

【注意事項】

R20TS0307JJ0100

Rev.1.00

2018.05.16 号

RX ファミリ

イーサネットモジュール Firmware Integration Technology,

RX Driver Package

概要

タイトルに記載している製品の使用上の注意事項を連絡します。

1. “R_ETHER_LinkProcess” 関数におけるオートネゴシエーション未完了時のリンクアップ処理の注意事項
2. 割り込み関数内で “R_ETHER_Read_ZC2” 関数または “R_ETHER_Read” 関数を呼び出す場合の注意事項
3. “R_ETHER_Read_ZC2” 関数におけるエラーフレーム受信時の注意事項

1. “R_ETHER_LinkProcess” 関数におけるオートネゴシエーション未完了時のリンクアップ処理の注意事項

1.1 該当製品

本関数は、r_ether_rx_v*.**.zip (*.**はリビジョン番号) ファイルで提供しており、このファイルは以下の(1)と(2) の製品に同梱されています。

- (1) RX ファミリ イーサネットモジュール Firmware Integration Technology
(以下、イーサネット FIT モジュール)

該当するリビジョンとドキュメントは「表 1.1 イーサネット FIT モジュール該当製品一覧」のとおりです。

表 1.1 イーサネット FIT モジュール該当製品一覧

イーサネット FIT モジュールのリビジョン	資料番号
Rev.1.00	R01AN2009JJ0100
Rev.1.01	R01AN2009JJ0101
Rev.1.02	R01AN2009JJ0102
Rev.1.10	R01AN2009JJ0110
Rev.1.11	R01AN2009JJ0111
Rev.1.12	R01AN2009JJ0112
Rev.1.13	R01AN2009JJ0113
Rev.1.14	R01AN2009JJ0114

(2) RX Driver Package

(1)のイーサネット FIT モジュールは、下記 RX Driver Package にも同梱されています。該当する RX Driver Package の製品名、リビジョンおよび同梱しているイーサネット FIT モジュールのリビジョンは、「表 1.2 イーサネット FIT モジュール同梱製品一覧」のとおりです。

表 1.2 イーサネット FIT モジュール同梱製品一覧

RX Driver Package の製品名	RX Driver Package のリビジョン	資料番号	同梱しているイーサネット FIT モジュールのリビジョン
RX64M グループ RX Driver Package ユーザズマニュアル	Rev.1.01	R01AN2460JJ0101	Rev.1.00
RX64M, RX71M グループ RX Driver Package Ver.1.02	Rev.1.04	R01AN2606JJ0104	Rev.1.02
RX ファミリ RX Driver Package Ver.1.10	Rev.1.10	R01AN3345JJ0100	Rev.1.10
RX ファミリ RX Driver Package Ver.1.11	Rev.1.11	R01AN3467JJ0111	Rev.1.11
RX ファミリ RX Driver Package Ver.1.12	Rev.1.12	R01AN3651JJ0112	Rev.1.12
RX ファミリ RX Driver Package Ver.1.13	Rev.1.13	R01AN3859JJ0113	Rev.1.13
RX ファミリ RX Driver Package Ver.1.14	Rev.1.14	R01AN4191JJ0114	Rev.1.14

1.2 該当 MCU

RX63N、RX65N、RX64M、および RX71M グループ

1.3 内容

“R_ETHER_LinkProcess” 関数におけるリンクアップ処理実行後にイーサネットフレームを送受信できない場合があります。

1.4 発生条件

“R_ETHER_LinkProcess” 関数を実行してリンクアップ処理を行ったときに、PHY のオートネゴシエーションが完了していない場合に発生します。

➤ 発生例

PHY のオートネゴシエーションが完了していない場合、“ether_do_link” 関数で ETHERC の送受信機能が許可されないため、以降正常にイーサネットフレームを送受信できません。

```
void R_ETHER_LinkProcess (uint32_t channel)
{
    ----- (中略) -----

    ether_configure_mac(channel, mac_addr_buf[channel],
        NO_USE_MAGIC_PACKET_DETECT);
    ether_do_link(channel, NO_USE_MAGIC_PACKET_DETECT);

    transfer_enable_flag[channel] = ETHER_FLAG_ON;

    if ((NULL != cb_func.pcb_func) && (FIT_NO_FUNC !=
        cb_func.pcb_func))
    {
        cb_arg.channel = channel;
        cb_arg.event_id = ETHER_CB_EVENT_ID_LINK_ON;
        (*cb_func.pcb_func)((void *) &cb_arg);
    }

    ----- (中略) -----
}
```

PHY のオートネゴシエーションが完了していない場合、
ETHERC の送受信機能が許可されません。

1.5 回避策

“R_ETHER_LinkProcess” 関数において、以下の (1) および (2) の変更を行ってください。

- (1) イーサネット FIT モジュールの送受信有効フラグのセット処理を、“ether_do_link” 関数を呼び出す前に実行するように変更してください。
- (2) リンクアップ処理実行時に、PHY のオートネゴシエーションが完了していない場合があります。その場合の処理として、再度本関数を呼び出したときに、リンクアップ処理を実行するように変更してください。

以下に詳細を記載します。青文字の処理を赤文字の処理に変更してください。

変更前:

```
void R_ETHER_LinkProcess (uint32_t channel)
{
----- (中略) -----
    #if (ETHER_CFG_USE_LINKSTA == 1)
----- (中略) -----

        /*
         * ETHERC and EDMAC are set after ETHERC and EDMAC are reset in
         * software
         * and sending and receiving is permitted.
         */
        ether_configure_mac(channel, mac_addr_buf[channel],
            NO_USE_MAGIC_PACKET_DETECT);
        ether_do_link(channel, NO_USE_MAGIC_PACKET_DETECT);

        transfer_enable_flag[channel] = ETHER_FLAG_ON;

        if ((NULL != cb_func.pcb_func) && (FIT_NO_FUNC !=
            cb_func.pcb_func))
        {
            cb_arg.channel = channel;
            cb_arg.event_id = ETHER_CB_EVENT_ID_LINK_ON;
            (*cb_func.pcb_func)((void *) &cb_arg);
        }

----- (中略) -----
    #elif (ETHER_CFG_USE_LINKSTA == 0)
----- (中略) -----

        /*
         * ETHERC and EDMAC are set after ETHERC and EDMAC are reset in
         * software
         * and sending and receiving is permitted.
         */
        ether_configure_mac(channel, mac_addr_buf[channel],
            NO_USE_MAGIC_PACKET_DETECT);
        ether_do_link(channel, NO_USE_MAGIC_PACKET_DETECT);

        transfer_enable_flag[channel] = ETHER_FLAG_ON;

        if ((NULL != cb_func.pcb_func) && (FIT_NO_FUNC !=
            cb_func.pcb_func))
        {
```

```

        cb_arg.channel = channel;
        cb_arg.event_id = ETHER_CB_EVENT_ID_LINK_ON;
        (*cb_func.pcb_func)((void *) &cb_arg);
    }

    ----- (中略) -----
}

```

変更後:

```

void R_ETHER_LinkProcess (uint32_t channel)
{
    ----- (中略) -----

    #if (ETHER_CFG_USE_LINKSTA == 1)
    ----- (中略) -----

    transfer_enable_flag[channel] = ETHER_FLAG_ON;

    /*
     * ETHERC and EDMAC are set after ETHERC and EDMAC are reset in
     * software
     * and sending and receiving is permitted.
     */
    ether_configure_mac(channel, mac_addr_buf[channel],
        NO_USE_MAGIC_PACKET_DETECT);
    ret = ether_do_link(channel, NO_USE_MAGIC_PACKET_DETECT);
    if (ETHER_SUCCESS == ret)
    {
        if ((NULL != cb_func.pcb_func) && (FIT_NO_FUNC !=
            cb_func.pcb_func))
        {
            cb_arg.channel = channel;
            cb_arg.event_id = ETHER_CB_EVENT_ID_LINK_ON;
            (*cb_func.pcb_func)((void *) &cb_arg);
        }
    }
    else
    {
        /* When PHY auto-negotiation is not completed */
        transfer_enable_flag[channel] = ETHER_FLAG_OFF;
        lchng_flag[channel] = ETHER_FLAG_ON_LINK_ON;
    }
}

```

回避策(1):
イーサネット FIT モジュールの送受信有効フラグのセット処理を、
“ether_do_link” 関数を呼び出す前に実行するように変更します。

回避策(2):
PHY のオートネゴシエーションが完了していない場合、再度、本関数を
呼び出したときに、リンクアップ処理を実行するように変更します。

```

----- (中略) -----
#elif (ETHER_CFG_USE_LINKSTA == 0)
----- (中略) -----

transfer_enable_flag[channel] = ETHER_FLAG_ON;

/*
 * ETHERC and EDMAC are set after ETHERC and EDMAC are reset in
 * software
 * and sending and receiving is permitted.
 */
ether_configure_mac(channel, mac_addr_buf[channel],
NO_USE_MAGIC_PACKET_DETECT);
ret = ether_do_link(channel, NO_USE_MAGIC_PACKET_DETECT);
if (ETHER_SUCCESS == ret)
{
    if ((NULL != cb_func.pcb_func) && (FIT_NO_FUNC !=
cb_func.pcb_func))
    {
        cb_arg.channel = channel;
        cb_arg.event_id = ETHER_CB_EVENT_ID_LINK_ON;
        (*cb_func.pcb_func)((void *) &cb_arg);
    }
}
else
{
    /* When PHY auto-negotiation is not completed */
    transfer_enable_flag[channel] = ETHER_FLAG_OFF;
    lchng_flag[channel] = ETHER_FLAG_ON_LINK_ON;
}
}

```

回避策(1) :
イーサネット FIT モジュールの送受信有効フラグのセット処理を、“ether_do_link” 関数を呼び出す前に実行するように変更します。

回避策(2) :
PHY のオートネゴシエーションが完了していない場合、再度、本関数を呼び出したときに、リンクアップ処理を実行するように変更します。

1.6 恒久対策

(1) イーサネット FIT モジュール

次期リリース予定の Rev.1.15 で改修します。

(2) RX Driver Package

次期リリース予定の RX ファミリ RX Driver Package Ver.1.15 に、本注意事項を改修したイーサネット FIT モジュール Rev.1.15 を同梱する予定です。

2. 割り込み関数内で“R_ETHER_Read_ZC2”関数または“R_ETHER_Read”関数を呼び出す場合の注意事項

2.1 該当製品

本関数は、r_ether_rx_v*.**.zip (*.**はリビジョン番号) ファイルで提供しており、このファイルは以下の(1)と(2)の製品に同梱されています。

- (1) RX ファミリ イーサネットモジュール Firmware Integration Technology
(以下、イーサネット FIT モジュール)

該当するリビジョンとドキュメントは「表 2.1 イーサネット FIT モジュール該当製品一覧」のとおりです。

表 2.1 イーサネット FIT モジュール該当製品一覧

イーサネット FIT モジュールのリビジョン	資料番号
Rev.1.00	R01AN2009JJ0100
Rev.1.01	R01AN2009JJ0101
Rev.1.02	R01AN2009JJ0102
Rev.1.10	R01AN2009JJ0110
Rev.1.11	R01AN2009JJ0111
Rev.1.12	R01AN2009JJ0112
Rev.1.13	R01AN2009JJ0113
Rev.1.14	R01AN2009JJ0114

- (2) RX Driver Package

(1)のイーサネット FIT モジュールは、下記 RX Driver Package に同梱されています。該当する RX Driver Package の製品名、リビジョン、および同梱しているイーサネット FIT モジュールのリビジョンは、「表 2.2 イーサネット FIT モジュール同梱製品一覧」のとおりです。

表 2.2 イーサネット FIT モジュール同梱製品一覧

RX Driver Package の製品名	RX Driver Package のリビジョン	資料番号	同梱しているイーサネット FIT モジュールのリビジョン
RX64M グループ RX Driver Package ユーザズマニュアル	Rev.1.01	R01AN2460JJ0101	Rev.1.00
RX64M, RX71M グループ RX Driver Package Ver.1.02	Rev.1.04	R01AN2606JJ0104	Rev.1.02
RX ファミリ RX Driver Package Ver.1.10	Rev.1.10	R01AN3345JJ0100	Rev.1.10
RX ファミリ RX Driver Package Ver.1.11	Rev.1.11	R01AN3467JJ0111	Rev.1.11
RX ファミリ RX Driver Package Ver.1.12	Rev.1.12	R01AN3651JJ0112	Rev.1.12
RX ファミリ RX Driver Package Ver.1.13	Rev.1.13	R01AN3859JJ0113	Rev.1.13
RX ファミリ RX Driver Package Ver.1.14	Rev.1.14	R01AN4191JJ0114	Rev.1.14

2.2 該当 MCU

RX63N、RX65N、RX64M、および RX71M グループ

2.3 内容

割り込み関数内で“R_ETHER_Read_ZC2”関数または“R_ETHER_Read”関数を呼び出している場合、正常にイーサネットフレームを受信できない場合があります。

2.4 発生条件

“R_ETHER_LinkProcess”関数において、“ether_do_link”関数で ETHERC の送受信を有効にした後、イーサネット FIT モジュールの送受信有効フラグをセットする前に、イーサネットフレームの受信が完了した場合に発生します。

➤ 発生例

以下の赤破線の区間で発生した割り込みによる割り込み関数内で“R_ETHER_Read_ZC2”関数または“R_ETHER_Read”関数を呼び出しても正常にイーサネットフレームを受信できません。

```
void R_ETHER_LinkProcess (uint32_t channel)
{
    ----- (中略) -----
    ether_configure_mac(channel, mac_addr_buf[channel],
        NO_USE_MAGIC_PACKET_DETECT);
    ether_do_link(channel, NO_USE_MAGIC_PACKET_DETECT);
    -----
    transfer_enable_flag[channel] = ETHER_FLAG_ON;
    -----
    if ((NULL != cb_func.pcb_func) && (FIT_NO_FUNC !=
        cb_func.pcb_func))
    {
        cb_arg.channel = channel;
        cb_arg.event_id = ETHER_CB_EVENT_ID_LINK_ON;
        (*cb_func.pcb_func)((void *) &cb_arg);
    }
    ----- (中略) -----
}
```

2.5 回避策

イーサネット FIT モジュールの送受信有効フラグのセット処理を、“ether_do_link”関数を呼び出す前に実行するように変更してください。本回避策は 1.5 回避策 (1)エラー! 参照元が見つかりません。と同一です。詳細については、1.5 項を参照してください。

2.6 恒久対策

(1) イーサネット FIT モジュール

次期リリース予定の Rev.1.15 で改修します。

(2) RX Driver Package

次期リリース予定の RX ファミリ RX Driver Package Ver.1.15 に、本注意事項を改修したイーサネット FIT モジュール Rev.1.15 を同梱する予定です。

3. “R_ETHER_Read_ZC2” 関数におけるエラーフレーム受信時の注意事項

3.1 該当製品

本関数は、r_ether_rx_v*.**.zip（*.**はリビジョン番号）ファイルで提供しており、このファイルは以下の(1)と(2)の製品に同梱されています。

(1) RX ファミリ イーサネットモジュール Firmware Integration Technology

(以下、イーサネット FIT モジュール)

該当するリビジョンとドキュメントは「表 3.1 イーサネット FIT モジュール該当製品一覧」のとおりです。

表 3.1 イーサネット FIT モジュール該当製品一覧

イーサネット FIT モジュールのリビジョン	資料番号
Rev.1.00	R01AN2009JJ0100
Rev.1.01	R01AN2009JJ0101
Rev.1.02	R01AN2009JJ0102
Rev.1.10	R01AN2009JJ0110
Rev.1.11	R01AN2009JJ0111
Rev.1.12	R01AN2009JJ0112
Rev.1.13	R01AN2009JJ0113
Rev.1.14	R01AN2009JJ0114

(2) RX Driver Package

(1)のイーサネット FIT モジュールは、下記 RX Driver Package に同梱されています。該当する RX Driver Package の製品名、リビジョンおよび同梱しているイーサネット FIT モジュールのリビジョンは、「表 3.2 イーサネット FIT モジュール同梱製品一覧」のとおりです。

表 3.2 イーサネット FIT モジュール同梱製品一覧

RX Driver Package の製品名	RX Driver Package のリビジョン	資料番号	同梱しているイーサネット FIT モジュールのリビジョン
RX64M グループ RX Driver Package ユーザーズマニュアル	Rev.1.01	R01AN2460JJ0101	Rev.1.00
RX64M, RX71M グループ RX Driver Package Ver.1.02	Rev.1.04	R01AN2606JJ0104	Rev.1.02
RX ファミリ RX Driver Package Ver.1.10	Rev.1.10	R01AN3345JJ0100	Rev.1.10
RX ファミリ RX Driver Package Ver.1.11	Rev.1.11	R01AN3467JJ0111	Rev.1.11
RX ファミリ RX Driver Package Ver.1.12	Rev.1.12	R01AN3651JJ0112	Rev.1.12
RX ファミリ RX Driver Package Ver.1.13	Rev.1.13	R01AN3859JJ0113	Rev.1.13
RX ファミリ RX Driver Package Ver.1.14	Rev.1.14	R01AN4191JJ0114	Rev.1.14

3.2 該当 MCU

RX63N、RX65N、RX64M、および RX71M グループ

3.3 内容

“R_ETHER_Read_ZC2” 関数の処理が正常に完了しない場合があります。

3.4 発生条件

“R_ETHER_Read_ZC2” 関数を実行中に、以下の (1) および (2) の条件を満たした場合に発生します。

- (1) 割り込み関数内で実行している “R_ETHER_LinkProcess” 関数でリンクダウン処理を実行している。
- (2) イーサネットフレームにおけるエラーフレームを受信している。

➤ 発生例

“R_ETHER_Read_ZC2” 関数の下位関数である “R_ETHER_Read_ZC2_BufRelease” 関数の処理でエラーを返却するため、再度リンクアップするまで、ループ処理を実行します。そのため、“R_ETHER_Read_ZC2” 関数の処理が正常に完了しません。

```
int32_t R_ETHER_Read_ZC2 (uint32_t channel, void **pbuf)
{
    ----- (中略) -----
    while (ETHER_SUCCESS != complete_flag)
    {
        if (RACT != (papp_rx_desc[channel]->status & RACT))
        {
            if (ETHER_MC_FILTER_ON == mc_filter_flag[channel])
            {
                if (RFS7_RMAF == (papp_rx_desc[channel]->status &
                    RFS7_RMAF))
                {
                    R_ETHER_Read_ZC2_BufRelease(channel);

                    ret = ETHER_ERR_MC_FRAME;
                    complete_flag = ETHER_SUCCESS;
                }
            }
            if (ETHER_ERR_MC_FRAME != ret)
            {
                if (RFE == (papp_rx_desc[channel]->status & RFE))
                {
                    R_ETHER_Read_ZC2_BufRelease(channel);
                }
            }
        }
    }
}
```

リンクダウンしている場合、“R_ETHER_Read_ZC2_BufRelease” 関数がエラーを返却するため処理が完了しません。

```

        else
        {
            (*pbuf) = (void *) papp_rx_desc[channel]->buf_p;

            num_recvd = papp_rx_desc[channel]->size;
            ret = num_recvd;
            complete_flag = ETHER_SUCCESS;
        }
    }
else
{
    ret = ETHER_NO_DATA;
    complete_flag = ETHER_SUCCESS;
}
}

----- (中略) -----
} /* End of function R_ETHER_Read_ZC2() */

```

3.5 回避策

“R_ETHER_Read_ZC2_BufRelease” 関数でエラーが返却された場合は、本関数の実行を終了するように “R_ETHER_Read_ZC2” 関数を変更してください。

以下に詳細を記載します。

(1) 赤文字の処理を追加してください。

変更前:

```

int32_t R_ETHER_Read_ZC2 (uint32_t channel, void **pbuf)
{
    int32_t num_recvd;
    int32_t ret;
    int32_t complete_flag;

```

変更後:

```

int32_t R_ETHER_Read_ZC2 (uint32_t channel, void **pbuf)
{
    int32_t num_recvd;
    int32_t ret;
    int32_t complete_flag;
    int32_t ret2;

```

(2) 青文字の処理を赤文字の処理に変更してください。

変更前:

```
int32_t R_ETHER_Read_ZC2 (uint32_t channel, void **pbuf)
{
    ----- (中略) -----

    if (ETHER_MC_FILTER_ON == mc_filter_flag[channel])
    {
        if (RFS7_RMAF == (papp_rx_desc[channel]->status &
            RFS7_RMAF))
        {
            R_ETHER_Read_ZC2_BufRelease(channel);

            ret = ETHER_ERR_MC_FRAME;
            complete_flag = ETHER_SUCCESS;
        }
    }

    if (ETHER_ERR_MC_FRAME != ret)
    {
        if (RFE == (papp_rx_desc[channel]->status & RFE))
        {
            R_ETHER_Read_ZC2_BufRelease(channel);
        }
        else
        {
            (*pbuf) = (void *) papp_rx_desc[channel]->buf_p;

            num_recvd = papp_rx_desc[channel]->size;
            ret = num_recvd;
            complete_flag = ETHER_SUCCESS;
        }
    }

    ----- (中略) -----
} /* End of function R_ETHER_Read_ZC2() */
```

変更後:

```
int32_t R_ETHER_Read_ZC2 (uint32_t channel, void **pbuf)
{
    ----- (中略) -----
    if (ETHER_MC_FILTER_ON == mc_filter_flag[channel])
    {
        if (RFS7_RMAF == (papp_rx_desc[channel]->status &
            RFS7_RMAF))
        {
            ret2 = R_ETHER_Read_ZC2_BufRelease(channel);
            if (ETHER_SUCCESS != ret2)
            {
                return ret2;
            }

            ret = ETHER_ERR_MC_FRAME;
            complete_flag = ETHER_SUCCESS;
        }
    }

    if (ETHER_ERR_MC_FRAME != ret)
    {
        if (RFE == (papp_rx_desc[channel]->status & RFE))
        {
            ret2 = R_ETHER_Read_ZC2_BufRelease(channel);
            if (ETHER_SUCCESS != ret2)
            {
                return ret2;
            }
        }
        else
        {
            (*pbuf) = (void *) papp_rx_desc[channel]->buf_p;

            num_recvd = papp_rx_desc[channel]->size;
            ret = num_recvd;
            complete_flag = ETHER_SUCCESS;
        }
    }

    ----- (中略) -----
} /* End of function R_ETHER_Read_ZC2() */
```

3.6 恒久対策

(1) イーサネット FIT モジュール

次期リリース予定の Rev.1.15 で改修します。

(2) RX Driver Package

次期リリース予定の RX ファミリ RX Driver Package Ver.1.15 に、本注意事項を改修したイーサネット FIT モジュール Rev.1.15 を同梱する予定です。

以上

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2018.05.16	-	新規発行

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）

■総合お問い合わせ先

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

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。

過去のニュース内容は発行当時の情報をもとにしており、現時点では変更された情報や無効な情報が含まれている場合があります。

ニュース本文中の URL を予告なしに変更または中止することがありますので、あらかじめご承知ください。

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。