EGN5 = 1 Down trigger  
interrupt */  
#define ACCLSNSR_SPI_INT_PPR0_LEVEL_DEV0 (1) /* PPR0 = 1 Level 3 */  
#define ACCLSNSR_SPI_INT_PPR1_LEVEL_DEV0 (1) /* PPR1 = 1 Level 3 */  
/* ===== */
```

## (e) タイマ出力端子の設定

タイマは、同期サンプリング用のタイマ出力目的で使用します。

タイマ出力端子は、ドライバ・インタフェース層の `r_acclsnsr_spi_drvif_timer_devN.c` (N はデバイス番号) で設定してください。

本制御ソフトウェアは、RL78/G14 のタイマ RJ パルス出力モードを使用する設定です。ユーザ環境に合わせて設定条件を見直してください。

表 2-2 にタイマ RJ の設定条件、表 2-3 にタイマ出力端子の設定方法を示します。また、変更箇所のソースコードを以下に示します。

表 2-2 使用するタイマ RJ の設定

項目	内容
使用端子	TRJO0
動作モード	パルス出力モード
出力極性	L から出力開始 (初期値レベル : L)
パルス出力期間	H 期間 : 5ms、L 期間 : 5ms

表 2-3 タイマ出力端子の設定方法

#define 定義【注 1】	設定方法
ACCLSNSR_SPI_TIMER_P_PORT_DEV0	ポート・レジスタの TRJOx 用ビット
ACCLSNSR_SPI_TIMER_PM_PORT_DEV0	ポート・モード・レジスタの TRJOx 用ビット
ACCLSNSR_SPI_TIMER_TRJIF_SFR_DEV0	割り込み要求フラグ・レジスタの TRJOx 用ビット
ACCLSNSR_SPI_TIMER_TRJMK_SFR_DEV0	割り込みマスク・フラグ・レジスタの TRJOx 用ビット
ACCLSNSR_SPI_TIMER_TRJEN_SFR_DEV0	タイマ RJx の入力クロック供給の制御ビット
ACCLSNSR_SPI_TIMER_TRJIOC_SFR_DEV0	タイマ RJ I/O 制御レジスタ x (TRJIOCx)
ACCLSNSR_SPI_TIMER_TRJMR_SFR_DEV0	タイマ RJ モードレジスタ x (TRJMRx)
ACCLSNSR_SPI_TIMER_TRJ_SFR_DEV0	タイマ RJ カウンタレジスタ x (TRJx)
ACCLSNSR_SPI_TIMER_TRJCR_SFR_DEV0	タイマ RJ 制御レジスタ x (TRJCRx)
ACCLSNSR_SPI_TIMER_TRJIOC_ENABLE_DEV0	TRJIOCx レジスタ設定用の#define 定義 タイマ RJ の入出力条件を設定してください。【注 2】
ACCLSNSR_SPI_TIMER_TRJMR_ENABLE_DEV0	TRJMRx レジスタ設定用の#define 定義 タイマ RJ の動作モードを設定してください。【注 2】
ACCLSNSR_SPI_TIMER_TRJ_ENABLE_DEV0	TRJx レジスタ設定用の#define 定義 タイマ RJ のカウンタ値を設定してください。【注 2】 設定可能範囲 : 0x0000 – 0xFFFF
ACCLSNSR_SPI_TIMER_TRJCR_ENABLE_DEV0	TRJCRx レジスタ設定用の#define 定義 タイマ RJ の開始条件を設定してください。【注 2】 但し、TRJCRx.TSTART ビットは必ず 1 にしてください。

注 1 : デバイス番号 1 の場合は、DEV0 を DEV1 に置き換えてください。

注 2 : 設定値の詳細は、使用する MCU のユーザーズマニュアル ハードウェア編をご参照ください。

変更箇所のソースコード : r\_acclsnsr\_spi\_drvif\_timer\_devN.c 54 行目付近

```

/*****
Macro definitions
*****/
#define ACCLSNSR_SPI_TIMER_HI_DEV0    (uint8_t) (0x01)    /* Port "H" */
#define ACCLSNSR_SPI_TIMER_LOW_DEV0   (uint8_t) (0x00)    /* Port "L" */
#define ACCLSNSR_SPI_TIMER_OUT_DEV0   (uint8_t) (0x00)    /* Port output setting */
#define ACCLSNSR_SPI_TIMER_IN_DEV0    (uint8_t) (0x01)    /* Port input setting */
#define ACCLSNSR_SPI_TIMER_CLEAR_DEV0 (0)
#define ACCLSNSR_SPI_TIMER_SET_DEV0   (1)
#define ACCLSNSR_SPI_TIMER_DISABLE_DEV0 (0)
#define ACCLSNSR_SPI_TIMER_ENABLE_DEV0 (1)
#define ACCLSNSR_SPI_TIMER_INT_DISABLE_DEV0 (1)
#define ACCLSNSR_SPI_TIMER_INT_ENABLE_DEV0 (0)

/* ==== Configuration ==== */
#define ACCLSNSR_SPI_TIMER_P_PORT_DEV0 (P3_bit.no0)
#define ACCLSNSR_SPI_TIMER_PM_PORT_DEV0 (PM3_bit.no0)
#define ACCLSNSR_SPI_TIMER_TRJIF_SFR_DEV0 (TRJIF0)
#define ACCLSNSR_SPI_TIMER_TRJMK_SFR_DEV0 (TRJMK0)
#define ACCLSNSR_SPI_TIMER_TRJEN_SFR_DEV0 (TRJ0EN)
#define ACCLSNSR_SPI_TIMER_TRJIOC_SFR_DEV0 (TRJIOC0)
#define ACCLSNSR_SPI_TIMER_TRJMR_SFR_DEV0 (TRJMR0)
#define ACCLSNSR_SPI_TIMER_TRJ_SFR_DEV0 (TRJ0)
#define ACCLSNSR_SPI_TIMER_TRJCR_SFR_DEV0 (TRJCR0)

#define ACCLSNSR_SPI_TIMER_TRJIOC_ENABLE_DEV0 (0x04) /* Enable TRJ0 output,
L output */
#define ACCLSNSR_SPI_TIMER_TRJMR_ENABLE_DEV0 (0x11) /* Pulse output mode,
TRJ_CLK = fCLK/8 = 4MHz (fCLK=32MHz) */
#define ACCLSNSR_SPI_TIMER_TRJ_ENABLE_DEV0 (20000) /* 20000 = (4MHz / 2 /
100Hz) */
#define ACCLSNSR_SPI_TIMER_TRJCR_ENABLE_DEV0 (0x01) /* Start timer */
/* ===== */

```

## 2.7 引数

API 関数の引数である構造体を示します。この構造体は API 関数のプロトタイプ宣言とともに `r_acclsnsr_spi_if.h` で記載されています。

```
typedef struct
{
    uint32_t    cnt;
    uint32_t    addr;
    uint8_t     * p_data;
} acclsnsr_spi_info_t;          /* 12 bytes          */
```

## 2.8 戻り値

API 関数の戻り値を示します。この列挙型は API 関数のプロトタイプ宣言とともに `r_acclsnsr_spi_if.h` で記載されています。

```
typedef enum e_acclsnsr_err
{
    ACCLSNSR_SPI_SUCCESS      = 0, /* Successful operation */
    ACCLSNSR_SPI_ERR_PARAM    = -1, /* Parameter error      */
    ACCLSNSR_SPI_ERR_HARD     = -2, /* Hardware error       */
    ACCLSNSR_SPI_ERR_TIMEOUT  = -6, /* Time out error       */
    ACCLSNSR_SPI_ERR_OTHER    = -7, /* Other error          */
} acclsnsr_spi_err_t;
```

### 3. API 関数

#### 3.1 R\_ACCLSNSR\_SPI\_Open

##### Format

```
acclsnsr_spi_err_t R_ACCLSNSR_SPI_Open(  
    uint8_t devno  
)
```

##### Parameters

*devno*

デバイス番号 (0, 1)

##### Return Values

*ACCLSNSR\_SPI\_SUCCESS* /\* 正常終了した場合 \*/

*ACCLSNSR\_SPI\_ERR\_PARAM* /\* パラメータ異常の場合 \*/

##### Properties

r\_acclsnsr\_spi\_if.h にプロトタイプ宣言されています。

##### Description

引数 *devno* で指定したデバイス番号のスレーブデバイスセレクト制御信号に使用するポートを初期化します。初期化後は、汎用出力ポート H 出力状態になります。通信中は本関数をコールしないでください。通信中にコールした場合の通信は保証されません。

##### Special Notes:

なし

### 3.2 R\_ACCLSNSR\_SPI\_Close

**Format**

```
acclsnsr_spi_err_t R_ACCLSNSR_SPI_Close(  
    uint8_t devno  
)
```

**Parameters**

*devno*

デバイス番号 (0, 1)

**Return Values**

*ACCLSNSR\_SPI\_SUCCESS* /\* 正常終了した場合 \*/

*ACCLSNSR\_SPI\_ERR\_PARAM* /\* パラメータ異常の場合 \*/

**Properties**

r\_acclsnsr\_spi\_if.h にプロトタイプ宣言されています。

**Description**

引数 *devno* で指定したデバイス番号のスレーブデバイスセレクト端子の機能を汎用入出力ポートにします。実行後は、汎用入力ポート状態になります。通信中は本関数をコールしないでください。通信中にコールした場合の通信は保証されません。

**Special Notes:**

なし

### 3.3 R\_ACCLSNSR\_SPI\_WriteReg

#### Format

```
acclsnsr_spi_err_t R_ACCLSNSR_SPI_WriteReg(  
    uint8_t devno,  
    acclsnsr_spi_info_t * p_acclsnsr_spi_info  
)
```

#### Parameters

*devno*

デバイス番号 (0, 1)

\* *p\_acclsnsr\_spi\_info*

加速度センサ設定情報構造体。

*cnt*

加速度センサのレジスタに書き込むデータサイズ (単位はバイトで、1 以上) を設定してください。

*addr*

加速度センサのレジスタアドレスを設定してください。

ただし、リードオンリーのレジスタアドレスを設定しないでください。設定した場合、そのアドレスに対して書き込み処理を実行します。

\* *p\_data*

加速度センサのレジスタに書き込むデータバッファのアドレスを設定してください。

#### Return Values

*ACCLSNSR\_SPI\_SUCCESS* /\* 正常終了した場合 \*/  
*ACCLSNSR\_SPI\_ERR\_PARAM* /\* パラメータ異常の場合 \*/  
*ACCLSNSR\_SPI\_ERR\_HARD* /\* ハードウェアエラーの場合 \*/  
*ACCLSNSR\_SPI\_ERR\_OTHER* /\* その他のエラーの場合 \*/

#### Properties

*r\_acclsnsr\_spi\_if.h* にプロトタイプ宣言されています。

#### Description

*p\_data* のデータを加速度センサの指定レジスタアドレスから指定バイト数分書き込みます。

#### Special Notes:

なし



### 3.4 R\_ACCLSNSR\_SPI\_ReadReg

#### Format

```
acclsnsr_spi_err_t R_ACCLSNSR_SPI_ReadReg(  
    uint8_t devno,  
    acclsnsr_spi_info_t * p_acclsnsr_spi_info  
)
```

#### Parameters

*devno*

デバイス番号 (0, 1)

\* *p\_acclsnsr\_spi\_info*

加速度センサ設定情報構造体。

*cnt*

加速度センサのレジスタから読み出すデータサイズ (単位はバイトで、1 以上) を設定してください。

*addr*

加速度センサのレジスタアドレスを設定してください。

ただし、ライトオンリーのレジスタアドレスを設定しないでください。設定した場合、そのアドレスに対して読み出し処理を実行します。

\* *p\_data*

加速度センサのレジスタから読み出した値を格納するデータバッファのアドレスを設定してください。

#### Return Values

*ACCLSNSR\_SPI\_SUCCESS* /\* 正常終了した場合 \*/  
*ACCLSNSR\_SPI\_ERR\_PARAM* /\* パラメータ異常の場合 \*/  
*ACCLSNSR\_SPI\_ERR\_HARD* /\* ハードウェアエラーの場合 \*/  
*ACCLSNSR\_SPI\_ERR\_OTHER* /\* その他のエラーの場合 \*/

#### Properties

*r\_acclsnsr\_spi\_if.h* にプロトタイプ宣言されています。

#### Description

加速度センサの指定レジスタアドレスから指定バイト数分、データを読み出し、*p\_data* に格納します。

#### Special Notes:

なし

### 3.5 R\_ACCLSNSR\_SPI\_ReadFIFO

#### Format

```
acclsnsr_spi_err_t R_ACCLSNSR_SPI_ReadFIFO(  
    uint8_t devno,  
    acclsnsr_spi_info_t * p_acclsnsr_spi_info  
)
```

#### Parameters

*devno*

デバイス番号 (0, 1)

\* *p\_acclsnsr\_spi\_info*

加速度センサ設定情報構造体。

*cnt*

加速度センサの FIFO バッファから読み出す FIFO サンプル数 (1~512) を設定してください。

1 サンプルは 2 バイトです。

*addr*

設定不要

\* *p\_data*

加速度センサの FIFO バッファから読み出した値を格納するデータバッファのアドレスを設定してください。

1 サンプルは 2 バイトのため、確保するバッファサイズに注意してください。

#### Return Values

*ACCLSNSR\_SPI\_SUCCESS* /\* 正常終了した場合 \*/

*ACCLSNSR\_SPI\_ERR\_PARAM* /\* パラメータ異常の場合 \*/

*ACCLSNSR\_SPI\_ERR\_HARD* /\* ハードウェアエラーの場合 \*/

*ACCLSNSR\_SPI\_ERR\_OTHER* /\* その他のエラーの場合 \*/

#### Properties

*r\_acclsnsr\_spi\_if.h* にプロトタイプ宣言されています。

#### Description

加速度センサの FIFO バッファから指定 FIFO サンプル数分、データを読み出し、*p\_data* に格納します。

事前に、FIFO トリガレベルは、FIFO\_CONTROL レジスタの AH ビットと FIFO\_SAMPLES レジスタを使って設定してください。

アナログ・デバイセズ社製 ADXL362 制御時、サンプリングされた各データは 2 バイト構成で、最下位バイト→最上位バイトの順で格納されます。

#### Special Notes:

なし

### 3.6 R\_ACCLSNSR\_SPI\_GetIntStatus

#### Format

```
acclsnsr_spi_err_t R_ACCLSNSR_SPI_GetIntStatus(  
    uint8_t devno,  
    uint8_t * p_status  
)
```

#### Parameters

*devno*

デバイス番号 (0, 1)

\* *p\_status*

割り込みステータスバッファポインタ (1 バイト) を設定してください。

ACCLSNSR\_SPI\_INT\_OCCUR : 割り込み要求検出

ACCLSNSR\_SPI\_INT\_NOT\_OCCUR : 割り込み要求未検出

#### Return Values

ACCLSNSR\_SPI\_SUCCESS /\* 正常終了した場合 \*/

ACCLSNSR\_SPI\_ERR\_PARAM /\* パラメータ異常の場合 \*/

#### Properties

r\_acclsnsr\_spi\_if.h にプロトタイプ宣言されています。

#### Description

*devno* で指定したデバイスの INT1 端子割り込み要求フラグ (割り込みハンドラで設定されることを想定) を確認します。

割り込み要求フラグがセット状態の場合、割り込み要求フラグをクリアし、*p\_status* に

ACCLSNSR\_SPI\_INT\_OCCUR をセットします。

割り込み要求フラグがクリア状態の場合、*p\_status* に ACCLSNSR\_SPI\_INT\_NOT\_OCCUR をセットします。

#### Special Notes:

本関数のコール後、*p\_status* が ACCLSNSR\_SPI\_INT\_OCCUR の場合、割り込み発生要因がクリアされるまで (サンプル数が FIFO トリガレベルを下回るまで) FIFO バッファからデータ読み出してください。そうすることで、MCU は再度 INT1 割り込み要求を受け付けることができます。

割り込みハンドラとして Int Drv I/F レイヤ r\_acclsnsr\_spi\_drvif\_int\_devN.c (N はデバイス番号) を使用します。割り込み発生フラグ g\_acclsnsr\_spi\_int\_flag\_devN (N はデバイス番号) を持ち、割り込み要求発生時に、このフラグをセットします。

### 3.7 R\_ACCLSNSR\_SPI\_StartSampling

**Format**

```
acclsnsr_spi_err_t R_ACCLSNSR_SPI_StartSampling(  
    uint8_t devno  
)
```

**Parameters**

*devno*

デバイス番号 (0, 1)

**Return Values**

*ACCLSNSR\_SPI\_SUCCESS* /\* 正常終了した場合 \*/

*ACCLSNSR\_SPI\_ERR\_PARAM* /\* パラメータ異常の場合 \*/

**Properties**

r\_acclsnsr\_spi\_if.h にプロトタイプ宣言されています。

**Description**

割り込み信号の受付を許可状態に設定し、タイマを起動し、同期用トリガ信号の出力を開始します。

**Special Notes:**

なし

### 3.8 R\_ACCLSNSR\_SPI\_StopSampling

**Format**

```
acclsnsr_spi_err_t R_ACCLSNSR_SPI_StopSampling(  
    uint8_t devno  
)
```

**Parameters**

*devno*

デバイス番号 (0, 1)

**Return Values**

ACCLSNSR\_SPI\_SUCCESS /\* 正常終了した場合 \*/

ACCLSNSR\_SPI\_ERR\_PARAM /\* パラメータ異常の場合 \*/

**Properties**

r\_acclsnsr\_spi\_if.h にプロトタイプ宣言されています。

**Description**

割り込み信号の受付を禁止状態に設定し、タイマを停止し、トリガ信号出力を停止します。

**Special Notes:**

なし

### 3.9 R\_ACCLSNSR\_SPI\_GetVersion

**Format**

```
uint32_t R_ACCLSNSR_SPI_GetVersion(  
    void  
)
```

**Parameters**

なし

**Return Values**

バージョン番号                      上位2バイト：メジャーバージョン、下位2バイト：マイナーバージョン

**Properties**

r\_flash\_spi\_if.h にプロトタイプ宣言されています。

**Description**

バージョン情報を返します。

**Special Notes:**

なし

## 4. 注意事項

### 4.1 RAM 初期化に関する注意事項

RL78 ファミリマイコンは、RAM パリティ・エラー検出機能（初期値：パリティ・エラー・リセット発生を許可）があります。

RAM パリティ・エラー・リセット発生を許可する場合、使用する RAM 領域をデータ読み出し前に初期化する必要があります。関数コール時のスタック割り付けを想定し、BSS 領域とスタック領域の RAM を初期化してください。

詳細は、使用するマイコンのユーザーズマニュアル ハードウェア編の「RAM パリティ・エラー検出機能」を参照してください。

### 4.2 加速度センサ制御ソフトウェア共通の注意事項

本制御ソフトウェアは、加速度センサのレジスタ空間へのアクセス API を提供するものであり、加速度センサの全ての機能を確認したものではありません。また、取得したセンサデータ値の妥当性を確認したものではありません。

加速度センサのデータシートを参照し、各加速度センサの機能を利用してください。

### 4.3 アナログ・デバイセズ社製 ADXL362 制御例

参照したデータシートは Rev.D です。

#### (1) デモプログラム r\_acclsnsr\_spi\_mcu\_demo\_adxl362.c/h について

##### (a) 処理概要

本制御ソフトウェアの初期化→ADXL362 のリセット制御→ADXL362 の初期化を実施後（注 1）、サンプリングを開始し（注 2）、FIFO ストリームモードを使って、FIFO バッファから X 軸/Y 軸/Z 軸のデータを取得します。そのため、INT1 を FIFO トリガレベル割り込み、INT2 を同期トリガ入力に割り当てます。またマクロ ADXL362\_AXIS\_BUFF\_SIZE に FIFO トリガレベルを設定してください。

INT1 端子が H→L に遷移した場合（注 3）、MCU は割り込み要求を検出し、ADXL362 の FIFO バッファから FIFO トリガレベルに設定したサンプル数分のデータを読み出します。読み出した順に g\_fifo[] の先頭からデータを格納します。割り込み発生時の読み出しについては、「3.6 R\_ACCLSNSR\_SPI\_GetIntStatus」を参照してください。

注 1：FIFO トリガレベル設定等、加速度センサの各制御レジスタをセットアップする必要があります。設定値については、システムに合わせて見直してください。

注 2：サンプリング条件を変更する場合、「2.6(1)(e) タイマ出力端子の設定」を参照してください。

注 3：割り込み条件を変更する場合、「2.6(1)(d) 割り込み入力端子の設定」を参照してください。

#### (2) 電源投入後の待ちについて

電源投入後、加速度センサをスタンバイ状態にするための待ち時間が必要です。

#### (3) ソフトウェアリセット実行後のリセット待ちについて

R\_ACCLSNSR\_SPI\_WriteReg()関数を使って、ソフト・リセット（コード：0x52）を実行した場合、リセット解除待ち（データシートに約 0.5ms と記載あり）が必要です。

ユーザアプリケーションにて、システムに合った待ち処理と十分な待ち時間を設定してください。

#### (4) 同期サンプリング時のトリガパルス幅とトリガ前アサート解除時間について

トリガ信号のパルス幅条件と、トリガ前のアサート解除時間条件があります。データシートの「同期データ・サンプリングの使い方」を参照してください。

#### (5) 同期サンプリングのサンプリング周波数について

最大サンプリング周波数条件があります。データシートの「同期データ・サンプリングの使い方」を参照してください。

#### (6) 複数デバイス制御について

複数デバイスをサポートする場合、デバイス毎にスレーブデバイスセレクト制御端子と割り込み入力端子を割り当ててください。

#### (7) 電源条件について

電源投入時、常に 0V からスタートアップする必要があります。電源切断時、電源（Vs、VDD I/O）を完全に放電させる必要があります。詳細は、加速度センサのデータシートを参照してください。



## ホームページとサポート窓口

ルネサス エレクトロニクスホームページ

<http://japan.renesas.com/>

お問い合わせ先

<http://japan.renesas.com/contact/>

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。

## 改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2016.08.31	—	新規作成

## 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. 未使用端子の処理

【注意】未使用端子は、本文の「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。未使用端子は、本文「未使用端子の処理」で説明する指示に従い処理してください。

### 2. 電源投入時の処置

【注意】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. リザーブアドレス（予約領域）のアクセス禁止

【注意】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、将来の機能拡張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。

### 4. クロックについて

【注意】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。

### 5. 製品間の相違について

【注意】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っていると、内部ROM、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。お客様の機器・システムの設計において、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には、お客様の責任において行ってください。これらの使用に起因して、お客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
2. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
3. 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に関し、当社は、何らの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。
4. 当社製品を改造、改変、複製等しないでください。かかる改造、改変、複製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。
5. 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）、交通用信号機器、  
防災・防犯装置、各種安全装置等  
当社製品は、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（原子力制御システム、軍事機器等）に使用されることを意図しておらず、使用することはできません。たとえ、意図しない用途に当社製品を使用したことによりお客様または第三者に損害が生じても、当社は一切その責任を負いません。なお、ご不明点がある場合は、当社営業にお問い合わせください。
6. 当社製品をご使用の際は、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他の保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障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。
7. 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、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、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。
8. 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。お客様が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。
9. 本資料に記載されている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。また、当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍用用途に使用しないでください。当社製品または技術を輸出する場合は、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。
10. お客様の転売等により、本ご注意書き記載の諸条件に抵触して当社製品が使用され、その使用から損害が生じた場合、当社は何らの責任も負わず、お客様にてご負担して頂きますのでご了承ください。
11. 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。

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とその総株主の議決権の過半数を直接または間接に保有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。



ルネサス エレクトロニクス株式会社

■営業お問合せ窓口

<http://www.renesas.com>

※営業お問合せ窓口の住所は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス株式会社 〒135-0061 東京都江東区豊洲3-2-24（豊洲フォレシア）

■技術的なお問合せおよび資料のご請求は下記へどうぞ。  
総合お問合せ窓口：<http://japan.renesas.com/contact/>