

RZ/T2ME グループ

IEEE 1588-2019 サンプルアプリケーション

要旨

このドキュメントでは Renesas Starter Kit+ for RZ/T2ME (RZ/T2ME RSK)(*)上で IEEE1588-2019(PTPv2)を実現するサンプルアプリケーションについて説明します。

PTPv2 による時刻同期動作の実現には、OSS の Linux PTP を改修したモジュールと、RZ/T2ME RSK に備わっている Ethernet Switch(ETHSW)モジュールの機能を使用しています。

本サンプルアプリケーションは RZ/T2ME RSK で PTPv2 を実現したものとなります。

【注】 *:以降 RZ/T2ME RSK と記載

動作確認デバイス

RZ/T2ME

目次

1. 特徴	3
1.1 IEEE1588 2019(PTPv2)	3
2. パッケージ概要	4
2.1 PTPv2 サンプルアプリケーション	4
2.1.1 開発環境	4
2.1.2 アプリケーション動作概要	5
2.1.2.1 PTPv2 動作仕様	6
2.1.3 アプリケーション動作確認内容	7
2.1.3.1 ログ出力内容	7
2.1.3.1.1 PTPv2 ログメッセージ	8
2.1.3.2 ピン出力内容	9
2.1.3.2.1 PTPv2 動作確認	9
2.1.3.3 ミラーポートによる PTPv2 パケットキャプチャ内容	11
2.1.3.3.1 PTPv2 パケットキャプチャログ	12
3. 事前準備	13
3.1 ハードウェア要件	13
3.1.1 RZ/T2ME RSK のハードウェア設定	13
3.1.1.1 ブートモードを変更する	14
3.1.1.2 Ethernet のハードウェア設定	15
3.1.1.3 外部エミュレータの有効/無効	16
3.1.2 RZ/T2ME RSK 同士を接続する	17

3.1.3	RZ/T2ME RSK と接続する	18
3.1.3.1	EWARM 使用時.....	18
3.1.3.2	e ² studio 使用時.....	19
3.2	ソフトウェア要件.....	20
3.2.1	RZ/T2ME RSK のハードウェア設定	20
3.2.1.1	統合開発環境に関するセットアップ	20
3.2.1.1.1	FSP Smart Configurator を 統合開発環境 から起動する	20
3.2.1.1.1.1	EWARM の場合	20
3.2.1.1.1.2	e ² studio の場合	21
3.2.1.1.2	複数のデバイス接続時にダウンロード先を選択できるようにする	21
3.2.1.1.2.1	EWARM の場合	21
3.2.1.1.2.2	e ² studio の場合	22
3.2.2	サードパーティーソフトウェア.....	23
3.2.2.1	シリアルターミナルツール.....	23
4.	サンプルアプリケーションの実行	24
4.1	ビルドとダウンロード.....	24
4.1.1	プロジェクトを開く	24
4.1.1.1	EWARM を使用する場合	24
4.1.1.2	e ² studio を使用する場合	24
4.1.2	デバイスの事前設定.....	24
4.1.2.1	MAC アドレス変更方法	25
4.1.2.2	IP アドレス変更方法.....	25
4.1.2.3	BMCA で使用するクロック情報.....	26
4.1.2.4	伝送遅延時間の計測方法	27
4.1.2.5	時刻配信の方式.....	27
4.1.2.6	トランスポート設定.....	28
4.1.3	ビルドと RZ/T2ME RSK へのダウンロード	28
4.1.3.1	プロジェクトソースの生成.....	28
4.1.3.2	ビルドとダウンロード	28
4.1.3.2.1	EWARM の場合.....	28
4.1.3.2.2	e ² studio 使用時.....	28
4.2	サンプルアプリケーションの起動	29
4.2.1	EWARM 使用時.....	29
4.2.2	e ² studio 使用時	29
	改訂記録	30

1. 特徴

PTP(Precision Time Protocol)とは、同一ネットワーク上でクロック同期させるために使用される通信プロトコルを指します。

本アプリケーションにおける PTP は、以下の要素によって構成されます。

- IEEE1588-2019(PTPv2)
 - 時刻同期を実現します

1.1 IEEE1588 2019(PTPv2)

IEEE1588-2019(PTPv2)による時刻同期は以下のように実現されます。

- 各デバイスは最適なクロックを選択するアルゴリズム(best master clock algorithm(BMCA))を実施します
- 最適なクロックを保持するデバイスは"timeTransmitter"、それ以外のデバイスは"timeReceiver"となります
- timeTransmitter となったデバイスは時刻情報とクロック情報の発信源となります
- timeReceiver となったデバイスは timeTransmitter から受信した時刻情報と、timeTransmitter 間のネットワーク伝送遅延時間をもとに自身の時刻情報を調整することで、timeTransmitter の時刻と同期します

PTPv2 には以下の特徴があります:

- 高精度な時刻同期
 - マイクロ秒以下の高精度時刻同期が可能です
- 伝搬遅延測定
 - 時刻情報の送受信にかかる伝送時間を計算するため、通信経路による情報伝達速度の異なる状況下でも正確な時刻同期が可能です
- 自動選択機能
 - ネットワーク内の PTPv2 対応デバイスの中から、最もクロック特性の優先度が高いデバイスを自動で timeTransmitter に設定することが可能です

2.1.2 アプリケーション動作概要

各 RZ/T2ME RSK の ETH ポートを接続することで PTPv2 が開始されます。

PTPv2 が開始されると、各 RZ/T2ME RSK は自身のクロック情報を含む ANNOUNCE メッセージを送信します。

ANNOUNCE メッセージ受信時、メッセージに含まれる Domain Number が同じであれば、

BMCA による timeTransmitter/timeReceiver の判定が行われます。

BMCA による判定時、クロック情報(*)を以下の順序で評価します。

値が小さいほど、優先度が高くなり、値が同一である場合は次の評価に移行します。

(*) クロック情報の設定方法に関しては”4.1.2.3 BMCA で使用するクロック情報”を参照してください。

1. Priority1
2. Clock Class
3. Clock Accuracy
4. Clock Variance
5. Priority2
6. 1-5 の値で評価できなかった場合、MAC Address で評価

優先度が最も高いクロックを保持するデバイスは”timeTransmitter”、それ以外のデバイスは”timeReceiver”となります。

本アプリケーションはネットワーク伝送遅延時間として、E2E(End to End Mechanism)と P2P(Peer to Peer Mechanism)の2種類が存在します。

- E2E では、timeReceiver が timeTransmitter に対して DELAY REQUEST メッセージを送信します。
timeTransmitter は DELAY REQUEST メッセージ受信時間を含めた DELAY RESPONSE メッセージを応答として送信します。
timeReceiver は自身の DELAY REQUEST メッセージ送信時間と timeTransmitter の DELAY RESPONSE メッセージ受信時間から伝送遅延時間を算出します。
- P2P では、timeTransmitter と timeReceiver の双方が PDELAY REQUEST メッセージを送信します。
PDELAY REQ メッセージ受信時、PDELAY REQUEST メッセージの受信時間を含めた PDELAY RESPONSE メッセージを応答として送信します。
その後、PDELAY RESPONSE メッセージの送信時刻を含めた PDELAY RESPONSE FOLLOW UP メッセージを送信します。
timeTransmitter と timeReceiver の双方は PDELAY REQUEST、PDELAY RESPONSE メッセージの送受信時間から、伝送遅延時間を算出します。
ネットワーク伝送遅延時間の設定方法に関しては”4.1.2.4 伝送遅延時間の計測方法”を参照してください。

本アプリケーションは時刻配信の方式として、one step clock と two step clock の2種類が存在します。

- one step clock では、timeTransmitter は自身の時刻情報とメッセージの送信時間を含めた SYNC メッセージを送信します。
timeReceiver は SYNC メッセージに含まれる時刻情報と timeTransmitter の送信時間から、時刻同期を行います。
- two step clock では、timeTransmitter は自身の時刻情報を含めた SYNC メッセージを送信します。
その後、SYNC メッセージの送信時間を含めた FOLLOW UP メッセージを送信します。
timeReceiver は SYNC メッセージに含まれる時刻情報と FOLLOW UP メッセージに含まれる SYNC の送信時間から、時刻同期を行います。
時刻配信の方式の設定方法に関しては”4.1.2.5 時刻配信の方式”を参照してください。

本アプリケーションは PTP 通信時のフレームとして、イーサネットフレームもしくは UDP フレームを指定できます。

フレームの設定方法に関しては”4.1.2.6 トランスポート設定”を参照してください。

なお、本サンプルアプリケーションでは動作確認用の機能として以下の動作が実装されています。

- UART によるログ出力によって動作状態を表示します。
- GPIO ピンを用いて PTPv2 動作確認用クロック波形の出力を行います。

2.1.2.1 PTPv2 動作仕様

PTPv2 の動作仕様は以下の通りです。

表 2-2 PTPv2 仕様

SYNC 送信間隔	DELAY 送信間隔	PDELAY 送信間隔
1sec	1sec 以内 (*)	1sec

*: DELAY 送信間隔は 1sec 以内のランダムな値となる

2.1.3 アプリケーション動作確認内容

各種出力によって確認できる内容を以下に示します。

2.1.3.1 ログ出力内容

本アプリケーションは、RZ/T2ME RSK の USB to Serial Port(CN16) から UART 経由でログメッセージを出力します。

ログメッセージを通してアプリケーションの動作状況を確認することが可能です。

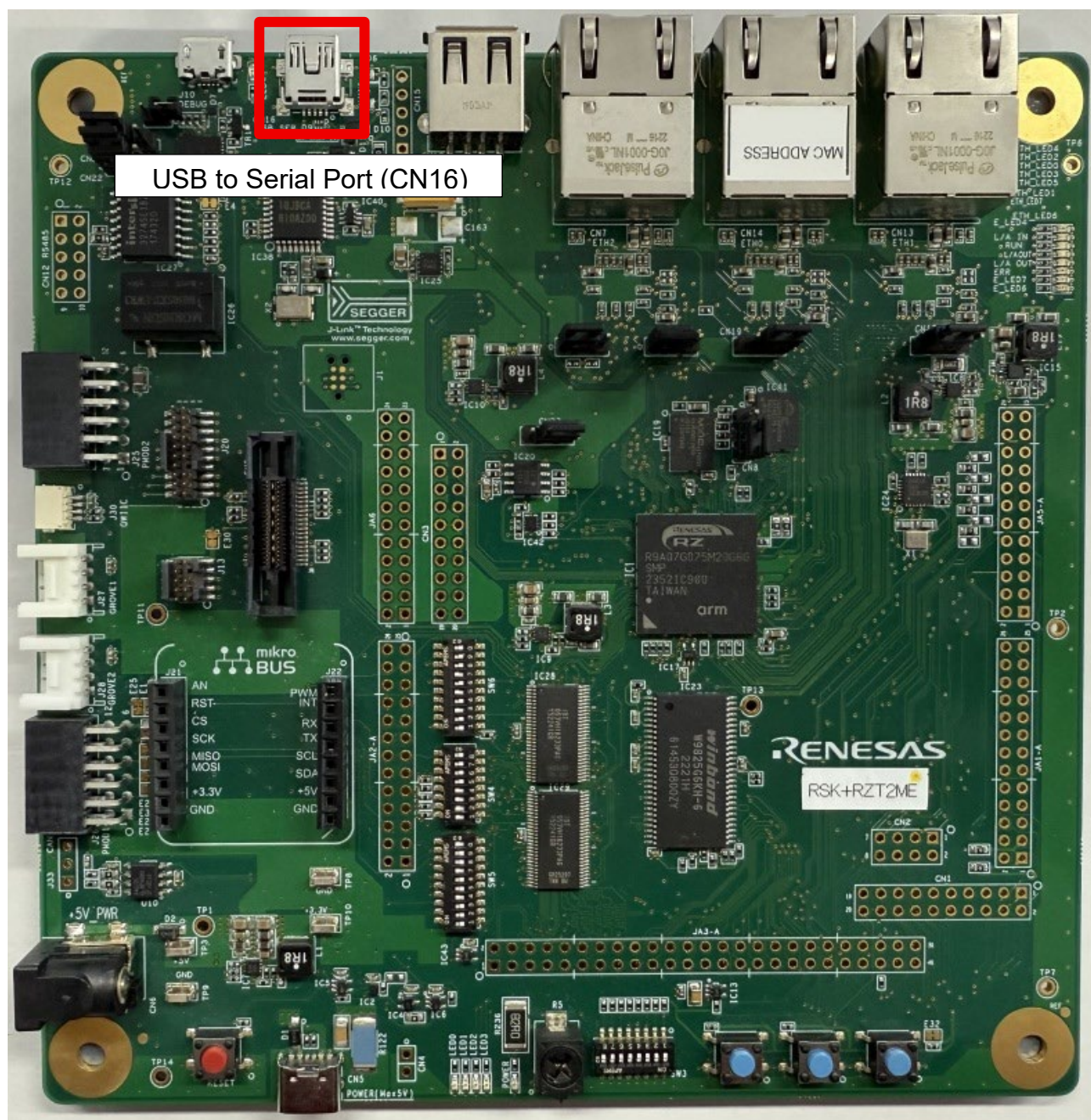


図 2-1 RZ/T2ME RSK board layout(USB to Serial Port USB Connector)

ログメッセージによって以下の内容が確認できます。

- PTPv2 動作状況

2.1.3.1.1 PTPv2 ログメッセージ

PTPv2 関連のログメッセージは以下の通りです。

```
>>Info : Initialized Ethernet network interface.
>>Info : Started Ethernet network interface.
>>Info : Initialized lwIP TCP/IP stack.
>>Info : Started lwIP TCP/IP stack tasks./** Started sample application (v1.0) for lwIP Port. **/
>>Info : Set the callback function into TCP/IP stack.
>>Info : Created sample application task.
>>Info : PTP notice AS_CAPABLE_UP: domainNumber=0   portIndex=0
>>Info : PTP notice NETDEV_UP: portIndex=0
>>Info : PTP notice GM_CHANGE: gmIdentity=11 22 33 ff fe 44 55 66
```

図 2-2 timeTransmitter 側の PTPv2 ログメッセージ

```
>>Info : Initialized Ethernet network interface.
>>Info : Started Ethernet network interface.
>>Info : Initialized lwIP TCP/IP stack.
>>Info : Started lwIP TCP/IP stack tasks./** Started sample application (v1.0) for lwIP Port. **/
>>Info : Set the callback function into TCP/IP stack.
>>Info : Created sample application task.
>>Info : PTP notice NETDEV_UP: portIndex=0
>>Info : PTP notice AS_CAPABLE_UP: domainNumber=0   portIndex=0
>>Info : PTP notice GM_CHANGE: gmIdentity=11 22 33 ff fe 77 88 99
>>Info : PTP notice GM_CHANGE: gmIdentity=11 22 33 ff fe 44 55 66
>>Info : PTP notice DEV_SYNCD:
```

図 2-3 timeReceiver 側の PTPv2 ログメッセージ

各ログメッセージは以下の状態を示しています。

- "PTP notice NETDEV_UP: portIndex=0"
— ポート 0 がリンクアップした
- "PTP notice AS_CAPABLE_UP: domainNumber=0 portIndex=0"
— ポート 0 が Capable となった
- "PTP notice GM_CHANGE: gmIdentity=11 22 33 ff fe 44 55 66"
— timeTransmitter が変更された
- "PTP notice DEV_SYNCD:"
— timeTransmitter との時刻補正が完了し、安定した状態となった

2.1.3.2 ピン出力内容

本アプリケーションは、GPIO ピンを用いて PTPv2 動作確認用クロック波形の出力を行います。

2.1.3.2.1 PTPv2 動作確認

本アプリケーションは、PTPv2 による時刻同期動作確認用に GPIO ピン(CN2-8)へクロック波形を出力しています。

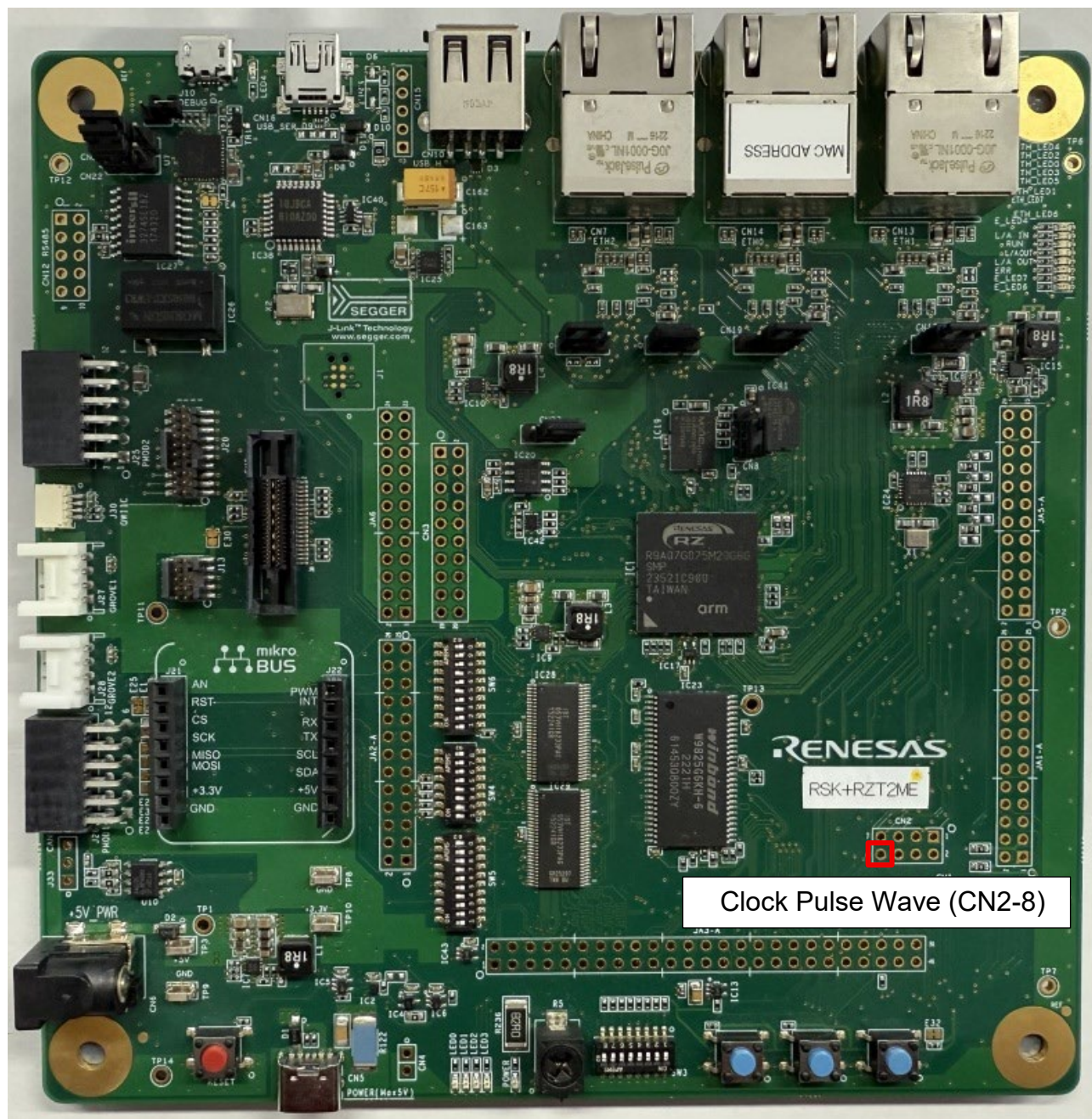


図 2-4 RZ/T2ME RSK board layout (GPIO)

【注】 クロック波形の周期は、"1[ms]"です

PTPv2 同期時におけるクロック波形を以下に示します。

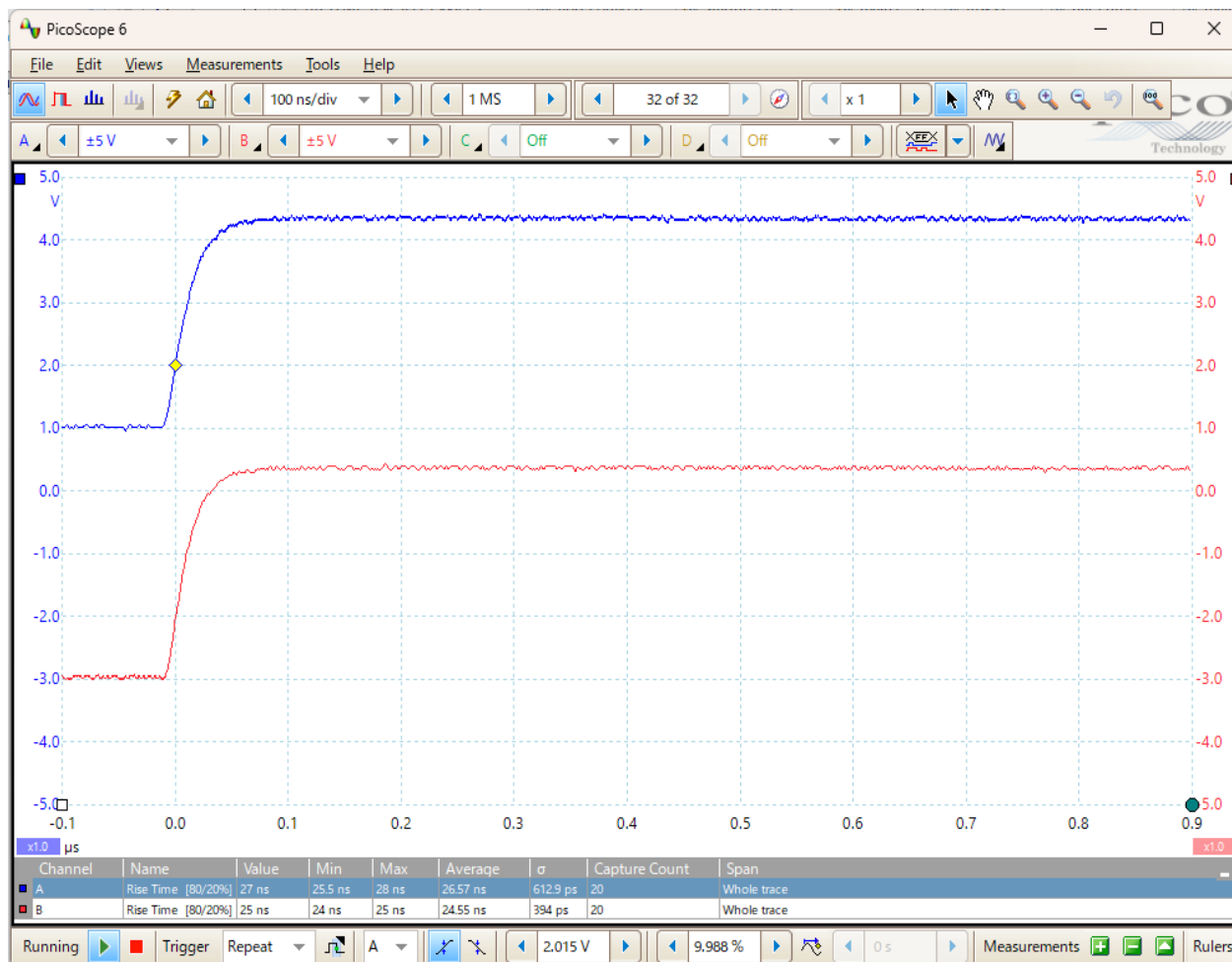


図 2-5 PTPv2 クロック同期

表 2-3 各チャネルの内容

チャネル	色	出力ピン	説明
CH1	青	GPIO (timeTransmitter)	timeTransmitter のクロック波形
CH2	赤	GPIO (timeReceiver)	timeTransmitter のクロック波形に同期するように動作する

上図に示したような各ピンの波形により、以下のことが確認できます。

- CH1 (timeTransmitter の GPIO)と CH2 (timeReceiver の GPIO)により、timeTransmitter と timeReceiver が正しく時刻同期していることが確認できます。

送受信されるパケットの種類および送信タイミングに関しては、2.1.2 アプリケーション動作概要を参照して下さい。

2.1.3.3 ミラーポートによる PTPv2 パケットキャプチャ内容

本アプリケーションは、Ethernet の 1ch/2ch いずれかを Ethernet 0ch のミラーポートに設定することができます。

ミラーポートと PC を Ethernet ケーブルで接続することで、パケットキャプチャツールにて PTPv2 のパケットのキャプチャログを出力します。

ログメッセージを通してアプリケーションの動作状況を確認することが可能です。

- ミラーポートの設定

プロジェクトのディレクトリより、"app_config.h"を開いてください。

```
/common/renesas/application/app_config.h
```

app_config.h の 48 行目より、

```
; // for debug option  
; // set mirror port to mirror eth0  
; // 0: no mirror port set  
; // 1-2:set eth(1 or 2) mirror port  
; #define MIRROR_PORT 0
```

のマクロ定義によってミラーポートの設定を切り替えることが可能です。

【注】 ミラーポートの設定を行わない場合は、0 を設定してください

【注】 Ethernet 0ch のミラーを 1ch で行う場合は、1 を設定してください

【注】 Ethernet 0ch のミラーを 2ch で行う場合は、2 を設定してください

2.1.3.3.1 PTPv2 パケットキャプチャログ

パケットキャプチャツールにて出力される PTPv2 関連のログメッセージは以下の通りです。

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	RenasasElect_44:55:66	Nearest-Bridge	PTPv2	68	Peer_Delay_Req Message
2	0.000463600	RenasasElect_77:88:99	Nearest-Bridge	PTPv2	68	Peer_Delay_Resp Message
3	0.000463600	RenasasElect_77:88:99	Nearest-Bridge	PTPv2	68	Peer_Delay_Resp_Follow_Up Message
4	0.279061000	RenasasElect_44:55:66	IEEI&MSocie_00:00:00	PTPv2	60	Sync Message
5	0.279061000	RenasasElect_44:55:66	IEEI&MSocie_00:00:00	PTPv2	60	Follow_Up Message
6	0.375058200	RenasasElect_77:88:99	Nearest-Bridge	PTPv2	68	Peer_Delay_Req Message
7	0.375531000	RenasasElect_44:55:66	Nearest-Bridge	PTPv2	68	Peer_Delay_Resp Message
8	0.375531000	RenasasElect_44:55:66	Nearest-Bridge	PTPv2	68	Peer_Delay_Resp_Follow_Up Message
9	1.000003300	RenasasElect_44:55:66	Nearest-Bridge	PTPv2	68	Peer_Delay_Req Message
10	1.000482600	RenasasElect_77:88:99	Nearest-Bridge	PTPv2	68	Peer_Delay_Resp Message
11	1.000482600	RenasasElect_77:88:99	Nearest-Bridge	PTPv2	68	Peer_Delay_Resp_Follow_Up Message
12	1.279057900	RenasasElect_44:55:66	IEEI&MSocie_00:00:00	PTPv2	60	Sync Message
13	1.279090700	RenasasElect_44:55:66	IEEI&MSocie_00:00:00	PTPv2	60	Follow_Up Message
14	1.284997700	RenasasElect_44:55:66	IEEI&MSocie_00:00:00	PTPv2	78	Announce Message
15	1.375093400	RenasasElect_77:88:99	Nearest-Bridge	PTPv2	68	Peer_Delay_Req Message
16	1.375568200	RenasasElect_44:55:66	Nearest-Bridge	PTPv2	68	Peer_Delay_Resp Message
17	1.375568200	RenasasElect_44:55:66	Nearest-Bridge	PTPv2	68	Peer_Delay_Resp_Follow_Up Message
19	2.000037000	RenasasElect_44:55:66	Nearest-Bridge	PTPv2	68	Peer_Delay_Req Message
20	2.000516900	RenasasElect_77:88:99	Nearest-Bridge	PTPv2	68	Peer_Delay_Resp Message
21	2.000516900	RenasasElect_77:88:99	Nearest-Bridge	PTPv2	68	Peer_Delay_Resp_Follow_Up Message
22	2.279120700	RenasasElect_44:55:66	IEEI&MSocie_00:00:00	PTPv2	60	Sync Message
23	2.279120700	RenasasElect_44:55:66	IEEI&MSocie_00:00:00	PTPv2	60	Follow_Up Message

図 2-6 PTPv2 パケットのキャプチャログ

各ログメッセージは以下の状態を示しています。

- "RenasasElect_77:88:99 ~ ~ Peer_Delay_Req Message"
— timeReceiver がリンク遅延測定のために timeTransmitter へ要求を送信
- "RenasasElect_44:55:66 ~ ~ Peer_Delay_Resp Message"
— timeTransmitter が受信時刻を含めて timeReceiver へ応答
- "RenasasElect_44:55:66 ~ ~ Peer_Delay_Resp_Follow_UP Message"
— timeTransmitter が正確な送信タイムスタンプを後追い通知
- "RenasasElect_44:55:66 ~ ~ Peer_Delay_Req Message"
— timeTransmitter が同じリンク遅延を測定するため timeReceiver へ要求を送信
- "RenasasElect_77:88:99 ~ ~ Peer_Delay_Resp Message"
— timeReceiver が受信時刻を含めて timeTransmitter へ応答
- "RenasasElect_77:88:99 ~ ~ Peer_Delay_Resp_Follow_UP Message"
— timeReceiver が正確な送信タイムスタンプを通知
- "RenasasElect_44:55:66 ~ ~ Sync Message"
— timeTransmitter が時刻同期の基準時刻を timeReceiver へ送信
- "RenasasElect_44:55:66 ~ ~ Follow_UP Message"
— Sync 送信時刻の正確なタイムスタンプを通知
- "RenasasElect_44:55:66 ~ ~ Announce Message"
— timeTransmitter が自身のクロック情報を通知し GM であることを宣言

本ログは一例であり、設定に応じて動作（メッセージの種類や順序）は変化します。

本例では、P2P 方式による双方向の遅延測定および two-step 方式による時刻同期（Sync/Follow_Up）が確認でき、Announce メッセージも含まれていることから、PTPv2 通信が正常に動作していることが確認できます。

3. 事前準備

本章では、サンプルアプリケーションのビルド及び実行に必要なツールを示します。

3.1 ハードウェア要件

サンプルアプリケーションの実行には以下の物が必要です。

- RZ/T2ME RSK (*1) ... 評価ボード
- I-jet (*1)(*2) ... デバッガ
- デバッガ接続用 USB ケーブル (*1)
(USB-Micro)
- シリアル通信用 USB ケーブル ... ログメッセージ取得用
(USB-Mini)
- Ethernet ケーブル ... RZ/T2ME RSK 同士の接続用 + パケットキャプチャ用
- (オプション)2 チャンネル以上のオシロスコープ

【注】 1. timeTransmitter 用と timeReceiver 用で計 2 つ用意してください
2. EWARM 使用時のみ

3.1.1 RZ/T2ME RSK のハードウェア設定

以下に変更する必要があるハードウェア設定を示します。

- ブートモードを 16 ビットバスブートモード(NOR フラッシュ)に変更する
- 外部エミュレータを有効にする(*)

【注】 *:EWARM 使用時のみ。

3.1.1.1 ブートモードを変更する

本サンプルアプリケーションをRAMで起動するために、外部フラッシュメモリがブランクである必要があります。RZ/T2ME RSKのNORフラッシュメモリはブランクであるため、動作モードを"16ビット・バス起動モード(NORフラッシュ)"に設定してください。

以下の操作が必要です:

表 3-1 ハードウェア設定(動作モードの変更)

Reference	設定
SW4-1	ON
SW4-2	OFF
SW4-3	ON

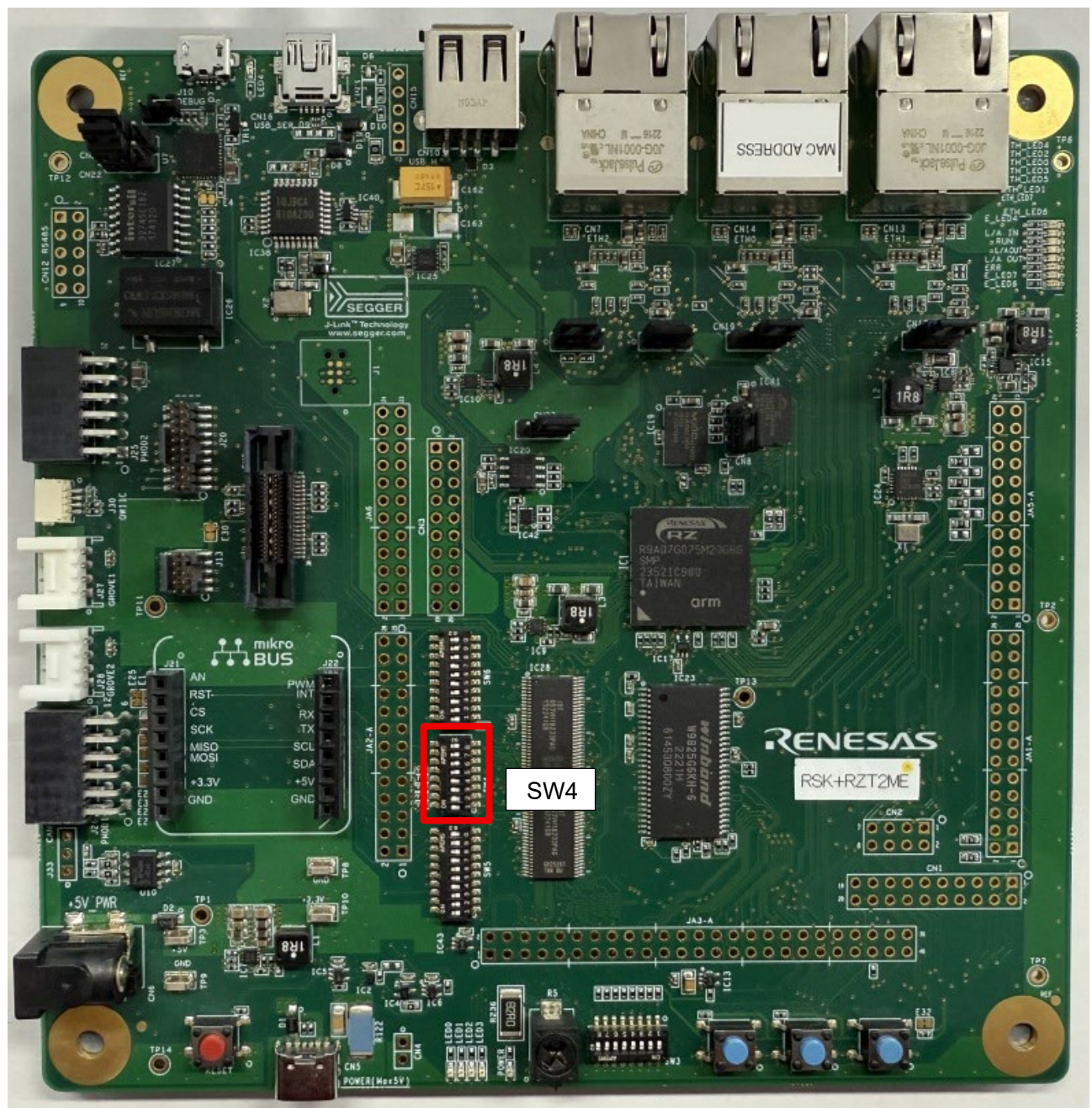


図 3-1 RZ/T2ME RSK board layout(SW4)

3.1.1.2 Ethernet のハードウェア設定

本サンプルアプリケーションは Ethernet の 0ch~2ch を使用するため、以下のハードウェア設定が必要です。

表 3-2 ハードウェア設定(Ethernet)

Reference	設定
SW6-1	OFF
CN17	2-3 Short
CN18	1-2 Short
CN19	1-2 Short
CN20	1-2 Short

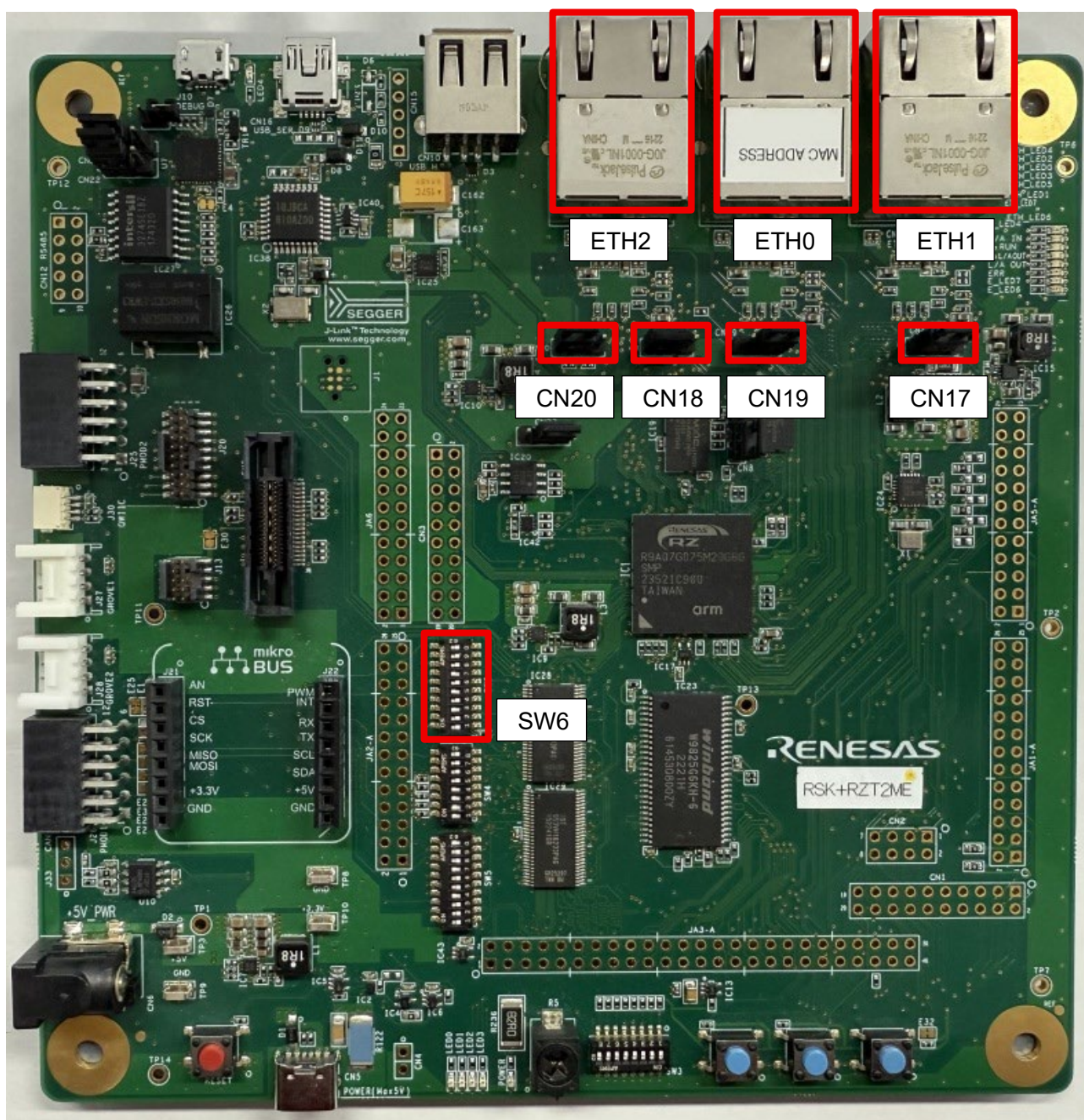


図 3-2 RZ/T2ME RSK board layout(SW6, CN17, CN18, CN19, CN20)

3.1.1.3 外部エミュレータの有効/無効

使用する統合開発環境によって、外部エミュレータの有効/無効を切り替える必要があります。

- EWARM を用いる場合
 - I-jet デバッガ(外部エミュレータ)を使用します。外部エミュレータを **有効**にしてください。
- e² studio を使用する場合
 - オンボードデバッガ(J-LinkOB)を使用します。外部エミュレータを **無効**にしてください。

外部エミュレータの有効/無効化は、RZ/T2ME RSK における J9 ジャンパによって設定します。

以下の操作が必要です:

表 3-3 ハードウェア設定(外部エミュレータの有効/無効化)

使用 IDE	外部エミュレータ設定	J9
EWARM 使用時	有効	Short
e ² studio 使用時	無効	Open

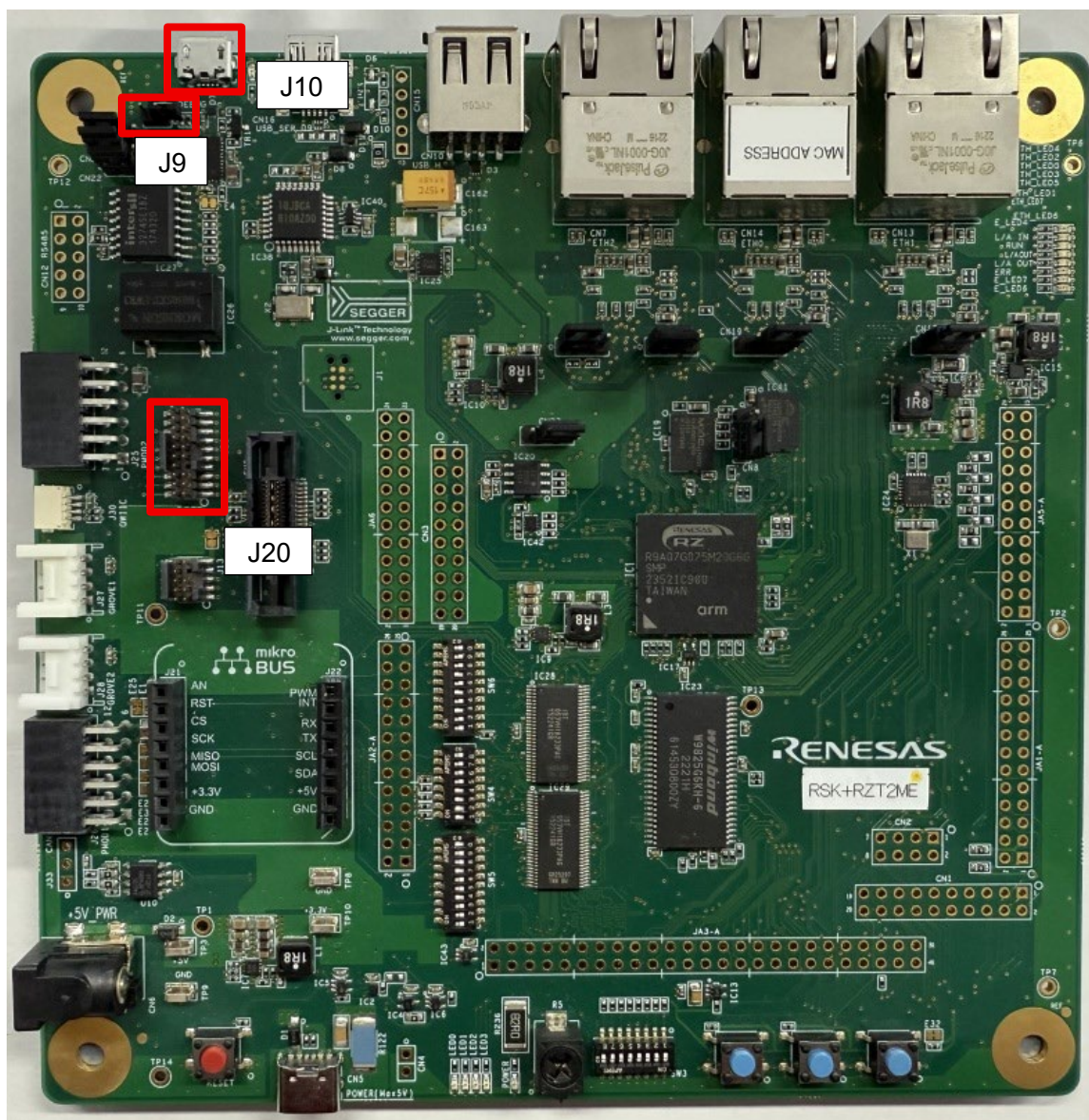


図 3-3 RZ/T2ME RSK board layout(J9, J10, J20)

3.1.2 RZ/T2ME RSK 同士を接続する

RZ/T2ME RSK 同士の接続には、Ethernet ケーブルを使用します。

timeTransmitter 用 RZ/T2ME RSK の CN7/CN13/CN14 いずれかと timeReceiver 用 RZ/T2ME RSK の CN7/CN13/CN14 いずれかを Ethernet ケーブルで接続してください。

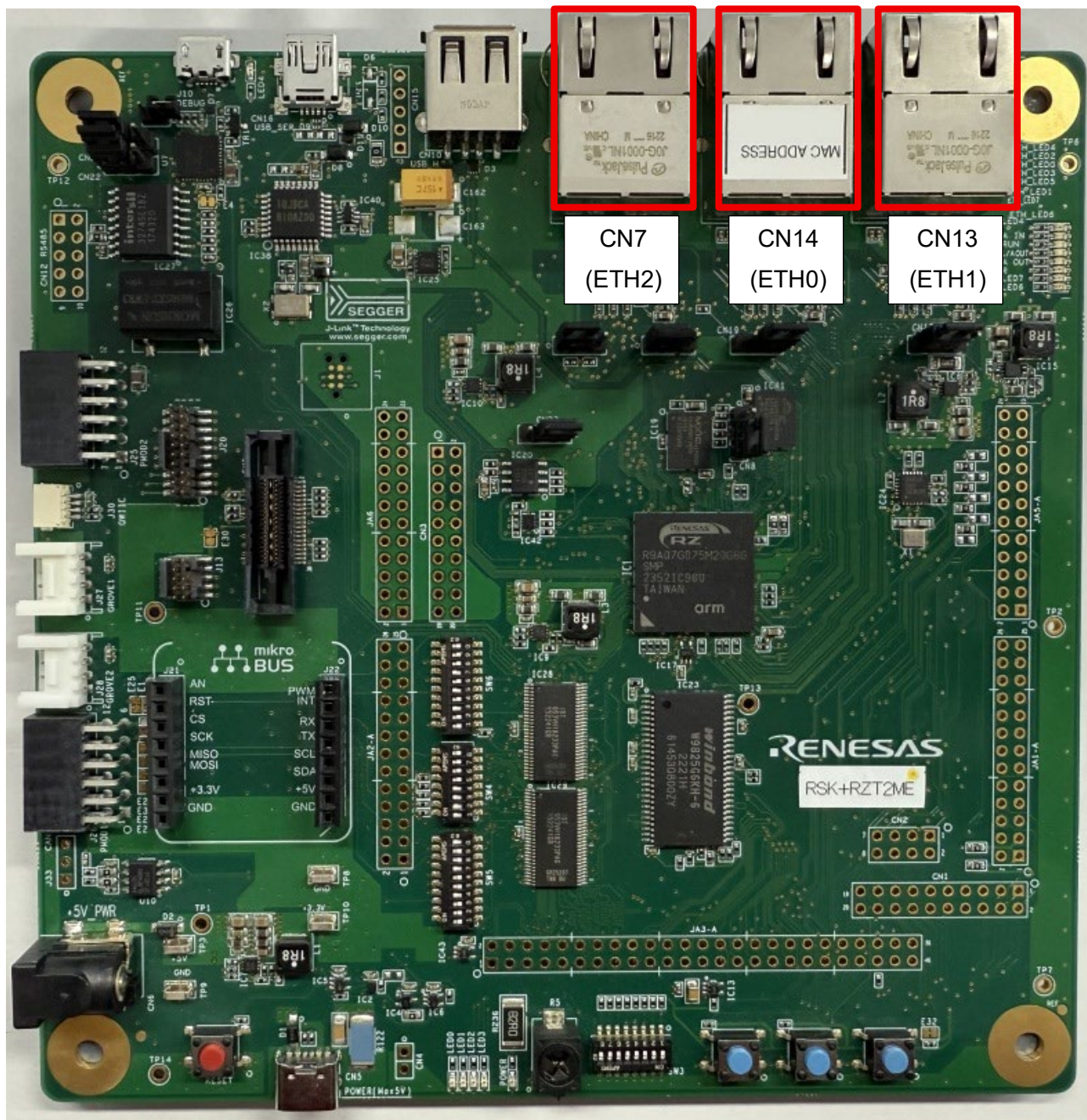


図 3-4 RZ/T2ME RSK board layout(CN7, CN14, CN13)

3.1.3 RZ/T2ME RSK と接続する

3.1.3.1 EWARM 使用時

RZ/T2ME RSK とホストとの接続には、I-jet デバッガを使用します。

まず I-jet とホスト PC を接続してください。

次に I-jet と RZ/T2ME RSK の J20 を接続してください。

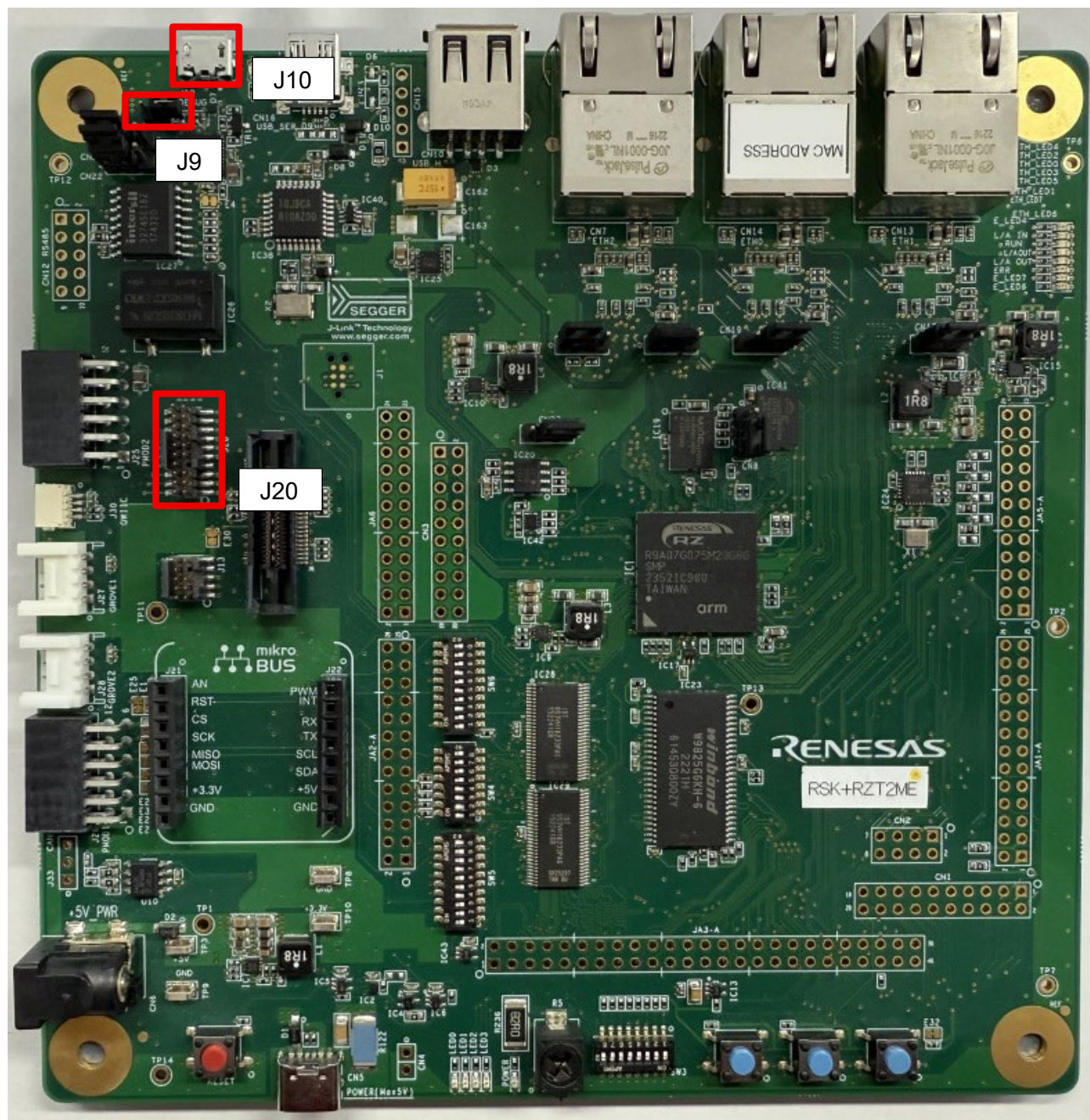


図 3-3 (再掲)RZ/T2ME RSK board layout(J9, J10, J20)

正常に接続されると、I-jet の"USB"と"DBG"のランプが点灯します。

【注】 "TPWR"のランプは消灯している必要があります。

3.1.3.2 e² studio 使用時

RZ/T2ME RSK とホストとの接続には、RZ/T2ME RSK 上のオンボードデバッガ(J-LinkOB)を使用します。

RZ/T2ME RSK の"J-Link OB USB Connector"(J10)とホスト PC を接続してください。

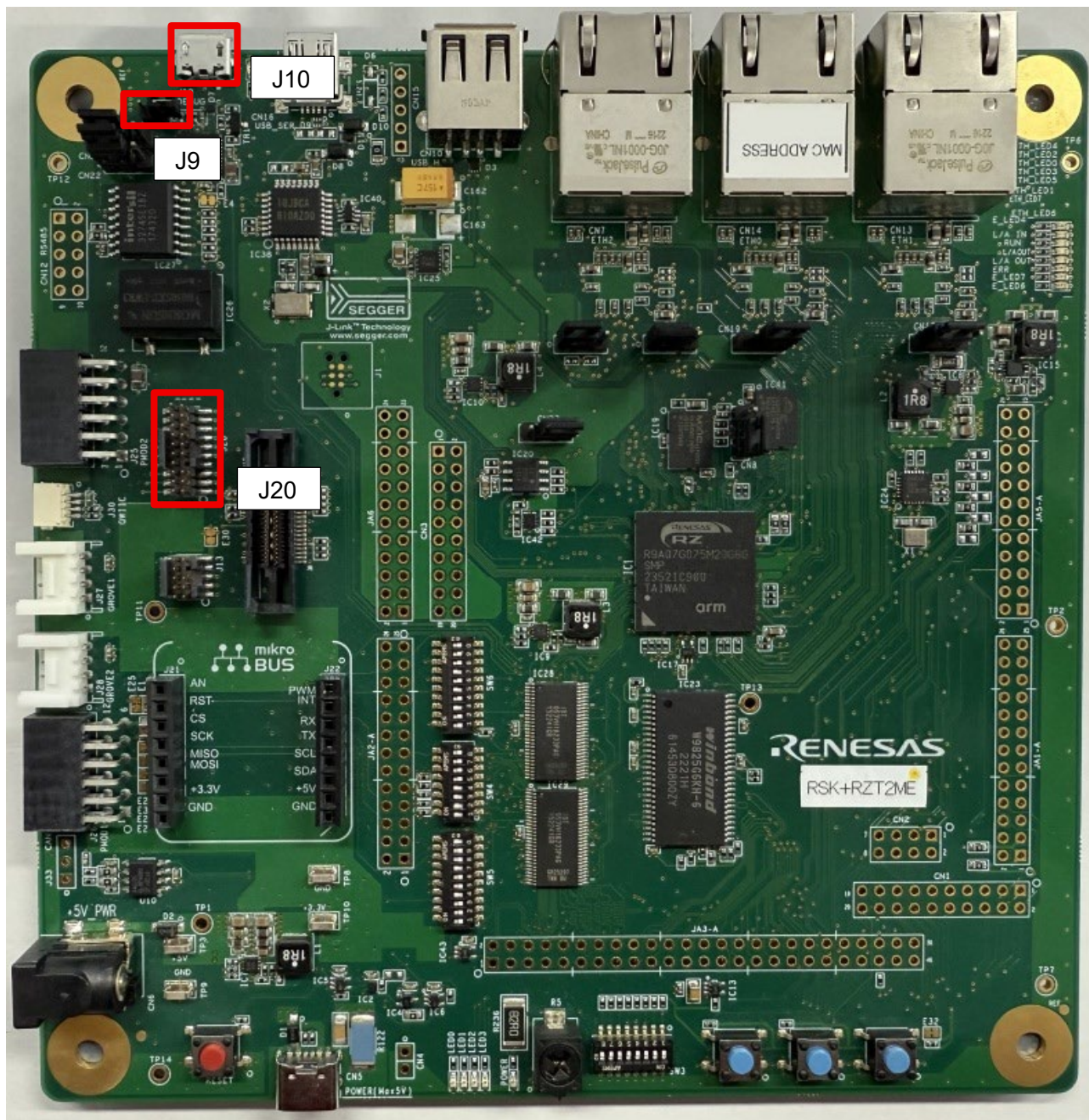


図 3-3 (再掲)RZ/T2ME RSK board layout(J9, J10, J20)

【注】 正しく接続できているかは、以下の手順で確認することができます。

- 【注】
1. e² studio 上のツールバーより"Launch Configuration"横の歯車アイコンをクリック
 2. 現れたダイアログより"Debugger"タブを選択
 3. 現れた画面より"Connection Setting"タブを選択
 4. "J-Link" -> "J-Link Serial" の項目にて、接続したボードの J-Link シリアル番号が選択できれば接続は成功です。

3.2 ソフトウェア要件

サンプルアプリケーションの実行に必要なソフトウェアを示します。

3.2.1 RZ/T2ME RSK のハードウェア設定

サンプルアプリケーションのビルド及び RZ/T2ME RSK へのダウンロードには以下のソフトウェアが必要です。

- 統合開発環境

EWARM と e² studio から選択できます。

— EWARM (*1)(*2)

【注】 1. 本ソフトウェアの使用にはライセンス認証が必要です。

2. RZ/T2ME RSK を使用するために、下記のソフトを導入する必要があります。

- RZ/T2 FSP SC(スマートコンフィグレータ) and FSP インストーラ(EWARM 向け) (*)

<https://www.renesas.com/rskrzt2me>

【注】 上記の統合開発環境 から起動します。

起動方法については"3.2.1.1.1 FSP Smart Configurator を 統合開発環境 から起動する"を参照してください。

— e² studio

e² studio and RZ/T2 FSP インストーラ

<https://www.renesas.com/rskrzt2me>

3.2.1.1 統合開発環境に関するセットアップ

3.2.1.1.1 FSP Smart Configurator を 統合開発環境 から起動する

3.2.1.1.1.1 EWARM の場合

ツールバーより"ツール" -> "ツールの設定"を選択して下さい。

"ツールの設定"より"新規作成"を選択して下さい。

以下の内容を設定して下さい。

表 3-4 ツールの設定 における指定内容

項目	内容
メニューテキスト	FSP Smart Configurator
コマンド	<FSP Smart Configurator(rasc.exe)のパス>
引数	--compiler IAR configuration.xml
初期ディレクトリ	\$PROJ_DIR\$

設定が完了したら、ツールバーより"ツール" -> "FSP Smart Configurator"を選択してください。
FSP Smart Configurator が起動します。

3.2.1.1.1.2 e² studio の場合

e² studio 上のファイルエクスプローラより、"configuration.xml"を開いてください。

FSP Smart Configurator が起動します。

3.2.1.1.2 複数のデバイス接続時にダウンロード先を選択できるようにする

3.2.1.1.2.1 EWARM の場合

1. ツールバーより、"プロジェクト" -> "オプション"を選択してください。

【注】 または IDE のファイルエクスプローラよりプロジェクトを右クリックし、"オプション"を選択してください。

2. "カテゴリ:"ボックス内から"I-jet"を選択してください。

3. "設定"タブ内より、エミュレータの項にある"プローブ選択プロンプトを常に表示"のチェックボックスにチェックを付けてください。

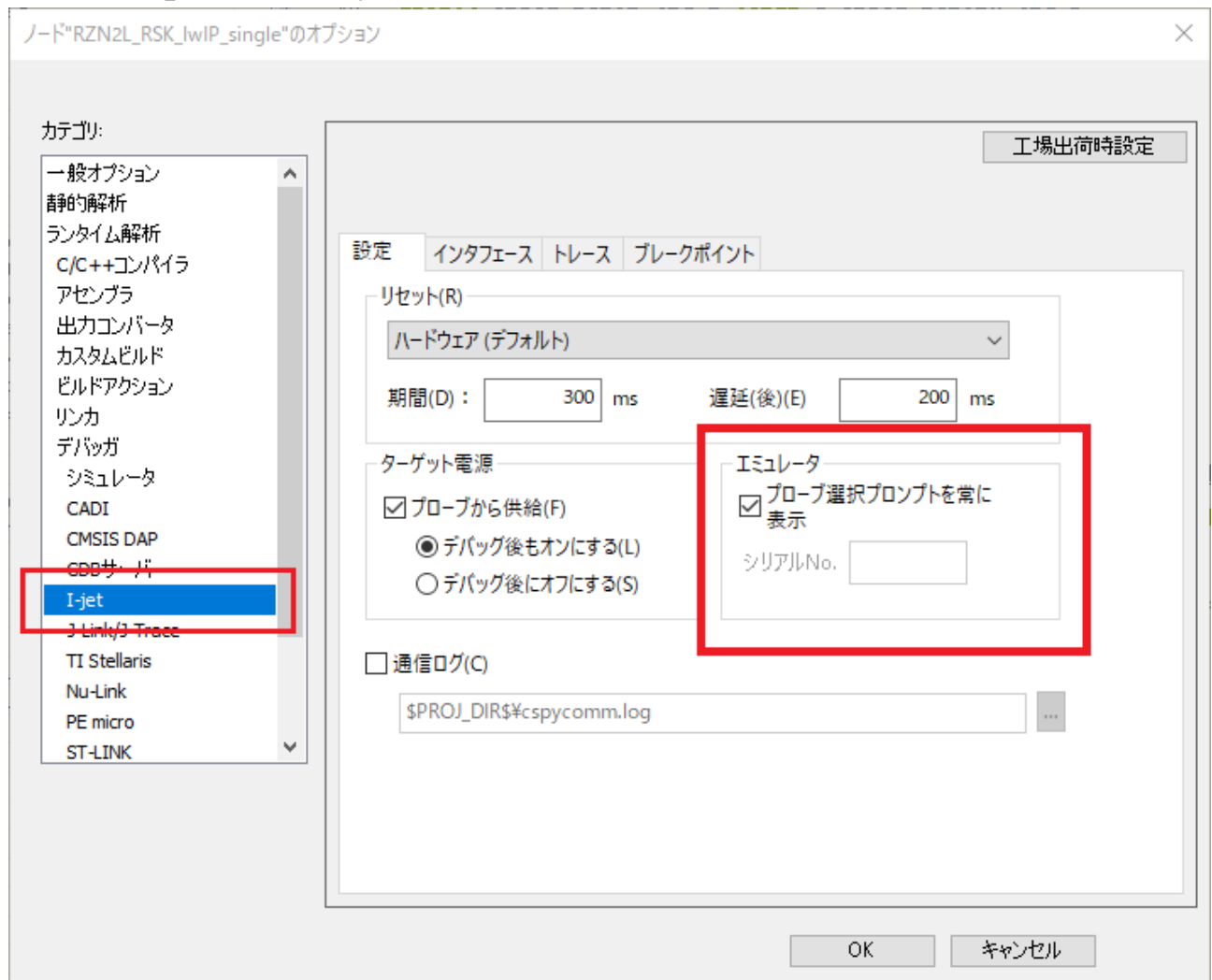


図 3-5 設定画面

以降、アプリケーションダウンロード時に使用する I-jet のシリアルを選択するダイアログが表示されるようになります。

ダイアログ内にて、アプリケーションをダウンロードさせたい RZ/T2ME RSK が接続されている I-jet を指定するようにしてください。

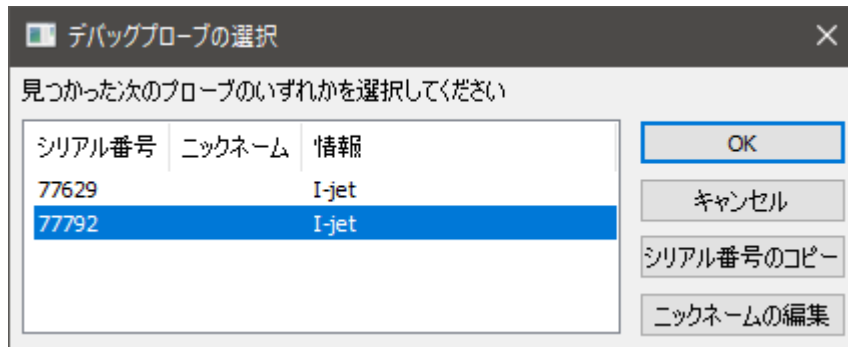


図 3-6 I-jet 選択画面

3.2.1.1.2.2 e² studio の場合

1. e² studio 上のツールバーより"実行(R)" -> "デバッグ構成(B)..."をクリック
2. 現れた画面より"RZT2ME_RSK_IEEE1588 Debug" -> "Debugger"タブを選択
3. 現れた画面より"Connection Settings"タブを選択
4. "J-Link" -> "J-Link Serial" の項目にて、接続したい RZ/T2ME RSK ボードの J-Link シリアル番号を選択してください

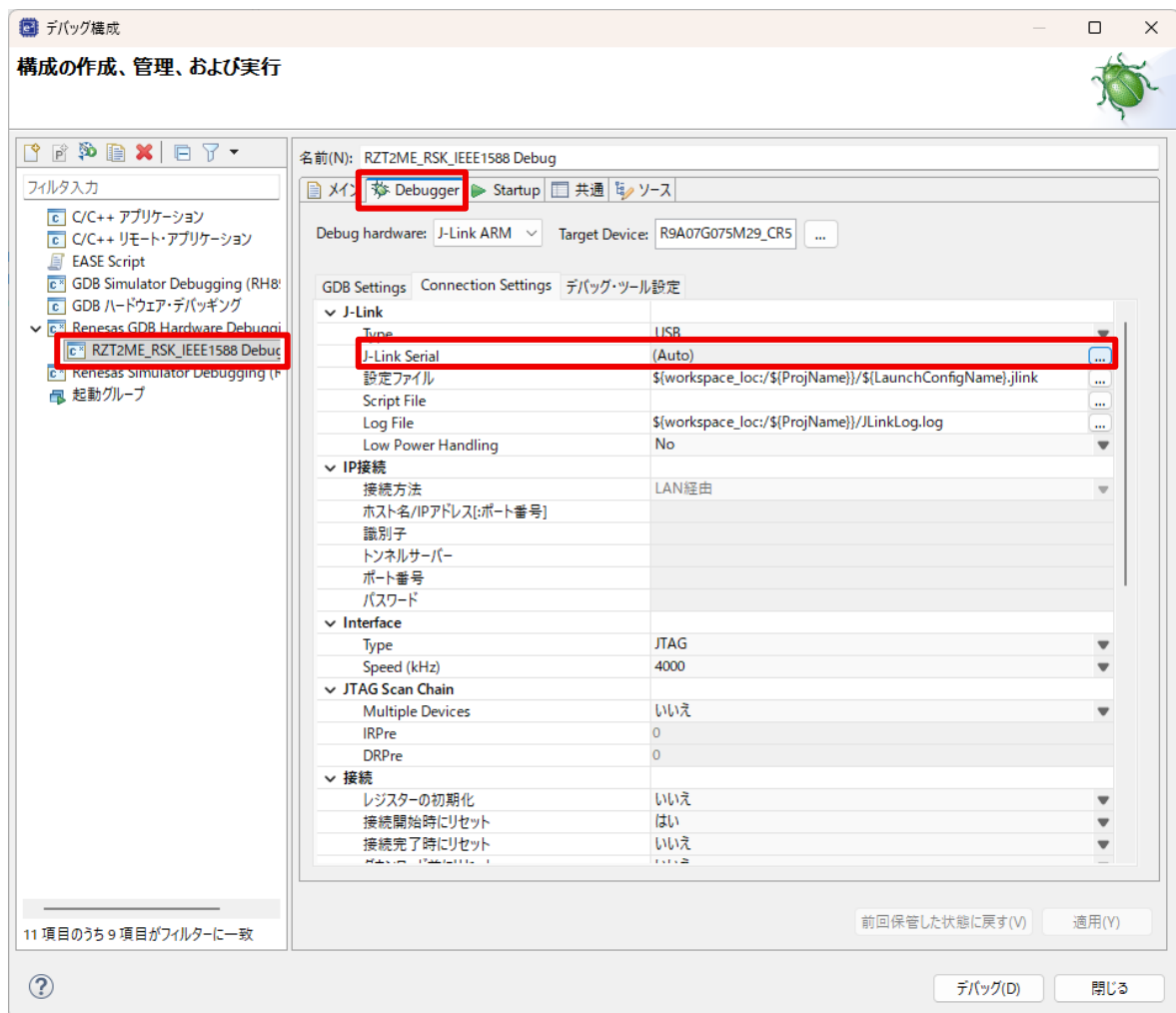


図 3-7 J-Link Serial 設定画面

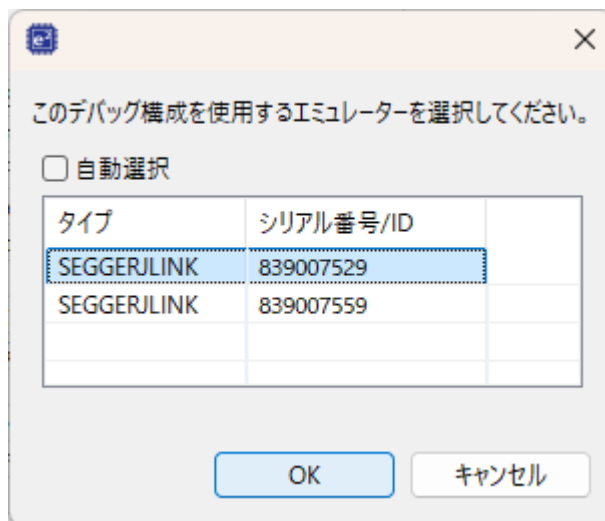


図 3-8 J-Link Serial 選択画面

3.2.2 サードパーティーソフトウェア

3.2.2.1 シリアルターミナルツール

サンプルアプリケーションはUART 経由でログメッセージを出力します。

ログメッセージの確認には、シリアルターミナルツールを使用してください。

- TeraTerm
[Tera Term Open Source Project \(teratermproject.github.io\)](https://teratermproject.github.io)
 TeraTerm におけるシリアルポート設定は以下の通りです。

表 3-5 シリアルポート設定

項目	設定内容
ボーレート	115200
データ	8bit (初期設定)
パリティ	none (初期設定)
ストップビット	1bit (初期設定)
フロー制御	none (初期設定)

4. サンプルアプリケーションの実行

本章は、サンプルアプリケーションのビルドから実行までの一連の手順を示します。

4.1 ビルドとダウンロード

4.1.1 プロジェクトを開く

4.1.1.1 EWARM を使用する場合

EWARM 用プロジェクトの含まれている zip ファイル("RZT2ME_RSK_IEEE1588.zip"(*))を解凍してください。

解凍によって得られる下記のファイルを EWARM で開いてください。

```
/project/rzt2m_rsk/lwip_single/ewarm/RZT2ME_RSK_IEEE1588.eww
```

4.1.1.2 e² studio を使用する場合

以下の手順にしたがってプロジェクトをインポートしてください。

1. e² studio 用プロジェクトの含まれている zip ファイル("RZT2ME_RSK_IEEE1588.zip"(*))を解凍する
2. 解凍によって得られるディレクトリ("RZT2ME_RSK_IEEE1588")をプロジェクト生成先となるディレクトリに移動させる
3. e² studio を起動し、プロジェクト作成したいディレクトリを開く
4. ツールバーの"File"より"Open Projects from fileSystem..."を選択する
5. 現れたダイアログより"[Directory...]"をクリックし、先ほど移動させたディレクトリ("RZT2ME_RSK_IEEE1588/project/rzt2m_rsk/lwip_single/e2studio")を選択する
6. "[Finish]"をクリックすることでプロジェクトがインポートされる

4.1.2 デバイスの事前設定

ビルド前に、デバイスごとの設定を行う必要があります。

設定内容は以下の通りです。

- デバイスの MAC アドレス
- デバイスの IP アドレス
- BMCA で使用するクロック情報
- 伝送遅延時間の計測方法(*)
- 時刻配信の方式(*)
- トランスポート設定(*)

【注】 *: 全てのデバイスで設定を統一する必要があります

4.1.2.1 MAC アドレス変更方法

1. FSP Smart Configurator を起動して下さい。

【注】 起動方法については"3.2.1.1.1 FSP Smart Configurator を 統合開発環境 から起動する"を参照してください。

2. FSP Smart Configurator にて、FSP Configuration 関連ビュー下部にある"Stack"タブを開いてください。

3. ビュー左上の"Threads"のリストより、Main Thread 内にある g_ether0 Ethernet Driver on r_ether を選択して下さい。

4. Properties ビューの Setting より、Module g_ether0 Ethernet Driver on r_ether->General まで展開して下さい。

5. General 内にある"MAC address"がデバイスの MAC アドレスとなります。

割り振りたいアドレスに応じて値を書き換えて下さい。

【注】 デバイスごとに異なる値を設定して下さい

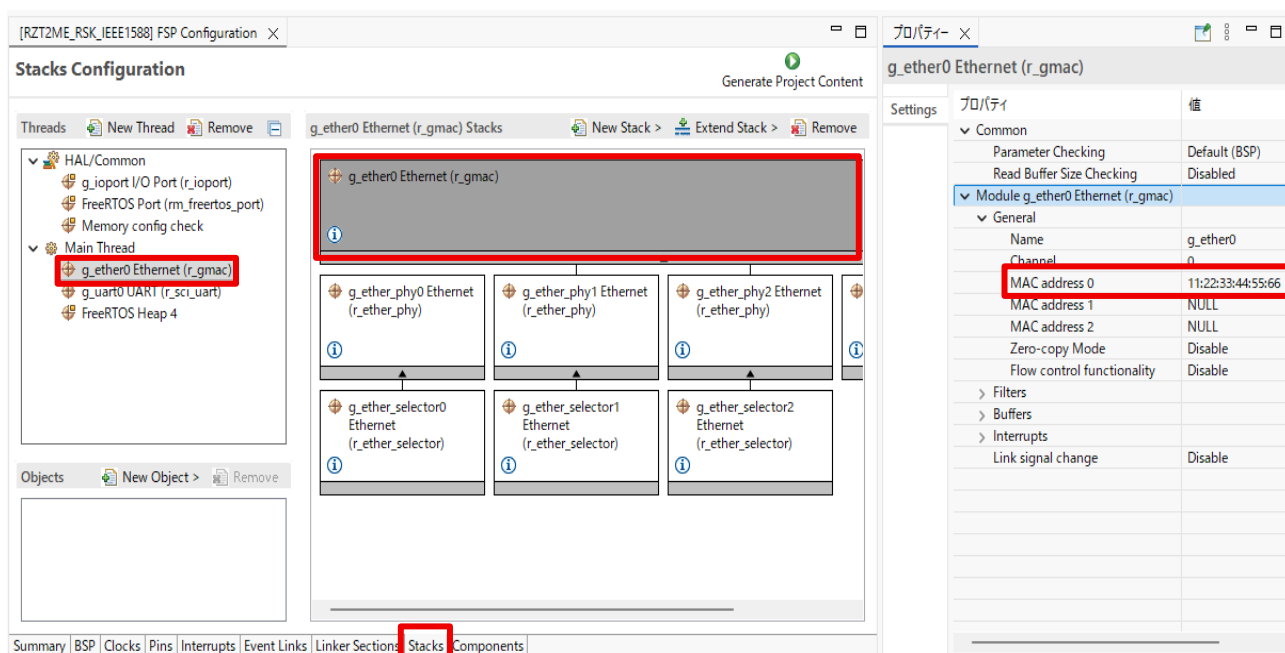


図 4-1 Mac アドレスの設定

4.1.2.2 IP アドレス変更方法

プロジェクトのディレクトリより、"app_config.h"を開いてください。

/common/renesas/application/app_config.h

app_config.h の 7 行目より、

```
; #define NETIF_0_TCPIP_IP LWIP_MAKEU32(192,168, 1,170)
```

がイーサネットに設定する IP アドレスを指定している箇所となります。

【注】 リンクアップしたイーサネットに NETIF_0_TCPIP_IP で指定されている IP アドレスが割り当てられません。

割り振りたいアドレスに応じて値を書き換えて下さい。

【注】 デバイスごとに異なる値を設定してください

4.1.2.3 BMCA で使用するクロック情報

プロジェクトのディレクトリより、"app_config.h"を開いてください。

```
; /common/renesas/application/app_config.h
```

app_config.h の 14 行目より、

```
; // setting BMCA Clock Parameter  
; // Priority1  
; #define BMCA_PRIORITY_1          12  
; // Clock Class  
; #define BMCA_PRIORITY_CLOCK_CLASS 248  
; // Clock Accuracy  
; #define BMCA_PRIORITY_CLOCK_ACCURACY 0xFE  
; // Clock Variance  
; #define BMCA_PRIORITY_CLOCK_VARIANCE 0xFFFF  
; // Priority2  
; #define BMCA_PRIORITY_2          128  
; // Domain Number  
; #define BMCA_DOMAIN_NUMBER      0
```

のマクロ定義によって BMCA で使用するクロック情報を設定することが可能です。

各項目で以下の設定が可能です：

- #define BMCA_PRIORITY_1
 - Priority1 の値です
 - 設定可能な値域は 0 ~ 255 です
- #define BMCA_PRIORITY_CLOCK_CLASS
 - Clock Class の値です
 - 設定可能な値域は 0 ~ 255 です
- #define BMCA_PRIORITY_CLOCK_ACCURACY
 - Clock Accuracy の値です
 - 設定可能な値域は 0 ~ 255 です
- #define BMCA_PRIORITY_CLOCK_VARIANCE
 - Clock Variance の値です
 - 設定可能な値域は 0 ~ 65535 です
- #define BMCA_PRIORITY_2
 - Priority2 の値です
 - 設定可能な値域は 0 ~ 255 です
- #define BMCA_DOMAIN_NUMBER
 - Domain Number の値です
 - 設定可能な値域は 0 ~ 255 です

【注】 ネットワーク上の同一"Domain Number"のデバイス同士が時刻同期します

4.1.2.4 伝送遅延時間の計測方法

プロジェクトのディレクトリより、"app_config.h"を開いてください。

```
; /common/renesas/application/app_config.h
```

app_config.h の 33 行目より、

```
; // setting ptp delay mechanism  
; // 0: E2E( End to End Mechanism )  
; // 1: P2P( Peer to Peer Mechanism )  
; #define PTP_DELAY_MECHANISM 1
```

のマクロ定義によって伝送遅延時間の計測方法を切り替えることが可能です。

【注】 E2E による伝送遅延時間の計測を行う場合は、0 を設定してください

【注】 P2P による伝送遅延時間の計測を行う場合は、1 を設定してください

4.1.2.5 時刻配信の方式

プロジェクトのディレクトリより、"app_config.h"を開いてください。

```
; /common/renesas/application/app_config.h
```

app_config.h の 38 行目より、

```
; // setting ptp time distribution  
; // 0:two step clock  
; // 1:one step clock  
; #define PTP_ONE_STEP_FLAG 0
```

のマクロ定義によって時刻配信の方式を切り替えることが可能です。

【注】 two step による時刻配信を行う場合は、0 を設定してください

【注】 one step による時刻配信を行う場合は、1 を設定してください

4.1.2.6 トランスポート設定

プロジェクトのディレクトリより、"app_config.h"を開いてください。

```
; /common/renesas/application/app_config.h
```

app_config.h の 28 行目より、

```
; // setting transport layer  
; // 0: Layer 2: Raw packet( IEEE 802.3 )  
; // 1: Layer 3: IPv4 UDP packet  
; #define PTP_TRANSPORT_LAYER 1
```

のマクロ定義によってトランスポートレイヤーを切り替えることが可能です。

【注】 Layer2(Ethernet Frame)による通信を行う場合は、0 を設定してください

【注】 Layer3(IPv4 UDP Frame)による通信を行う場合は、1 を設定してください

4.1.3 ビルドと RZ/T2ME RSK へのダウンロード

4.1.3.1 プロジェクトソースの生成

ビルドを行う前に、FSP Smart Configurator を用いてプロジェクトを生成してください。

- IDE から FSP Smart Configurator を起動してください。

【注】 起動方法については"3.2.1.1.FSP Smart Configurator を 統合開発環境 から起動する"を参照してください。

- FSP Smart Configurator 内より、"Generate Project Content"を選択してください。プロジェクトが生成されます。

4.1.3.2 ビルドとダウンロード

4.1.3.2.1 EWARM の場合

プロジェクト生成後、ツールバーより"プロジェクト" -> "メイク" を選択してください。

アプリケーションのビルドが実行されます。

ビルド完了後、"プロジェクト" -> "ダウンロードしてデバッグ"を選択してください。

アプリケーションの RZ/T2ME RSK へのダウンロードが開始します。

4.1.3.2.2 e² studio 使用時

プロジェクト生成後、ツールバーより"Project" -> "build" を選択してください。

アプリケーションのビルドが実行されます。

ビルド完了後、"Run" -> "Debug"を選択してください。

アプリケーションの RZ/T2ME RSK へのダウンロードが開始します。

4.2 サンプルアプリケーションの起動

RZ/T2ME RSK へのダウンロードが完了すると、アプリケーション実行待機状態となります。

以下の手順でアプリケーションを起動して下さい。

【注】 動作確認内容については"2.1.3 アプリケーション動作確認内容"を参照してください。

【注】 本プログラムの動作中に、イーサネットケーブルの挿抜は行わないでください。動作中にイーサネットケーブルの挿抜が発生した場合、時刻同期が正常に復旧しない場合があります。

4.2.1 EWARM 使用時

ツールバーより"デバッグ" -> "実行"を選択してください。

アプリケーションが起動します。

4.2.2 e²studio 使用時

ツールバーより"Run" -> "Resume"を選択してください。

アプリケーションが起動します。

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	Aug. 29, 2024	-	新規作成
1.10	Apr. 16, 2026	-	FSP v4.0.0 対応

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えば、フラッシュメモリ、レイアウトパターンの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ輻射量などが異なる場合があります。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

ご注意書き

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、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。
 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いません。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

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/

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。