
RX ファミリ

R01AN1984JJ0112

Rev.1.12

2017.03.31

EPTPC : PTP タイマ同期開始サンプルプログラム

要旨

本アプリケーションノートは、EPTPC FIT (Firmware Integration Technology) モジュール[1]を使用した応用例です。IEEE1588-2008[1]で定義された PTP (Precision Time Protocol) で同期した時刻でのモータ制御タイマの開始を、CPU を介さないで実行する例を説明します。

動作確認デバイス

以下のデバイスがこの実例でサポートされます。

- RX64M グループ
- RX71M グループ

本アプリケーションノートを他のマイコンへ適用する場合、そのマイコンの仕様にあわせて変更し、十分評価してください。

目次

1. 概要	2
2. 機能情報	6
3. サンプルプログラムの仕様	10
4. 参考資料	18

1. 概要

本アプリケーションノートでは、EPTPC FIT モジュール（以後、PTP ドライバ）を使用した典型的な使用例を説明します。この例では、マルチファンクションタイマパルスユニット（MTU3）や汎用 PWM タイマ（GPT）を同期した時刻での開始を、イベントリンクコントローラ（ELC）を使用することで CPU 処理を介さず行っています。時刻同期プロトコルは PTP を使用しています。ユーザは同期した時刻で出力した PWM 波形の位相差を観測し、比較することができます。

1.1 FIT モジュールを使用した PTP タイマ同期開始サンプルプログラム

このサンプルプログラムはプロジェクト形式で提供し、PTP ドライバの応用例として使用できます。

1.2 関連ドキュメント

- [1] RX ファミリ EPTPC モジュール Firmware Integration Technology, Rev.1.13, Document No. R01AN1943JJ0113, Mar 31, 2017
- [2] IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems, Revision of IEEE Std 1588-2008, Mar 2008
- [3] RX ファミリ イーサネットモジュール Firmware Integration Technology, Rev.1.12, Document No. R01AN2009JJ0112, Nov 11, 2016
- [4] Renesas USB MCU USB Basic Host and Peripheral firmware using Firmware Integration Technology, Rev.1.10, Document No. R01AN2025EJ0110, Dec 26, 2015
- [5] Renesas USB MCU USB Host Mass Storage Class Driver (HMSC) using Firmware Integration Technology, Rev.1.10, Document No. R01AN2026EJ0110, Dec 26, 2015
- [6] RX ファミリ オープンソース FAT ファイルシステム M3S-TFAT-Tiny モジュール Firmware Integration Technology, Rev.3.00, Document No. R20AN0038JJ0300, Apr.01, 2014
- [7] HMI 拡張ボード ユーザーズマニュアル、Rev.1.01, R0K50564MB001BR, Jan 16, 2015
- [8] Renesas Starter Kit+ for RX64M, ユーザーズマニュアル、Rev.1.20, Document No. R20UT2590JG0102, Jun 25, 2015
- [9] Renesas Starter Kit+ for RX71M, ユーザーズマニュアル、Rev.1.00, Document No. R20UT3217JG0100, Jan 23, 2015

1.3 語彙・略語

EPTPC FIT モジュール アプリケーションノート[1]の 1.3 節を参照ください。

1.4 ハードウェアの構成

RX64M/71M グループのイーサネットモジュールは、EPTPC、PTP 用イーサネットコントローラ用 DMA コントローラ (PTPEDMAC)、2 チャンネルのイーサネットコントローラ (ETHERC (CH0)、ETHERC (CH1))、および2 チャンネルのイーサネットコントローラ用 DMA コントローラ (EDMAC (CH0)、EDMAC (CH1)) で構成しています。EPTPC は、PTP 同期フレーム処理部 (CH0)、PTP 同期フレーム処理部 (CH1)、パケット中継部 (PRC-TC)、および統計的クロック補正部で構成しています。また、EPTPC は、I/O ポートとモータ制御タイマ (MTU3 および GPT 周辺モジュール) に ELC 周辺モジュールを介して接続しており、PTP で時刻同期したパルスを出力できます。

イーサネットモジュールの概要を以下に示します。また、ハードウェアブロック図を図 1.1 に示します。

1. 時刻同期機能

- ・ IEEE1588-2008 バージョン2 に準拠
- ・ PTP メッセージ (イーサネットフレーム¹とUDP/IPv4 フォーマット²) による時刻同期
- ・ マスタまたはスレーブとして動作し、OC、BC、TC デバイスのいずれにも対応
- ・ 時刻偏差の統計的な補正 (傾き予測時刻補正アルゴリズム)
- ・ タイマイベント出力 (6 チャンネル、立ち上がり/立ち下がりエッジ、イベントフラグの自動クリア)
- ・ ELC 経由のタイマイベントで、モータ制御タイマ (MTU3、GPT) の同期スタート
- ・ EPTPCとELC経由で接続したI/OポートからPTPで時刻同期したパルスを出力
- ・ PTP メッセージの処理を選択可能 (PTP モジュール内部での処理、PTPEDMAC 経由のCPU 処理、他のポートを使用)

2. 標準イーサネット機能

- ・ 独立した2 チャンネルのイーサネット動作
- ・ ハードウェア簡易スイッチ (フレームの中継はカットスルー方式またはストア&フォワード方式から選択可能)
- ・ ハードウェアマルチキャストフレームのフィルタリング (全て受信、全て受信しない、特定の2 種類のフレームのみ受信)

¹RX64M グループではEthernet II フレームフォーマットのみサポートします (IEEE802.3 フレームフォーマットはサポートしません)。

²UDP IPv6 はサポートしません。

詳細に関しては「RX64M/71M グループユーザズマニュアル ハードウェア編」を参照ください。

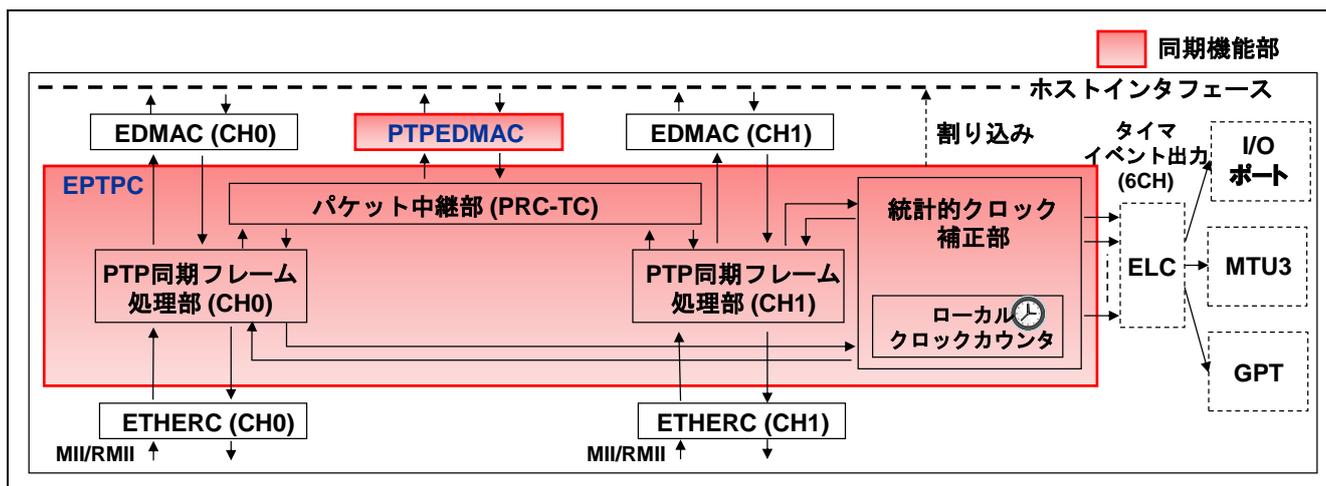


図1.1 ハードウェアブロック図

1.5 ソフトウェアの構成

このサンプルプログラムは複数の FIT モジュールを使用したアプリケーション層での実行例です。実行される動作は、PTP 構成情報の USB メモリからの読み出し、PTP ドライバへの PTP 構成情報の設定、USB メモリからタイマ開始時刻の読み出しと PTP ドライバへの設定、EPTPC と MTU3/GPT 間のイベントリンクの設定、MTU3/GPT ドライバへの PWM 出力の設定、PTP および Ether ドライバを使用した PTP による制御です。PTP 構成情報には MAC アドレス、IP アドレス、クロックの種類、マスタ/スレーブの別、遅延メカニズム (P2P/E2E の設定) が含まれています。PTP ドライバは必ず Ether ドライバ[3]と共に使用する必要があります。TCP/IP ミドルウェアはこの例には含まれていませんので、TCP/IP を使用するには、別途、TCP/IP ミドルウェアを用意する必要があります。SUSB ホストドライバ[4]、[5]は、ATA インタフェースと USB マスタストレージクラス機能が組み込まれており、USB メモリのマウント、データの読み出しと書き込み、USB メモリの検出など、USB メモリへのアクセスを実行します。M3S-TFS-Tiny [6]は、ファイルのオープン、クローズ、リード、ライトなどのファイルのアクセスを実行する FAT ファイルシステムとして組み込まれています。図 1.2にサンプルプログラムを使用した場合のソフトウェア構成を示します。

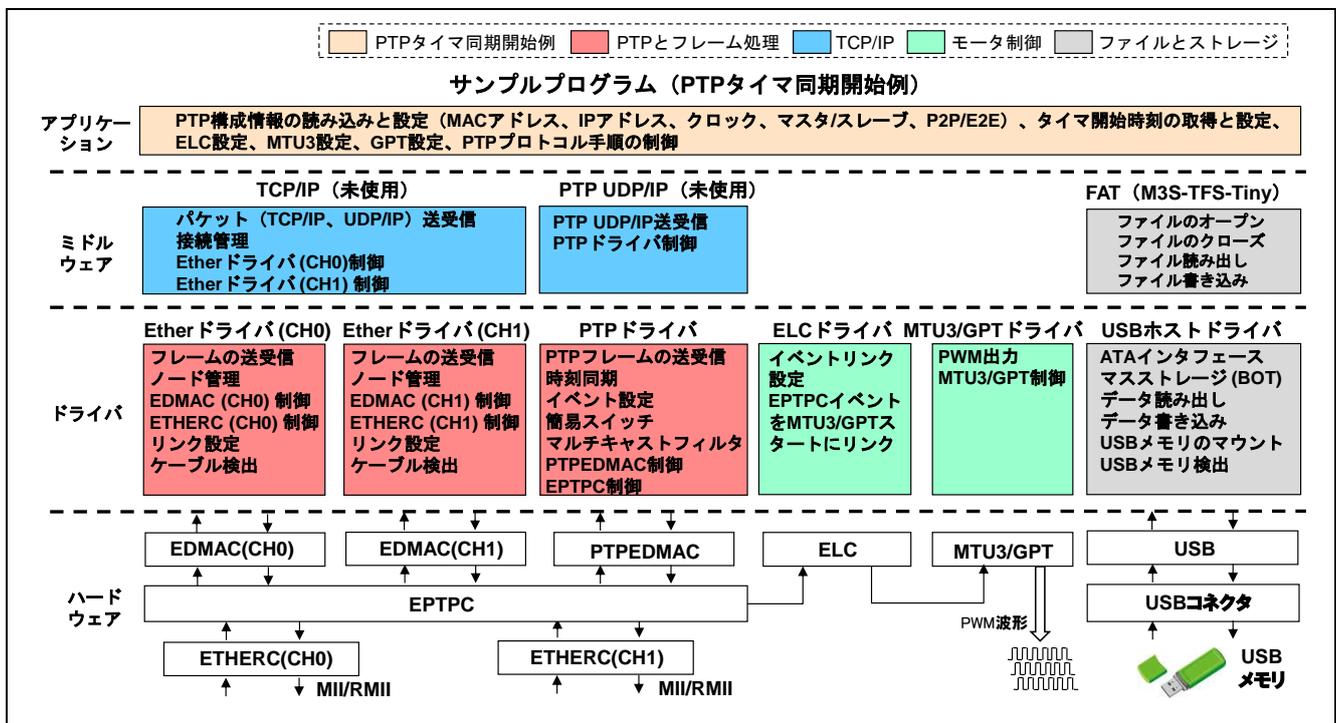


図1.2 ソフトウェア構成

1.6 ファイル構成

このサンプルプログラムのコードは `demo_src` と下位階層のフォルダに格納されています。また、ELC、MTU3、および GPT ドライバはそれぞれのドライバフォルダに格納されています。図 1.3はファイル構成を示しています。PTP ドライバを含め他の FIT ベースのモジュールに関しては、それぞれの FIT モジュール (BSP、イーサネットドライバ、PTP ドライバ、FAT ファイルシステム、USB ドライバ) のドキュメントを参照してください。

demo_src:メイン動作と構成設定	r_config: FITモジュールの構成設定
sample_main.c	r_bsp_config.h
sample_main.h	r_bsp_interrupt_config.h
	r_cmt_rx_config
+ --- led_7seg: 7セグメントLED制御	r_ether_rx_config.h
led_7seg.c	r_ptp_rx_config.h
led_7seg.h	r_usb_basic_config.h
	r_usb_hmsc_config.h
+ --- sync_if: PTP同期動作	r_elc_rx: ELCドライバ フォルダ
sync_if.c	r_elc_rx_if.h ;ELCドライバ ヘッダファイル
sync_if.h	
+ --- tfat_if:ファイルシステムとUSBドライバ間のインタフェース	+ --- src:
file_if.c	r_elc.c ; ELCドライバ ソースファイル
file_if.h	r_ether_rx:イーサネットドライバ FITモジュール
r_data_file.c	r_gpt_rx: GPTドライバ フォルダ
r_data_file.h	r_gpt_rx_if.h ;GPTドライバ ヘッダファイル
r_tfat_drv_if.c ; USBドライバインタフェース	
+ --- usb_if: USBホスト メモリアクセス制御	+ --- src:
r_usb_hmsc_defep.c	r_gpt.c ; GPTドライバ ソースファイル
usb_memory_access.c	
+ --- usb_memory_sample: USBメモリのデータサンプル	r_mtu3_rx: MTU3ドライバ フォルダ
+ --- PARAM: 同期構成情報サンプル	r_mtu3_rx_if.h ;MTU3ドライバ ヘッダファイル
+ --- PULSE: タイマ開始情報サンプル	
+ --- MTU3: MTU3を使用する場合	+ --- src:
+ --- GPT: GPTを使用する場合	r_mtu3.c ; MTU3ドライバ ソースファイル
+ --- usr: LED制御	r_ptp_rx: PTPドライバ FITモジュール
led.c	r_tfat_rx: FATファイルシステム (M3S-TFS-Tiny) FITモジュール
led.h	
r_bsp: BSP (Board Support Package) FITモジュール	r_usb_basic: USBドライバ(USB基本処理) FITモジュール
r_cmt_rx: コンペアマッチタイマ FITモジュール	r_usb_hmsc: USBドライバ(ホストマスストレージクラス) FITモジュール

図1.3 ファイル構成

2. 機能情報

本サンプルプログラムは以下の要件に合わせて設計されています。

2.1 ハードウェア要件

サンプルプログラムでは、使用する MCU が以下の機能をサポートしている必要があります。

- EPTPC
- ETHERC
- EDMAC
- ELC
- MTU3
- GPT
- CMT
- USB

2.2 ハードウェアリソース要件

サンプルプログラムで使用するドライバが必要とする周辺回路ハードウェアについて説明します。特に明記されていない限り、周辺回路は専らドライバで使用され、ユーザアプリケーションで使用することはできません。

2.2.1 EPTPC チャンネル

この例では、EPTPC を使用します。この回路 PTP に準拠した時刻同期動作と ELC 経由での MTU3/GPT の開始に必要です。

2.2.2 ETHERC チャンネル

この例では、クロック（ノード）の種類に応じて、ETHERC（CH0）、ETHERC（CH1）、またはその両方を使用します。これらの周辺回路はイーサネット MAC 動作に必要です。

2.2.3 EDMAC チャンネル

この例では、クロック（ノード）の種類に応じて、EDMAC（CH0）、EDMAC（CH1）、またはその両方を使用します。これらの周辺回路は通常のイーサネットフレーム動作における CPU ホストインタフェースとして必要です。

2.2.4 ELC

この例では EPTPC と MTU3/GPT の間のイベント接続に ELC を使用します。この周辺回路は MTU3/GPT の同期開始に必要です。

2.2.5 MTU3 チャンネル

この例では同期した PWM 出力に MTU3 チャンネルを使用します。この回路の動作中は設定変更や周辺チャンネルの使用を行うことはできません。

2.2.6 GPT チャンネル

この例では同期した PWM 出力に GPT チャンネルを使用します。この回路の動作中は設定変更や周辺チャンネルの使用を行うことはできません。

2.2.7 CMT チャンネル

この例では、CMT を使用します。この回路は 7 セグメント LED の更新に必要です。

2.2.8 USB チャンネル

この例では、USB 2.0 FS ホスト/ファンクションを使用します。この周辺モジュールは USB メモリからの同期構成情報とタイマ開始情報の読み出しに必要です。

2.3 ソフトウェアリソース要件

サンプルプログラムは以下の FIT モジュールを使用しています。

- r_bsp
- r_cmt_rx
- r_ether_rx
- r_ptp_rx
- r_tfat_rx
- r_usb_basic
- r_usb_hmsc

2.4 サポートされているツールチェーン

サンプルプログラムは次のツールチェーンでテストと動作確認を行っています。

- Renesas RX Toolchain v2.06.00

2.5 ヘッダファイル

すべての関数呼び出しは、ドライバのプロジェクトコードとともに提供されているヘッダファイル sample_main.h、led_7seg.h、sync_if.h、file_if.h、r_data_file.h、r_elc_rx_if.h、r_mtu3_rx_if.h、r_gpt_rx_if.h、led.h のうち 1 個のファイルをインクルードすることで行われます。

2.6 整数型

このプロジェクトでは ANSI C99 を使用しています。整数型は stdint.h で定義されています。

2.7 コンパイル時の設定

サンプルプログラムのコンフィグレーションオプションは sample_main.h と r_data_file.h で設定します。オプション名と設定値を以下の表に記載します。

構成設定	
#define LINK_CH - Default value = 1	最初にリンクするイーサネットコントローラのチャンネルを設定します。 <ul style="list-style-type: none"> • 0 を設定すると、CH0 を選択します。 • 1 を設定すると、CH1 を選択します。
#define NUM_CH - Default value = 1	使用するイーサネットチャンネル数を設定します。 <ul style="list-style-type: none"> • RX64M/71M では、1 か 2 を設定してください。 • クロックが BC または TC の時には 2 を設定してください。
#define USE_7SEG_LED - defined	7 セグメント LED ¹ の使用の有無を設定します。 <ul style="list-style-type: none"> • 「USE_7SEG_LED」が定義している場合、7 セグメント LED を使用します。 • 「USE_7SEG_LED」が定義しない場合、7 セグメント LED は使用しません。
#define PARAM_FILE - Default value "PARAM.txt"	同期構成情報を格納したパラメータファイルの名前と拡張子を指定します。

構成設定	
#define FILESIZE - Default value = 2048	FAT ファイルシステムのデータバッファサイズを設定します。 RX64M/71M では 2048 (デフォルト値) を設定します。

¹7セグメントLEDはHMI拡張ボードに実装しています[7]。

2.8 データ構造

サンプルプログラムの関数で使用するデータ構造について説明します。これらの構造体は sample_main.h、sync_if.h、sync_if.c にプロトタイプ宣言と共に定義しています。

```
/* Ether & USB access state */
typedef enum
{
    APL_READ = 0, /* USB memory read state */
    APL_COM,      /* Ether communication state */
    APL_STOP,     /* Operation stop state */
} APLState;
```

```
/* Synchronous configuration structure */
typedef struct {
    uint8_t id;
    uint8_t mac[2][6];
    uint8_t ip[2][4];
    uint8_t mode;
    uint8_t ms[2];
    uint8_t sync[2];
} SyncConfig;
```

```
/* Timer configuration structure */
typedef struct {
    uint8_t tmr;
    uint8_t ch;
    uint8_t edge;
    uint32_t time_u;
    uint32_t time_l;
} PulseConfig;
```

2.9 戻り値

サンプルプログラムの関数の戻り値を示します。これらの戻り値は sync_if.h、file_if.h、r_elc_rx_if.h、r_mtu3_rx_if.h、r_gpt_rx_if.h、led_7seg.h にプロトタイプ宣言と共に定義しています。

```
/* PTP synchronous operation return value */
typedef enum
{
    SYNCIF_TOUT = -3, /* Timeout error */
    SYNCIF_ETHERR = -2, /* Standard Ether error */
    SYNCIF_ERR = -1, /* General error */
    SYNCIF_OK = 0,
} syncif_t;
```

```
/* File access return value */
typedef enum
{
    FLIF_ERR = -1, /* General error */
    FLIF_OK = 0,
} flif_t;
```

```
/* ELC driver return value */  
ELC_OK (0) /* No error */  
ELC_ERROR (-1) /* General error */
```

```
/* MTU3 driver return value */  
MTU3_OK (0) /* No error */  
MTU3_ERROR (-1) /* General error */
```

```
/* GPT driver return value */  
GPT_OK (0) /* No error */  
GPT_ERROR (-1) /* General error */
```

```
/* 7segment LED driver return value */  
LED_7SEG_OK (0) /* No error */  
LED_7SEG_ERR (-1) /* General error */
```

3. サンプルプログラムの仕様

3.1 関数の概要

サンプルプログラムの関数を表 3.1に示します。

表3.1 サンプルプログラムの関数

関数	内容
main()	このプロジェクトのメイン処理
led_init()	ユーザ LED の初期化
led_ctrl()	ユーザ LED 表示の更新
usb_memory_start()	USB メモリタスクの開始
Sample_Task()	サンプルアプリケーションのタスク処理
PTPCom ()	PTP 時刻同期
SyncErr ()	PTP 時刻同期、またはイーサネットのエラーから復帰
ReadPTPMsg()	PTP メッセージの読み出し。Announce メッセージの受信時には、マスタの port identity を更新します。
led_callback()	CMT による 7セグメント LED 更新用のコールバック関数
file_read()	ファイル読み出し (USB メモリからの同期構成情報とタイマ開始情報の読み出し)
file_stop()	ファイル操作の終了 (USB メモリをアンマウント)
file_err()	ファイル操作でのエラーから復帰 (USB メモリをアンマウント)
get_sync_config()	同期構成情報の取得
get_pulse_time()	タイマ開始情報の取得
R_LED_Open ()	7セグメント LED の初期化と開始
R_LED_UpdTime()	7セグメント LED 表示データの更新
rx64m_led_cmt_cyclic_isr()	7セグメント LED 割り込みハンドラ (7セグメント LED へのデータ出力)

3.2 環境とプログラムの実行

サンプルプログラムは、2台以上のRX64M/71M RSKボード¹ (マスタノードと1個以上のスレーブノード)、イーサネットハブ (以後、ハブ)、イーサネットケーブル、USBメモリ、オシロスコープを使用します。個々のRX64M/71M RSKボードのPWM出力端子はオシロスコープの入力に接続します。

実行手順の概要を以下に説明します。

- 全てのRX64M/71M RSKボード (以後、RSKボード) にプロジェクトの実行コードを書き込んでください。
- 全RSKボードを互いにイーサネットケーブルによりハブ経由で接続してください。
- 個々のRSKボードのPWM出力端子をオシロスコープに接続してください。
- USBメモリを全RSKボードのUSBポートに接続してください。USBメモリは同期構成情報とタイマ開始情報を格納します
- 全RSKボードに電源を投入してください。
- 個々のクロック (RSKボード) は同期構成情報とタイマ開始情報をUSBメモリから読み出し、互いに同期を開始します。
- USBメモリの内容はパラメータ格納ファイル (PARAM.txt) とタイマ開始時刻ファイル (TEST_1.txt) で構成します。パラメータ格納ファイルは同期構成情報を格納し、評価開始前にUSBメモリのルートディレクトリに置いておく必要があります。パラメータ格納ファイルとタイマ開始時刻ファイルのサンプル^{2,3}はプロジェクトに準備しています。タイマ開始時刻ファイルのタイマ開始情報の形式は、ナノ秒単位の値の上位と下位、各32ビット (TMSTTRUmとTMSTTRLm)⁴です。
- オプションのHMI拡張ボード[7]が実装されている場合は、7セグメントLEDにタイマ開始時刻が表示されます。表示はナノ秒単位の開始時刻の下位32ビット (TMSTTRLm) です。
- 個々のクロックは、PTPで同期したローカルクロックカウンタが特定の時刻に一致したときに、ELC経由でMTU3/GPTタイマを開始します。
- 出力されるPWM波形の互いの位相差を確認することができます。

¹ 製品名は Renesas Starter Kit+ for RX64M [8] または Renesas Starter Kit+ for RX71M [9]です。

² demo_src/usb_memory_sample/param/ID=X,,YPARAM.txt。1.6節を参照ください。

³ これらファイルのMACアドレスのベンダIDフィールドには、ルネサスベンダID (74-90-50)、固有IDフィールドには、このサンプル固有の値を設定しています。お客様のシステムに適用する場合、必ずこの値を変更してください。

⁴ 識別子の”m”は0~5でEPTPCのタイマチャンネルを示します。

図3.1は3台のRSKボードを使用する構成を示しています。

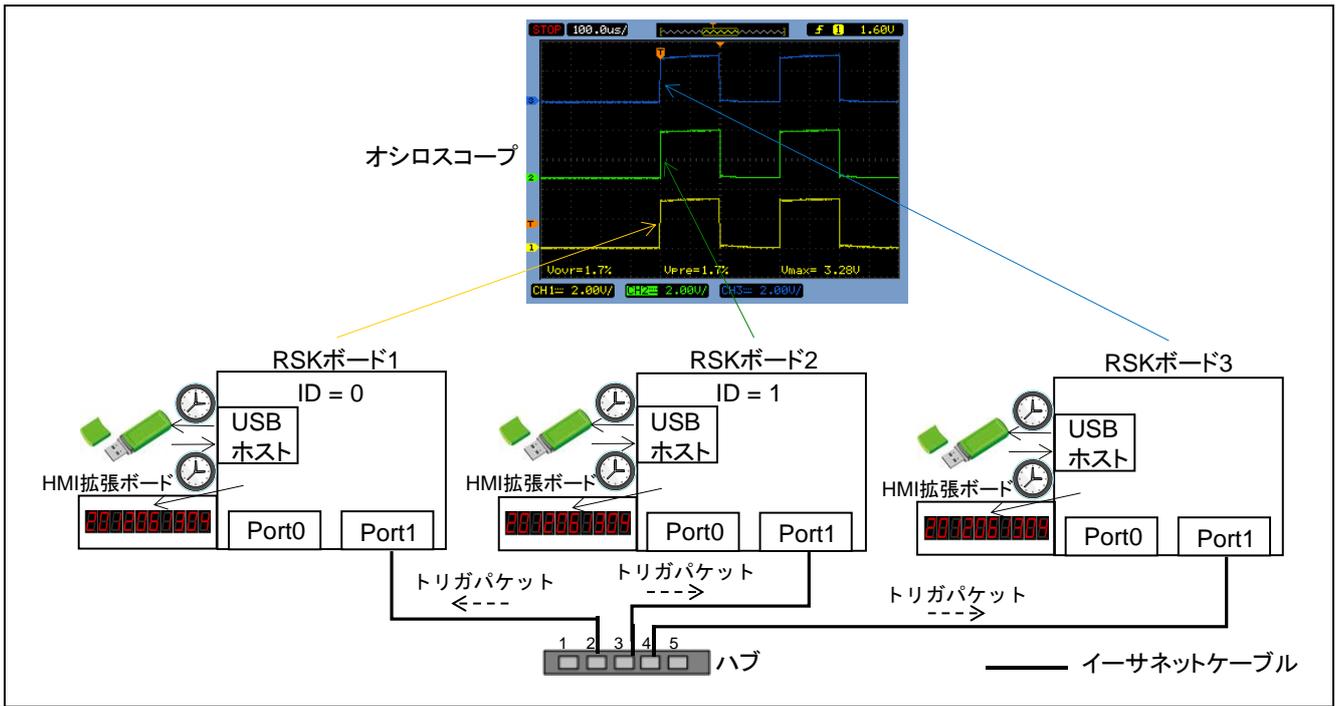


図3.1 実行環境 (3 ボード構成)

図 3.2と図 3.3はそれぞれサンプルプログラム動作の概要と USB メモリの内容の一例を示しています。図 3.4 は OC で port1 を使用した場合の 3 台のボードのパラメータ格納ファイルの例、図 3.5はタイマ開始時刻ファイルの例を示しています。

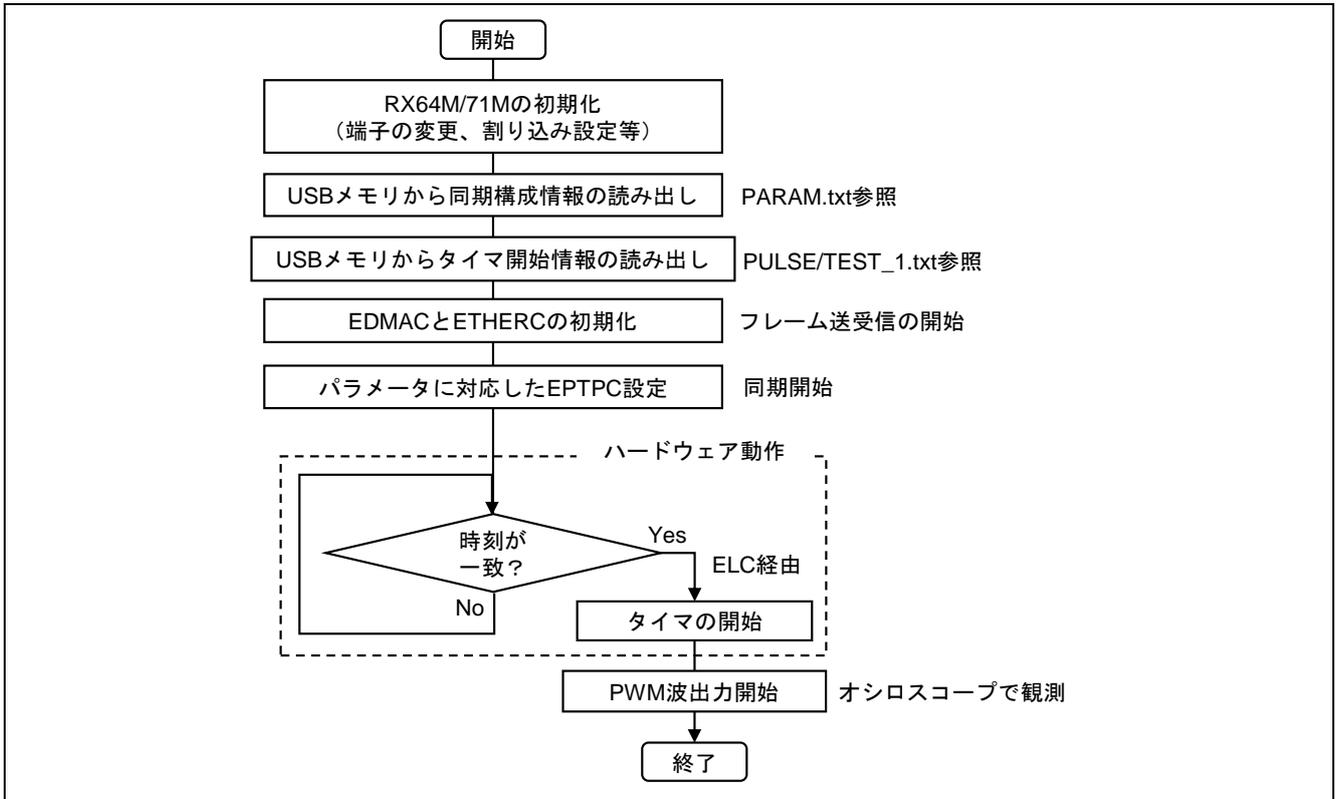


図3.2 ソフトウェア動作の概要

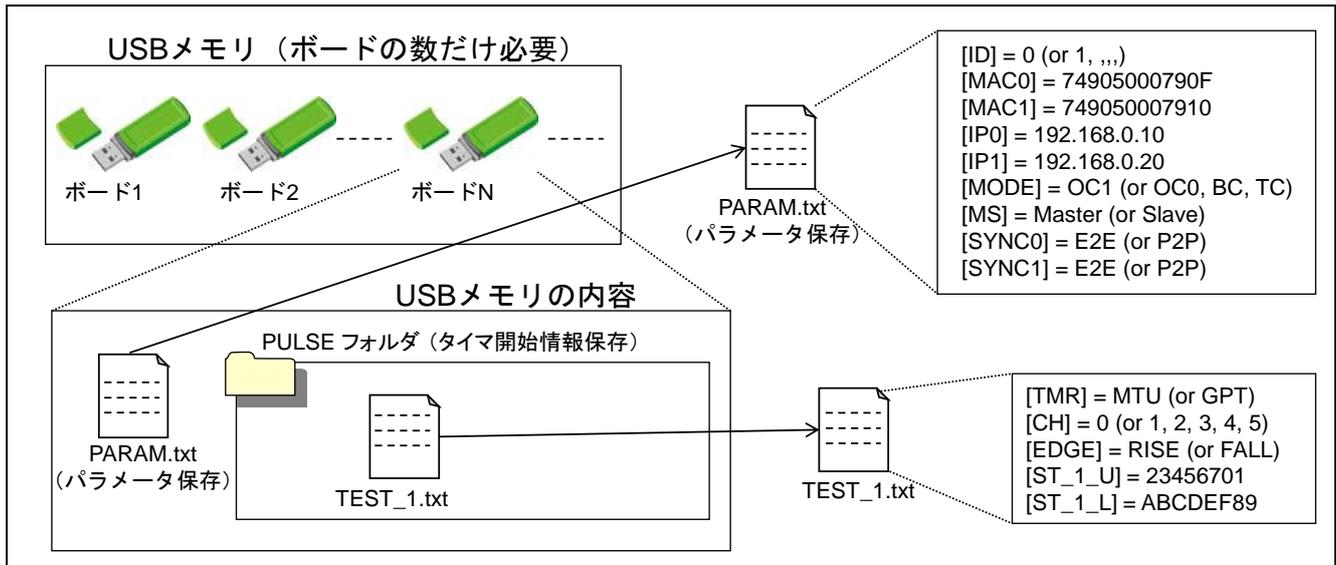


図3.3 USBメモリ内容の例

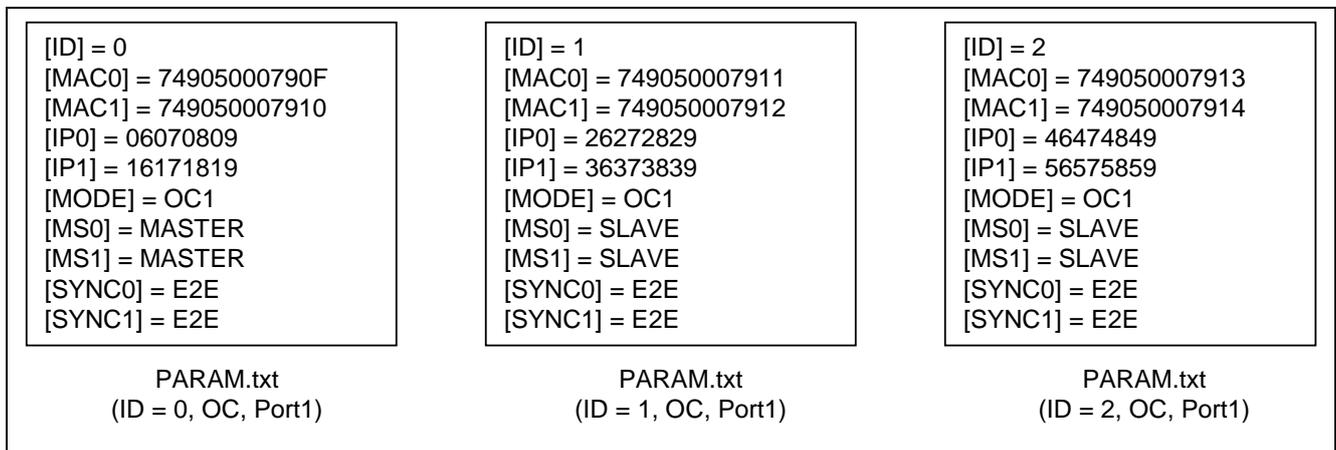


図3.4 パラメータ格納ファイルの例

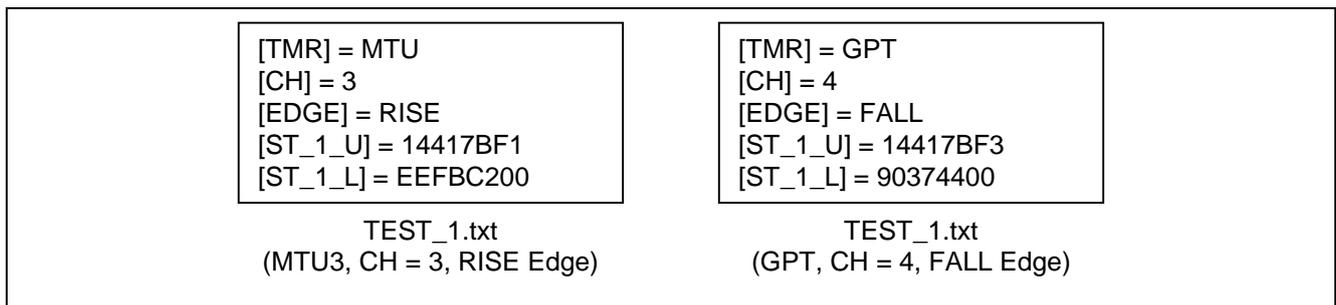


図3.5 タイマ開始時刻ファイルの例

3.3 ボードの設定

サンプルプログラムの動作にはRX64M/71M RSK ボードのジャンパをデフォルト設定から変更する必要があります。イーサネットPHYにアクセスするチャンネルはソフトウェア設定に合わせ、無矛盾に設定してください。USBにアクセスするチャンネルとPWM出力端子設定はボードのデフォルト設定から変更する必要があります。RX64M/71M RSK ボードの型名がR0K50564MC001BRまたはR0K5RX71MC010BRの場合、ジャンパ設定を図3.6に示します。また、RX71M RSK ボード型名がR0K50571MC000BRの場合、ジャンパ設定を図3.7に示します。

- イーサネットPHY アクセスの設定			
ジャンパ	LINK_CH = 1 (デフォルト設定)	LINK_CH = 0	使用機能
J3	2-3	1-2	ETHERC ET0MDIO または ET1MDIO
J4	2-3	1-2	ETHERC ET0MDC または ET1MDC

- USB アクセスの設定			
ジャンパ	ボードのデフォルト設定	サンプル動作の設定	使用機能
J2	2-3	1-2	USBでホストモードを有効化
J6	1-2	2-3	USB USB0VBUSEN

- PWM 出力端子の設定			
ジャンパ	ボードのデフォルト設定	サンプル動作の設定	使用機能
J15	1-2	2-3	MTU3 MTIOC0C (CH0)

図3.6 ジャンパ設定

- イーサネットPHY アクセスの設定			
ジャンパ	LINK_CH = 1 (デフォルト設定)	LINK_CH = 0	使用機能
J13	2-3	1-2	ETHERC ET0MDIO または ET1MDIO
J9	2-3	1-2	ETHERC ET0MDC または ET1MDC

- USB アクセスの設定			
ジャンパ	ボードのデフォルト設定	サンプル動作の設定	使用機能
J1	2-3	1-2	USBでホストモードを有効化
J3	1-2	2-3	USB USB0VBUSEN

- PWM 出力端子の設定			
ジャンパ	ボードのデフォルト設定	サンプル動作の設定	使用機能
J15	1-2	2-3	MTU3 MTIOC0C (CH0)

図3.7 ジャンパ設定

RX64M/71M RSK ボードのPWM出力端子をオシロスコープの端子に接続してください。PWM波形を出力するボード上の端子を図3.8に示します。

アプリケーションヘッダ	ピン番号	ヘッダ名	MCU端子	タイマのPWM出力
JA2	23	IRQ2/M1_EncZ/M1_H SIN2	29	MTU3 MTIOC0C (CH0)
JA3	18	D1	156	GPT GTIOC1A (CH1)

図3.8 PWM出力端子

3.4 HMI 拡張ボードの設定

7セグメントLEDを使用するには、HMI 拡張ボード上の一部の0Ω抵抗をデフォルトの設定から変更する必要があります。デフォルト設定からの変更部分を図3.9に示します。

抵抗	デフォルト設定	デモの設定	使用機能
R387	実装	なし	7セグメントLED
R433	実装	なし	7セグメントLED
R464	実装	なし	7セグメントLED
R348	なし	実装	7セグメントLED
R404	なし	実装	7セグメントLED
R429	なし	実装	7セグメントLED
R445	なし	実装	7セグメントLED

図3.9 抵抗の設定（7セグメントLED 使用時）

3.5 測定結果の例

参考データとして、3 台の RSK ボードを使用した場合の測定結果を以下に示します。

これらの測定結果は測定条件と環境により異なることに注意してください。

3.5.1 測定条件

- トポロジー
3 台の RSK ボードによる OC E2E の構成を適用した。
- 同期モード
STCA の機能による傾き補正を行う（モード 2）同期モードを適用した。（推奨設定）
- PWM 出力タイマ
MTU3 のチャンネル 3 と GPT のチャンネル 4 を使用した。
- PTP コマンドの間隔
PTP コマンドの間隔は 1 秒*とした。

¹ Sync と Delay_Req メッセージの送信間隔が 1 秒。

3.5.2 測定方法

MTU3 および GPT が PWM 出力タイマとして使用し、PWM 出力波形をオシロスコープにより観測することで、PWM 波形の位相差を確認した（図 3.1 の構成と同じ）。

3.5.3 測定結果

図 3.10 と図 3.11 に、それぞれ横軸は 1 目盛り 100 マイクロ秒と 100 ナノ秒での MTU3 のチャンネル 3 と GPT のチャンネル 4 で出力した PWM 波形を示す。また、黄、緑、青のラインがそれぞれマスタ、スレーブ 1、スレーブ 2 の出力を示す。PWM 波形の位相差は約 100 ナノ秒となった。

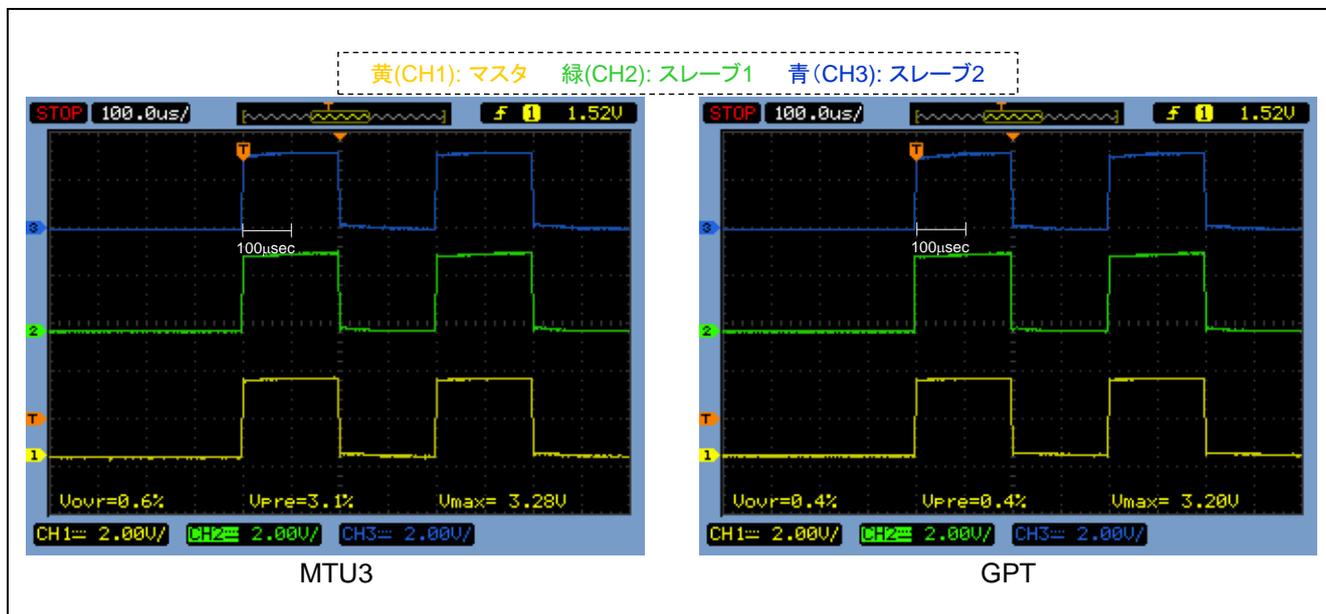


図3.10 PWM 波形の例 (100 マイクロ秒単位)

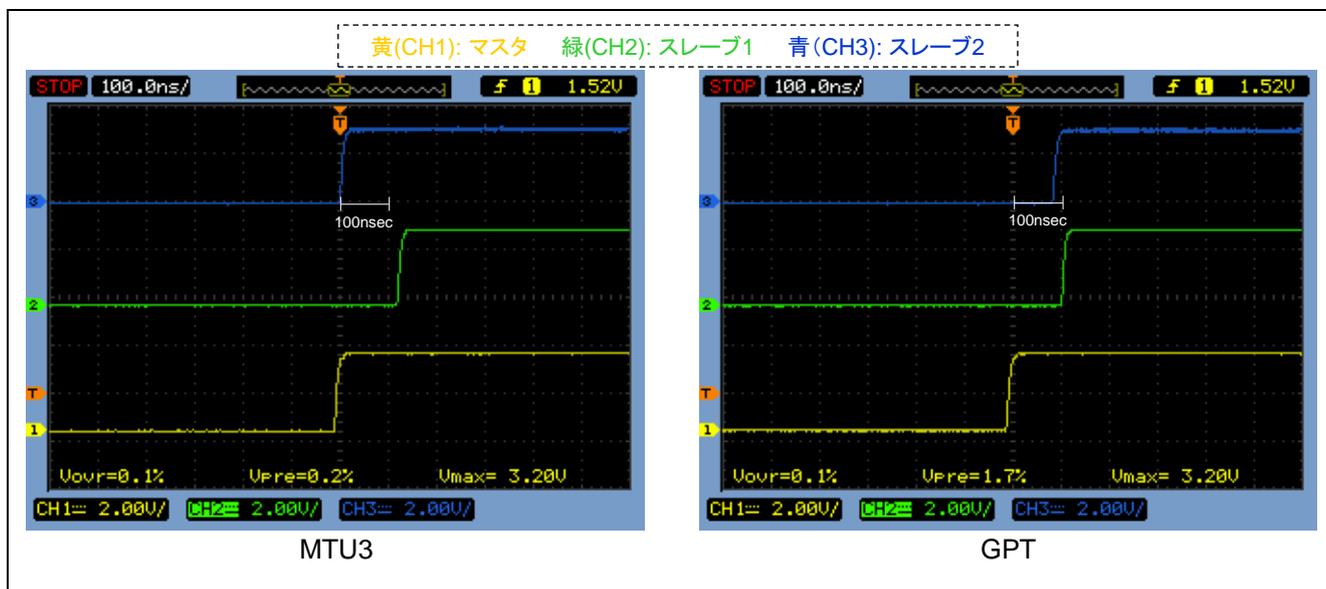


図3.11 PWM 波形の例 (100 ナノ秒単位)

4. 参考資料

ユーザーズマニュアル: ハードウェア

RX64M グループユーザーズマニュアル ハードウェア編 Rev.1.10 (R01UH0377JJ)

RX71M グループユーザーズマニュアル ハードウェア編 Rev.1.00 (R01UH0493JJ)

最新版はルネサス エレクトロニクスのウェブサイトからダウンロードできます。

ユーザーズマニュアル: ソフトウェア

RX ファミリ RXv2 命令セットアーキテクチャ ユーザーズマニュアル ソフトウェア編 (R01US0071JJ)

最新版はルネサス エレクトロニクスのウェブサイトからダウンロードできます。

技術情報/ニュース

最新版はルネサス エレクトロニクスのウェブサイトからダウンロードできます。

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	2014.07.10	—	初版発行
1.01	2014.12.12	—	オプションの7セグメントLED機能の追加
1.02	2014.12.31	—	PTPドライバ Rev.1.02 対応とファイル構成の変更
1.10	2016.03.31	—	PTPドライバ Rev.1.10 対応とファイルアクセスインタフェースの変更
1.11	2016.11.11	—	PTPドライバ Rev.1.12 とイーサネットドライバ Rev.1.12 を適用
		14	PWM出力端子設定の説明の追加と修正
1.12	2017.03.31	—	PTPドライバ Rev.1.13 を適用
		—	ELCのタイマ開始接続の設定を修正

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

- 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。
 - 当社製品、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、全部または一部を問わず、改造、改変、複製、その他の不適切に使用しないでください。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。
 - 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。
標準水準： コンピュータ、OA機器、通信機器、計測機器、AV機器、
家電、工作機械、パーソナル機器、産業用ロボット等
高品質水準： 輸送機器（自動車、電車、船舶等）、交通制御（信号）、大規模通信機器、
金融端末基幹システム、各種安全制御装置等
当社製品は、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、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損害が生じて、当社は一切その責任を負いません。
 - 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。また、当社製品および技術を、(1)核兵器、化学兵器、生物兵器等の大量破壊兵器およびこれらを運搬することができるミサイル（無人航空機を含みます。）の開発、設計、製造、使用もしくは貯蔵等の目的、(2)通常兵器の開発、設計、製造または使用の目的、または(3)その他の国際的な平和および安全の維持の妨げとなる目的で、自ら使用せず、かつ、第三者に使用、販売、譲渡、輸出、賃貸もしくは使用許諾しないでください。
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。
 - お客様の転売、貸与等により、本書（本ご注意書きを含みます。）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、当社は一切その責任を負わず、お客様にかかる使用に基づく当社への請求につき当社を免責いただきます。
 - 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。
 - 本資料に記載された情報または当社製品に関し、ご不明点がある場合には、当社営業にお問い合わせください。
- 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。

(Rev.3.0-1 2016.11)



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（豊洲フォレストシア）

■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：<https://www.renesas.com/contact/>