

## RX ファミリ

### EtherCAT モジュール Firmware Integration Technology

---

#### 要旨

本アプリケーションノートは、Firmware Integration Technology(FIT)を使用した EtherCAT®モジュールについて説明します。本モジュールは産業イーサネット通信用 EtherCAT スレーブコントローラ (EtherCAT Slave Controller : ESC) を内蔵した RX ファミリで Beckhoff 社製 EtherCAT スレーブスタックコード (Slave Stack Code : SSC)を使用するためのインタフェースを提供します。

本モジュールには SSC は含まれておりません。EtherCAT Technology Group(ETG 協会)より SSC ツールを入手の上、SSC を生成してください。

以降、本モジュールを EtherCAT FIT モジュールと称します。

#### 対象デバイス

- RX72M 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

## 内容

1. 概要 .....	3
1.1 EtherCAT FIT モジュールとは .....	3
2. API 情報.....	4
2.1 ハードウェアの要求 .....	4
2.2 ソフトウェアの要求 .....	4
2.3 サポートされているツールチェーン .....	4
2.4 使用する割り込みベクタ .....	4
2.5 ヘッダファイル .....	4
2.6 整数型 .....	4
2.7 コンパイル時の設定 .....	5
2.8 コードサイズ .....	7
2.9 引数 .....	7
2.10 戻り値 .....	7
2.11 コールバック関数 .....	8
2.12 FIT モジュールの追加方法 .....	10
2.13 for 文、while 文、do while 文について .....	11
3. API 関数.....	12
3.1 R_ECAT_Initial .....	12
3.2 R_ECAT_Open.....	13
3.3 R_ECAT_Close .....	14
3.4 R_ECAT_Process .....	15
4. 端子設定 .....	16
5. デモプロジェクト .....	17
5.1 ecat_demo_rskrx72m.....	17
5.1.1 ボードの設定と接続 .....	17
5.1.2 デモプロジェクト動作手順 .....	17
5.2 ecat_demo_comrx72m.....	17
5.2.1 ボードの設定と接続 .....	17
5.2.2 デモプロジェクト動作手順 .....	17
6. 付録 .....	18
6.1 動作確認環境 .....	18
7. 参考ドキュメント .....	22
改訂記録.....	23

## 1. 概要

EtherCAT FIT モジュールは、SSC を使用して EtherCAT 通信を行うための手段を提供します。

以下に本モジュールがサポートしている機能を列挙します。

- SSC から ESC に対するハードウェアアクセス制御を行うインタフェース
- PHY の初期化
- EtherCAT スレーブスタックの開始、実行、終了処理
- ESC 割り込み

### 1.1 EtherCAT FIT モジュールとは

本モジュールは API として、プロジェクトに組み込んで使用します。本モジュールの組み込み方法については「2.12 FIT モジュールの追加方法」を参照してください。

## 2. API 情報

本 FIT モジュールは、下記の条件で動作を確認しています。

### 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。

- EtherCAT スレーブコントローラ(ESC)

### 2.2 ソフトウェアの要求

FIT モジュールは以下の FIT モジュールに依存しています。

- ボードサポートパッケージモジュール(r\_bsp) v5.20 以上
- CMT モジュール(r\_cmt) v3.40 以上

### 2.3 サポートされているツールチェーン

本 FIT モジュールは「6.1 動作確認環境」に示すツールチェーンで動作確認を行っています。

### 2.4 使用する割り込みベクタ

ESC の割り込みは、R\_ECAT\_Open 関数を実行することで有効にされます。

表 2.1 に本 FIT モジュールが使用する割り込みベクタを示します。

表 2.1 使用する割り込みベクター一覧

デバイス	割り込みベクタ
RX72M	GROUPAL1(ベクタ番号 : 113) ESCI (EtherCAT 割り込み) : ビット 13 ESC SYNC0 割り込み(ベクタ番号 : 252※) ESC SYNC1 割り込み(ベクタ番号 : 253※)

※ESC SYNC0 割り込み、および ESC SYNC1 割り込みは選択型割り込み A であり、ベクタ番号 208~255 に割り当てることが可能です。ベクタ番号を変更する場合は r\_ecat\_rx\_private.h の割り込み関連定義を適宜修正してください。

### 2.5 ヘッダファイル

すべての API 呼び出しとそれをサポートするインタフェース定義は r\_ecat\_rx\_if.h に記載しています。

ビルド時に設定可能なコンフィギュレーションオプションは r\_ecat\_rx\_config.h ファイルで選択または定義されています。

本 FIT モジュールの API をユーザプログラムから参照するには、r\_ecat\_rx\_if.h をインクルードしてください。

### 2.6 整数型

このドライバは 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。

## 2.7 コンパイル時の設定

本モジュールのコンフィギュレーションオプションの設定は、`r_ecat_rx_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。

コンフィギュレーションオプション( <code>r_ecat_rx_config.h</code> )	
ECAT_CFG_MODE_SEL ※デフォルト値は"0"	このオプションは ESC と PHY の接続インタフェースを指定します。 ● 0 : MII インタフェースを指定します。 ※現状は 0 固定で変更できません
ECAT_PHY_OFFSET_ADDRESS ※デフォルト値は"1"	このオプションは PHY アドレスオフセット設定レジスタ (PHYOFF) に設定する値を定義します。 お使いのシステム構成に従い適切な値を設定してください。
ECAT_CFG_CH0_PHY_ADDRESS ※デフォルト値は"1" ECAT_CFG_CH1_PHY_ADDRESS ※デフォルト値は"2"	このオプションは PHY-LSI のアドレスを定義します。 お使いのシステムの構成に従い適切な値を設定してください。
ECAT_CFG_PHYLINK0 ECAT_CFG_PHYLINK1 ※デフォルト値は"1"	このオプションは CATn_LINKSTA 端子極性ビット (LINKPOLn) に設定する値を定義します。 (n=0,1) ● 0 : アクティブ High ● 1 : アクティブ Low お使いのシステムの構成に従い適切な値を設定してください。
ECAT_CFG_EEPROM_SIZE ※デフォルト値は"0"	このオプションは EEPROM サイズ設定ビット (PROMSIZE) に設定する値を定義します。 ● 0 : 16K ビット以下 ● 1 : 32K ビット~4M ビット お使いのシステムの構成に従い適切な値を設定してください。
ECAT_CFG_ESC_INT_COND ※デフォルト値は"0"	このオプションは ESCI 割り込み生成条件設定ビット (ESCIC) に設定する値を定義します。 ● 0 : PDI_IRQ が"1"のとき割り込み発生 ● 1 : PDI_IRQ が"0"のとき割り込み発生
ECAT_CFG_SYNC0_INT_COND ECAT_CFG_SYNC1_INT_COND ※デフォルト値は"0"	このオプションは SYNCn 割り込み生成条件設定ビット (SYNCnC) に設定する値を定義します。 (n=0,1) ● 0 : SYNCn 信号の立ち上がりで割り込み発生 ● 1 : SYNCn 信号の立ち下がりで割り込み発生
ECAT_CFG_AL1_INT_PRIORITY ECAT_CFG_SYNC0_INT_PRIORITY ECAT_CFG_SYNC1_INT_PRIORITY ※デフォルト値は"15"	このオプションは ESCI 割り込み、または SYNCn 割り込み (n=0,1) の優先レベルを設定します。 優先レベルの最低値が 1、最高値が 15 です。
ECAT_CFG_TX_SHIFT0 ECAT_CFG_TX_SHIFT1 ※デフォルト値は"0"	このオプションはポート n 送信信号遅延設定ビット (TXSFTn[1:0]) に設定する値を定義します。 (n=0,1) ● 00 : 0ns ● 01 : 10ns ● 10 : 20ns ● 11 : 30ns お使いのシステムの構成に従い適切な値を設定してください。
ECAT_CFG_PHY_DLAY_RESET ※デフォルト値は"1000"	このオプションは PHY リセットから PHY レジスタにアクセス可能となるまでの時間を定義します。 (単位 : $\mu$ s) お使いのシステム構成に従い適切な値を設定してください。

コンフィギュレーションオプション(r_ecat_rx_config.h)	
ECAT_CFG_USE_SUPPORTED_PHY ※デフォルト値は"0"	<p>このオプションは FIT モジュールがサポートしている PHY を使用するかどうかを定義します。 サポートしている PHY を設定した場合は PHY レジスタの初期設定は不要となります。</p> <ul style="list-style-type: none"><li>● 0 : 使用しない</li><li>● 1 : KSZ8041NL を使用する</li><li>● 2 : KSZ8081MNX を使用する</li></ul> <p>Renesas Starter Kit+ for RX72M で使用する場合は"1"を テセラ・テクノロジー社製 RX72M 搭載評価ボードを使用する 場合は"2"を設定してください。</p>

## 2.8 コードサイズ

本モジュールのコードサイズを下表に示します。

掲載した値は、「2.3 サポートされているツールチェーン」の C コンパイラでのコンパイルオプションがデフォルト時の参考値です。コンパイルオプションのデフォルトは最適化レベル：2、最適化のタイプ：サイズ優先、データ・エンディアン：リトルエンディアンです。コードサイズは 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。

ROM および RAM のサイズ		
デバイス	領域	サイズ
RX72M	ROM	1417 バイト
	RAM	92 バイト

## 2.9 引数

API 関数の引数を示します。この列挙型は API 関数のプロトタイプ宣言とともに r\_ecat\_rx\_if.h で記載されています。

```
/* EtherCAT スレーブスタックを開始する契機 */
typedef enum
{
    OPEN_BOOT, /* ブート */
    OPEN_REBOOT /* リブート */
} ecat_open_t;
```

## 2.10 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\_ecat\_rx\_if.h で記載されています。

```
/* ECAT API エラーコード */
typedef enum
{
    ECAT_SUCCESS = 0, /* 関数は正常終了したことを示します。 */
    ECAT_ERR_INVALID_PTR = -1, /* ポインタ引数が NULL または FIT_NO_PTR です。 */
    ECAT_ERR_INVALID_DATA = -2, /* 引数の値が範囲外です。 */
    ECAT_ERR_INVALID_ARG = -3, /* 引数が不正です。 */

    ECAT_ERR_EEPROM = -4, /* EEPROM のローディングに失敗したことを示します。 */
    ECAT_ERR_OTHER = -5 /* その他のエラーです。 */
} ecat_return_t;
```

## 2.11 コールバック関数

本モジュールでは下記のタイミングでユーザが設定したコールバック関数を呼び出します。

- R\_ECAT\_Open 関数終了タイミング
- R\_ECAT\_Close 関数終了タイミング
- ESCI 割り込み発生タイミング
- SYNC0 割り込み発生タイミング
- SYNC1 割り込み発生タイミング

各コールバック関数のポインタ構造体のメンバを示します。コールバック関数は、記載された構造体メンバに、ユーザの関数のアドレスを格納することで設定されます。

```
/* コールバック関数のポインタ構造体 */
typedef struct
{
    void (*pcb_open_func) (void *); /* R_ECAT_Open 関数終了タイミング */
    void (*pcb_close_func) (void *); /* R_ECAT_Close 関数終了タイミング */
    void (*pcb_intesci_hnd) (void*); /* ESCI 割り込み発生タイミング */
    void (*pcb_intsync0_hnd) (void*); /* SYNC0 割り込み発生タイミング */
    void (*pcb_intsync1_hnd) (void*); /* SYNC1 割り込み発生タイミング */
} ecacb_t;
```

各コールバック関数の引数構造体のメンバを示します。

コールバック関数が呼び出されるとき、記載の構造体メンバに格納された変数が引数として渡されます。

※現在、引数は未使用のためダミーとなっています。使用する場合は適宜修正してください。

```
/* コールバック関数の引数構造体 */
typedef struct
{
    uint32_t      arg1; /* 第 1 引数 */
    uint32_t      arg2; /* 第 2 引数*/
} ecacb_arg_t;
```

引数の型は void ポインタ型で渡されるため、コールバック関数の引数は次のように実装されています。

```
/* R_ECAT_Open 関数終了タイミングコールバック関数の例 */
if ((NULL != cb_func.pcb_open_func) && (FIT_NO_FUNC != cb_func.pcb_open_func))
{
    cb_arg.arg1 = 0;
    cb_arg.arg2 = 1;
    (*cb_func.pcb_open_func)((void *) &cb_arg);
}
```



本モジュールではコールバック関数の登録は API を使わず、R\_ECAT\_Initial 関数の中で行っており、次の様に実装してあります。

```
// --- Initialize the callback function pointer
cb_func.pcb_open_func = NULL;
cb_func.pcb_close_func = NULL;

// --- Initialize the interrupt handler pointer
cb_func.pcb_intesci_hnd = &ecat_esci_inthdr;
cb_func.pcb_intsync0_hnd = &ecat_sync0_inthdr;
cb_func.pcb_intsync1_hnd = &ecat_sync1_inthdr;
```

R\_ECAT\_Open 関数および R\_ECAT\_Close 関数のコールバック関数は未登録となっています。

必要に応じて追加してください。

また、割り込みについては SSC の割り込みハンドラを実行するようにコールバック関数を実装してあります。処理を追加する場合はコールバック関数を適宜修正してください。

呼び出しタイミング	コールバック関数	実行する SSC の関数
ESCI 割り込み	ecat_esci_inthdr	PDI_Isr 関数
SYNC0 割り込み	ecat_sync0_inthdr	Sync0_Isr 関数
SYNC1 割り込み	ecat_sync1_inthdr	Sync1_Isr 関数

## 2.12 FIT モジュールの追加方法

スマートコンフィグレータで EtherCAT FIT モジュールを使用できるようにするには、e<sup>2</sup> studio に追加する必要があります。

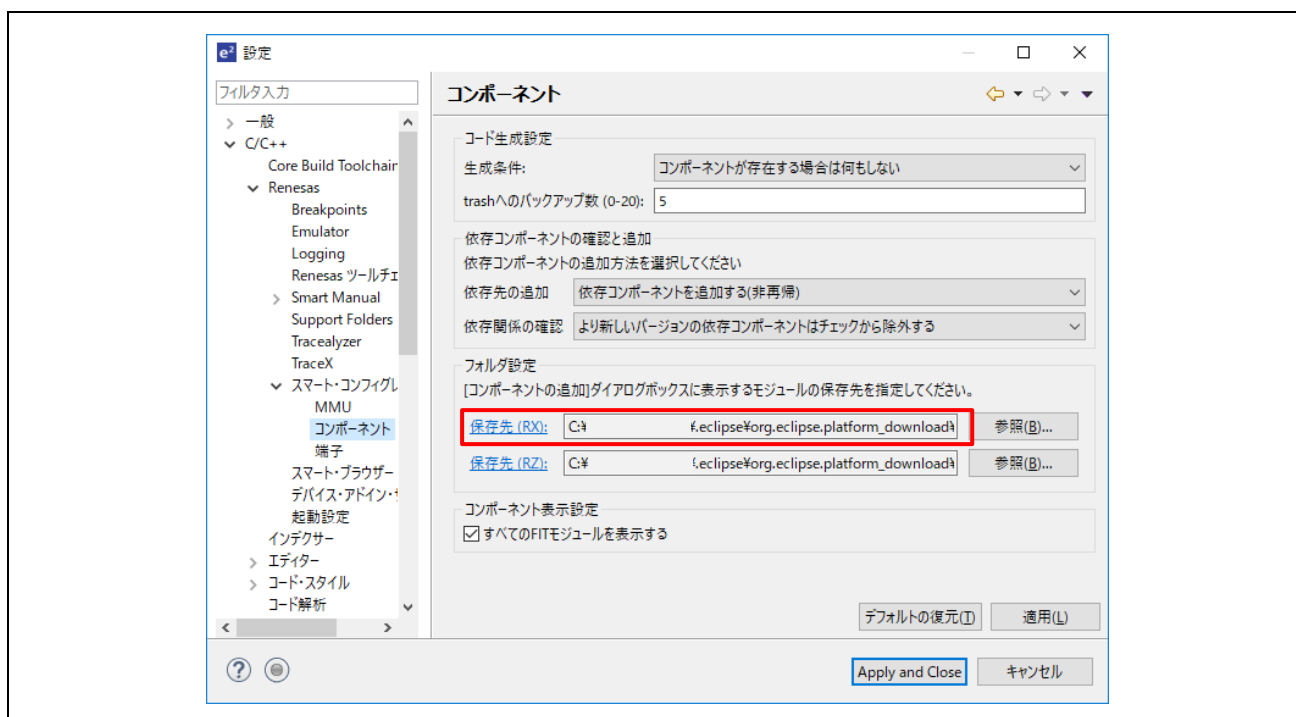
手動で追加する方法を示します。

(1) e<sup>2</sup> studio の FIT モジュールの保存先フォルダに EtherCAT FIT モジュールをコピーする

e<sup>2</sup> studio で FIT モジュールの保存先を確認します。

- 「ウインドウ」→「設定」→設定ウインドウが開きます。
- 「C/CC+」→「Renesas」→「スマートコンフィグレータ」→「コンポーネント」を選択してください。

フォルダ設定の保存先(RX):として表示されているフォルダになります。



EtherCAT FIT モジュールはサンプルプログラムの FITModules フォルダに格納されています。

- an-r01an4881xxNNNN-rx-ecat\FITModules フォルダにあるファイルを FIT モジュールの保存先フォルダにコピーしてください。

r\_ecat\_rx\_vN.NN.xml

r\_ecat\_rx\_vN.NN.zip

r\_ecat\_rx\_vN.NN\_extend.mdf

なお、NNNN および N.NN はバージョンを表す数値になります。

## 2.13 for 文、while 文、do while 文について

本モジュールでは、レジスタの反映待ち処理等で for 文、while 文、do while 文（ループ処理）を使用しています。これらループ処理には、「WAIT\_LOOP」をキーワードとしたコメントを記述しています。そのため、ループ処理にユーザがフェイルセーフの処理を組み込む場合は、「WAIT\_LOOP」で該当の処理を検索できます。

以下に記述例を示します。

while 文の例：

```
/* WAIT_LOOP */
while(0 == SYSTEM.OSCOVFSR.BIT.PLOVF)
{
    /* The delay period needed is to make sure that the PLL has stabilized.*/
}
```

for 文の例：

```
/* Initialize reference counters to 0. */
/* WAIT_LOOP */
for (i = 0; i < BSP_REG_PROTECT_TOTAL_ITEMS; i++)
{
    g_protect_counters[i] = 0;
}
```

do while 文の例：

```
/* Reset completion waiting */
do
{
    reg = phy_read(ether_channel, PHY_REG_CONTROL);
    count++;
} while ((reg & PHY_CONTROL_RESET) && (count < ETHER_CFG_PHY_DELAY_RESET)); /*
WAIT_LOOP */
```

---

### 3. API 関数

---

#### 3.1 R\_ECAT\_Initial

---

この関数は、EtherCAT モジュールの初期設定を行います。この関数は他の API 関数を使用する前に実行される必要があります。

##### Format

```
void R_ECAT_Initial(void)
```

##### Parameters

なし

##### Return Values

なし

##### Properties

r\_ecat\_rx\_if.h にプロトタイプ宣言されています。

##### Description

本関数では下記のコールバック関数の登録を行います。

- R\_ECAT\_Open 関数終了タイミング
- R\_ECAT\_Close 関数終了タイミング
- ESCI 割り込み発生タイミング
- SYNC0 割り込み発生タイミング
- SYNC1 割り込み発生タイミング

##### Example

```
/* Initialize EtherCAT module */  
R_ECAT_Initial();
```

---

## 3.2 R\_ECAT\_Open

---

この関数は、EtherCAT スレーブスタックを開始する際に必要な設定を行います。また、ESC 割り込みを有効にします。この関数は R\_ECAT\_Process 関数を使用する前に実行される必要があります。

### Format

```
ecat_return_t R_ECAT_Open(uint32_t mode)
```

### Parameters

uint32\_t mode

EtherCAT スレーブスタックを開始する契機を指定します。

OPEN\_BOOT /\* ブート \*/

OPEN\_REBOOT /\* リブート \*/

### Return Values

[ECAT\_SUCCESS] /\* 関数は正常終了したことを示します。 \*/

[ECAT\_ERR\_EEPROM] /\* EEPROM のロードに失敗しました。 \*/

[ECAT\_ERR\_OTHER] /\* PHY-LSI の初期設定に失敗しました。 \*/

### Properties

r\_ecat\_rx\_if.h にプロトタイプ宣言されています。

### Description

EtherCAT スレーブスタックを開始する際に必要な設定を行います。

- ESC の初期化
- EEPROM のロード
- PHY-LSI のリセット解除および PHY レジスタ初期設定※
- ESC 割り込みの有効化
- EtherCAT スレーブスタックの初期化

※PHY レジスタの初期設定は使用する PHY-LSI に応じて適宜修正してください

### Example

```
/* Open EtherCAT module */  
R_ECAT_Open(OPEN_BOOT);
```

---

### 3.3 R\_ECAT\_Close

---

この関数は EtherCAT スレーブスタックを終了する際に必要な設定を行います。

#### Format

```
ecat_return_t R_ECAT_Close(void)
```

#### Parameters

なし

#### Return Values

[ECAT_SUCCESS]	/* 関数は正常終了したことを示します。 */
[ECAT_ERR_OTHER]	/* その他のエラーです。 */

#### Properties

r\_ecat\_rx\_if.h にプロトタイプ宣言されています。

#### Description

EtherCAT スレーブスタックを終了する際に必要な設定を行います。

#### Example

```
/* Close EtherCAT module */  
R_ECAT_Close();
```

---

### 3.4 R\_ECAT\_Process

---

この関数は EtherCAT スレーブスタックを実行します。

#### Format

ecat\_return\_t R\_ECAT\_Process(void)

#### Parameters

なし

#### Return Values

なし

#### Properties

[ECAT_SUCCESS]	/* スレーブスタックは正常終了したことを示します。 */
[ECAT_ERR_OTHER]	/* スレーブスタックを終了する必要があることを示します。*/

#### Description

EtherCAT スレーブスタックは SSC の MainLoop 関数により実行されます。

#### Example

```
/* Execute EtherCAT Slave Stack */  
R_ECAT_Process();
```

4. 端子設定

EtherCAT モジュールを使用するためには、マルチファンクションピンコントローラ（MPC）で周辺機能の入出力信号を端子に割り付ける（以下、端子設定と称す）必要があります。端子設定は、R\_ECATCH\_Open関数を呼び出す前に行ってください。

e<sup>2</sup> studio の場合は「FIT Configurator」または「Smart Configurator」の端子設定機能を使用することができます。FIT Configurator、Smart Configurator の端子設定機能を使用すると、端子設定画面で選択したオプションに応じて、ソースファイルが出力されます。そのソースファイルで定義された関数を呼び出すことにより端子を設定できます。詳細は表 4.1 を参照してください。

表 4.1 FIT Configurator が出力する関数一覧

使用マイコン	出力される関数名	備考
RX72M	R_ECATCH_PinSet_ESC R_ECATCH_PinSet_ESC_MII0 R_ECATCH_PinSet_ESC_MII1	



## 5. デモプロジェクト

デモプロジェクトには、FIT モジュールとそのモジュールが依存するモジュール（例：r\_bsp）を使用する main()関数が含まれます。本 FIT モジュールには以下のデモプロジェクトが含まれます。

---

### 5.1 ecat\_demo\_rskrx72m

---

ecat\_demo\_rskrx72m は、Renesas Starter Kit+ for RX72M ボード(以下 RSKRX72M ボード)で EtherCAT 通信を行うためのデモです。EtherCAT マスタから RSKRX72M ボードの LED を点灯することができます。

デモプログラムのセットアップ方法と動作手順について、以下に説明します。

#### 5.1.1 ボードの設定と接続

ボードの詳細情報に関しては、「Renesas Starter Kit+ for RX72M ユーザーズマニュアル」(以下 RSKRX72M ボードマニュアル)をご参照ください。

ボードの接続に関しては「RX72M グループ RSK ボード EtherCAT スタートアップマニュアル」(以下 RSKRX72M スタートアップマニュアル)をご参照ください。

#### 5.1.2 デモプロジェクト動作手順

デモプロジェクトの動作手順に関しては「RSKRX72M スタートアップマニュアル」をご参照ください。

---

### 5.2 ecat\_demo\_comrx72m

---

ecat\_demo\_comx72m は、RX72M 通信ボードで EtherCAT 通信を行うためのデモです。EtherCAT マスタから RX72M 通信ボードの LED を点灯することができます。

デモプログラムのセットアップ方法と動作手順について、以下に説明します。

#### 5.2.1 ボードの設定と接続

ボードの詳細情報に関しては、「RX72M グループ 通信ボードハードウェアマニュアル」(以下 COMRX72M ボードマニュアル)をご参照ください。

ボードの接続に関しては「RX72M グループ 通信ボード EtherCAT スタートアップマニュアル」(以下 COMRX72M スタートアップマニュアル)をご参照ください。

#### 5.2.2 デモプロジェクト動作手順

デモプロジェクトの動作手順に関しては「COMRX72M スタートアップマニュアル」をご参照ください。

## 6. 付録

## 6.1 動作確認環境

このセクションでは、EtherCAT モジュールの動作確認用の環境について説明します。

表 6.1 動作確認環境(Rev1.02)

項目	内容
統合開発環境 (IDE)	ルネサスエレクトロニクス製 e2 studio (V.7.5.0 以降) IAR Embedded Workbench for Renesas RX (V4.13.1 以降)
C コンパイラ	ルネサスエレクトロニクス製 C/C++ Compiler Package for RX Family (V3.01.00 以降) コンパイルオプション：統合開発環境のデフォルト設定に以下のオプションを追加 -lang = c99 GCC for Renesas RX (4.8.4.201803 以降) コンパイルオプション：統合開発環境のデフォルト設定に以下のオプションを追加 -std=gnu99
エンディアン	リトルエンディアン
モジュールのリビジョン	Rev.1.00
使用ボード	<ul style="list-style-type: none"> <li>● Renesas Starter Kit+ for RX72M (型名：RTK5572MNxCxxxxxBJ)</li> <li>● テセラ・テクノロジー社製 RX72M 搭載評価ボード (型名：TS-RX72M-COM)</li> </ul>

表 6.2 動作確認環境(Rev1.20)

項目	内容
統合開発環境 (IDE)	ルネサスエレクトロニクス製 e2 studio (V.7.8.0) IAR Embedded Workbench for Renesas RX (V4.20.1)
C コンパイラ	ルネサスエレクトロニクス製 C/C++ Compiler Package for RX Family (V3.02.00) コンパイルオプション：統合開発環境のデフォルト設定に以下のオプションを追加 -lang = c99 GCC for Renesas RX (8.3.0.202004) コンパイルオプション：統合開発環境のデフォルト設定に以下のオプションを追加 -std=gnu99
エンディアン	リトルエンディアン
モジュールのリビジョン	Rev.1.20
使用ボード	<ul style="list-style-type: none"> <li>● Renesas Starter Kit+ for RX72M (型名：RTK5572MNxCxxxxxBJ)</li> <li>● テセラ・テクノロジー社製 RX72M 搭載評価ボード (型名：TS-RX72M-COM)</li> </ul>

表 6.3 動作確認環境(Rev1.30)

項目	内容
統合開発環境 (IDE)	ルネサスエレクトロニクス製 e2 studio 2022-10 IAR Embedded Workbench for Renesas RX (V4.20.1)
C コンパイラ	ルネサスエレクトロニクス製 C/C++ Compiler Package for RX Family (V3.04.00) コンパイルオプション：統合開発環境のデフォルト設定に以下のオプションを追加 -lang = c99 GCC for Renesas RX (8.3.0.202202) コンパイルオプション：統合開発環境のデフォルト設定に以下のオプションを追加 -std=gnu99
エンディアン	リトルエンディアン
モジュールのレビジョン	Rev.1.30
使用ボード	<ul style="list-style-type: none"><li>● Renesas Starter Kit+ for RX72M (型名：RTK5572MNxCxxxxxBJ)</li><li>● テセラ・テクノロジー社製 RX72M 搭載評価ボード (型名：TS-RX72M-COM)</li><li>● RX72M CPU Card with RDC-IC (型名：RTK0EMXDE0C00000BJ)</li></ul>

## 6.2 e2 studio に EtherCAT FIT モジュールを組み込む方法

EtherCAT FIT モジュールは RX Driver パッケージに含まれていないためサンプルプロジェクト以外でスマートコンフィグレータのソフトウェアコンポーネントとして利用する場合は手で e2 studio に組み込む必要があります。

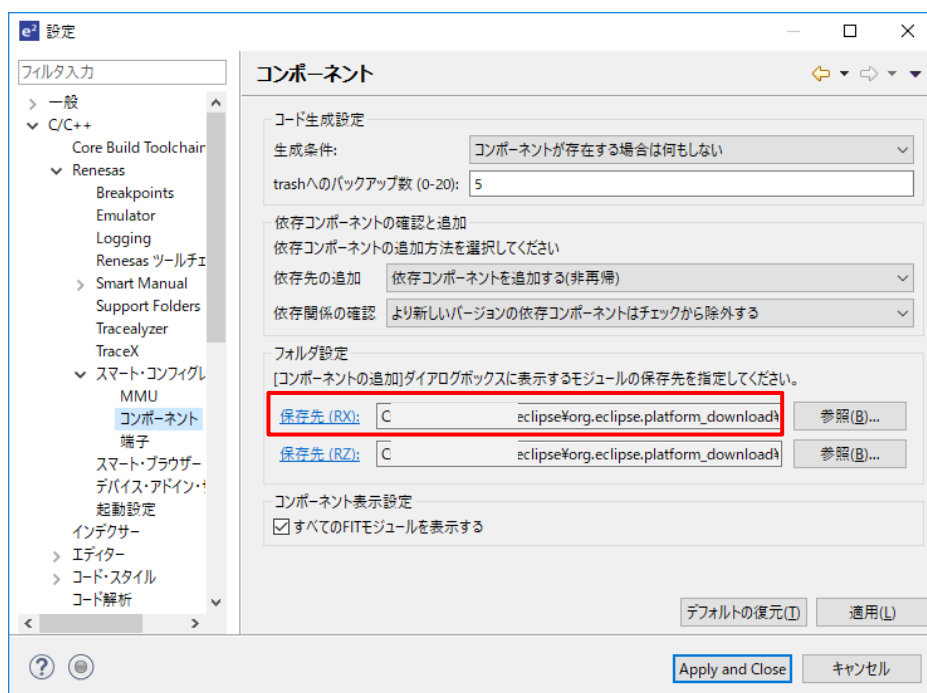
このセクションでは、EtherCAT FIT モジュールを e2studio の FIT モジュールの保存先フォルダにコピーする方法について説明します。

- e2studio の FIT モジュールの保存先フォルダ

「ウインドウ」→「設定」→設定ウインドウが開く

「C/CC+」→「Renesas」→「スマートコンフィグレータ」→「コンポーネント」を選択

フォルダ設定の保存先(RX):として表示されているフォルダになります。



- EtherCAT FIT モジュール

EtherCAT Slave サンプルプログラムパッケージの FITModules フォルダに格納されている次の 3 種類のファイルをコピーしてください。

CAT > Package > an-r01an4881xx0101-rx-ecat > FITModules		
名前		更新日時
r_ecat_rx_v1.00.xml		2019/08/26 13:44
r_ecat_rx_v1.00.zip		2019/11/01 13:25
r_ecat_rx_v1.00_extend.mdf		2019/08/22 16:53

【注】 EtherCAT FIT モジュールは最新のバージョンをご使用ください。

- e2 studio を再起動してスマートコンフィグレータのソフトウェアコンポーネントとして EtherCAT FIT モジュールが登録されていることを確認してください。

EtherCAT FIT モジュールは RX72M 通信ボードサンプルプログラムパッケージ内の EtherCAT Slave サンプルプログラムパッケージに含まれています。

- RX72M 通信ボードサンプルプログラムパッケージ  
r01an4882xxNNNN-rx72m-sample-package.zip
- EtherCAT Slave サンプルプログラムパッケージ  
an-r01an4881xxNNNN-rx-ecat.zip

なお、NNNN は各パッケージのリビジョンを表す 4 桁の数です。

- スマートコンフィグレータ使用する際の注意事項

スマートコンフィグレータで FIT モジュールのコードを生成すると"smc\_gen "フォルダは上書きされます。SSC ツールで生成した EtherCAT スレーブスタックコードを"r\_ecat\_rx "フォルダ下などに保存すると FIT モジュールのコード生成時に上書き消去されますので、ご注意ください。

"src¥application "フォルダのように"smc\_gen "フォルダと同階層のフォルダに保存することで上書き消去を回避することができます。

## 7. 参考ドキュメント

ユーザーズマニュアル：ハードウェア

RX72M グループ ユーザーズマニュアル ハードウェア編（ドキュメント No. R01UH0804）

Renesas Starter Kit+ for RX72M ユーザーズマニュアル（ドキュメント No. R20UT4383）

RX72M グループ 通信ボードハードウェアマニュアル（ドキュメント No. R01AN4661）

（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。）

スタートアップマニュアル

RX72M グループ RSK ボード EtherCAT スタートアップマニュアル（ドキュメント No. R01AN4689）

RX72M グループ通信ボード EtherCAT スタートアップマニュアル（ドキュメント No. R01AN4672）

（最新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。）

テクニカルアップデート／テクニカルニュース

（最新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。）

ユーザーズマニュアル：開発環境

RX ファミリ C/C++コンパイラ、アセンブラ、最適化リンケージエディタコンパイラパッケージ

（R20UT0570）

（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。）

## 改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	Aug.31.2019	—	初版発行
1.01	Nov.01.2019	デモプロジェクト	FITDemos フォルダに RSK ボードサンプルプロジェクト (ecat_demo_rskrx72m.zip)を追加
1.10	Apr.30.2020	17	表 6.1 動作確認環境」に「IAR Embedded Workbench for Renesas RX」を追加
		プログラム	EtherCAT スレーブスタックを FIT モジュール管理外とするために src/ssc 、src/appl フォルダを削除
			BSP モジュールマクロを使用する形式に CATSYNC0、CATSYNC1 の割り込みハンドラ定義を変更。合わせて CATSYNC1 の割り込みベクタに誤りがあったのを修正
1.11	Aug.31.2020	9	「2.12 FIT モジュールの追加方法」を手動で追加する方法に変更
		プログラム	スマートコンフィグレータによる CLKOUT25M 端子の設定を追加
1.20	Aug.31.2021	6	「2.7 コンパイル時の設定」コンフィギュレーションオプション(r_ecat_rx_config.h)に ECAT_CFG_USE_SUPPORTED_PHY の説明を追加
		プログラム	コンフィギュレーションに ECAT_CFG_USE_SUPORTED_PHY を追加
			PHY レジスタの初期設定処理 PHY.C に移動
			DIP SW の読み込み処理を sampleappl.c に移動
			ECAT_CFG_TX_SHIFT1 が TXSFT1 に設定されていなかった問題を修正
1.21	Feb. 01.2022	19-20	「6.2 e2 studio に EtherCAT FIT モジュールを組み込む方法」を追加
		プログラム	FIT モジュールの xml ファイルを変更
1.30	Jan.31.2023	4	「表 2.1 使用する割り込みベクター一覧」の ESC SYNC 割り込みベクタ番号を変更 ESC SYNC0 : 210→252 ESC SYNC1 : 211→253
		18	動作確認環境からビッグエンディアンを削除
		プログラム	SSC 5.13 に対応
			e2 studio 64 ビット版に対応
			Utilities フォルダ以下を削除。同フォルダに含まれていたファイルはサンプルプログラムで提供する

## 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、ノイズ幅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。

○Arm<sup>®</sup> およびCortex<sup>®</sup> は、Arm Limited（またはその子会社）のEUまたはその他の国における登録商標です。 All rights reserved.

○Ethernetおよびイーサネットは、富士ゼロックス株式会社の登録商標です。

○IEEEは、the Institute of Electrical and Electronics Engineers, Inc. の登録商標です。

○OTRONは” The Real-time Operation system Nucleus” の略称です。

○ITRONは” Industrial TRON” の略称です。

○ $\mu$ ITRONは” Micro Industrial TRON” の略称です。

○OTRON、ITRON、および $\mu$ ITRONは、特定の商品ないし商品群を指す名称ではありません。

○EtherCAT<sup>®</sup>, およびTwinCAT<sup>®</sup>は、ドイツBeckhoff Automation GmbHによりライセンスされた特許取得済み技術であり登録商標です。

○その他、本資料中の製品名やサービス名は全てそれぞれの所有者に属する商標または登録商標です。



## 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、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、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。
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。

6. 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。
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、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。
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。かかる法令を遵守しないこと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。
9. 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。
10. 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。
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。
12. 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.4.0-1 2017.11)

## 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](http://www.renesas.com)

## お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。

[www.renesas.com/contact/](http://www.renesas.com/contact/)

## 商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。すべての商標および登録商標は、それぞれの所有者に帰属します。