

お客様各位

---

## カタログ等資料中の旧社名の扱いについて

---

2010年4月1日を以ってNECエレクトロニクス株式会社及び株式会社ルネサステクノロジが合併し、両社の全ての事業が当社に承継されております。従いまして、本資料中には旧社名での表記が残っておりますが、当社の資料として有効ですので、ご理解の程宜しくお願ひ申し上げます。

ルネサスエレクトロニクス ホームページ (<http://www.renesas.com>)

2010年4月1日  
ルネサスエレクトロニクス株式会社

【発行】ルネサスエレクトロニクス株式会社 (<http://www.renesas.com>)

【問い合わせ先】 <http://japan.renesas.com/inquiry>

## ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。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。
3. 当社製品を改造、改変、複製等しないでください。
4. 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。
5. 輸出に際しては、「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。本資料に記載されている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事用途の目的で使用しないでください。また、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に使用することができません。
6. 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、誤りが無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。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されてい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生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。  
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  
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、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）  
特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社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。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、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、社会的損害など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、機器またはシステムとしての出荷保証をお願いいたします。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、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、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、当社は、一切その責任を負いません。
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固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

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。

# H8/300L SLP シリーズ

## I/O ポートを使用した SCI のエミュレーション (SCI ポート)

---

### 要旨

アプリケーションによっては、様々な外部デバイスとのマルチチャネル通信が必要になります。シリアルポートを使用する方法で、容易に実現することができます。しかし、使用できるシリアルポートが制限されるため、I/O ポートを使用した通信を実行する必要があります。

本アプリケーションノートでは、2本の I/O 線を使用した調歩同期式通信を実行します。PC との送受信のリンクは 9600bps に設定しました。

### 動作確認デバイス

H8/38024 SLP

### 目次

1. 理論 .....	2
2. 動作説明 .....	5
3. プログラムリスト .....	10

## 1. 理論

### 1.1 概要

シミュレーションされたシリアル通信インタフェースのソフトウェアは、移植性を容易にするため、C言語でプログラムされます。本アプリケーションノートでは、H8/38024 で動作確認しました。

本アプリケーションノートで使用する UART プロトコルは、以下に示す 1 スタートビット、8 データビット、パリティビットなし、1 ストップビットです。

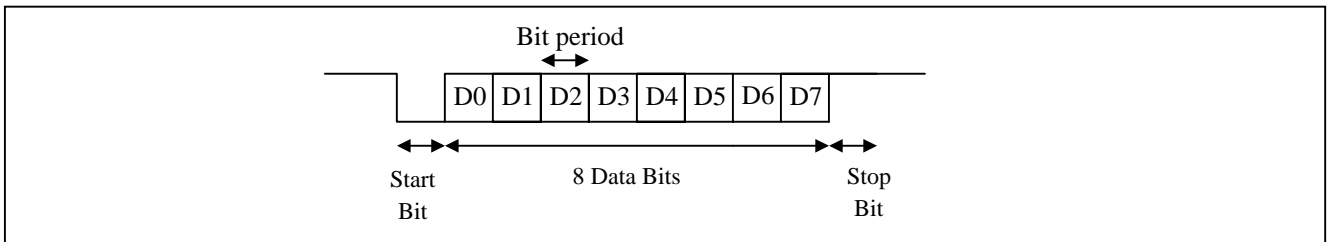


図 1 UART プロトコル

データを正確に送受信するために、ビット周期は正確でなくてはなりません。わずかな変動であってもそれが蓄積されるとタイミングエラーとなり、データが誤ってデコードされます。ビット周期は次のように計算されます。

$$\text{ビット周期} = 1 / \text{ボーレート}$$

9600bps のボーレートの場合、ビット周期が以下になります。

$$\text{ビット周期} = 1 / 9600 = 104.167 \mu\text{s}$$

このビット周期を生成するために、タイマ機能または for ループのどちらかを使用します。タイマレジスタ値および for ループには正確な値を使用してください。

送信には出力ポートピンを High にプルアップします。ビット“0”がスタートビットとして送られ、その後に 8 ビットのデータが続いて送られます。LSB (D0) が最初に送られ、最後にビット“1”がストップビットとして送られます。

受信ポートピンも同様に、High にプルアップします。信号レベルが Low になると、スタートビットを受信するか、または不要ノイズが電圧レベルを低下させるかのいずれかが発生します。したがって、サンプリングポイントを  $\frac{1}{2}$  ビット周期に遅延させ、スタートビット“0”の検証を行います。

### 1.2 動作

本アプリケーションノートでは H8/38024F の CPU ボードを使用します。ポート 1 の 6 ピン (P16) は送信チャンネルに使用し、4 ピン (P4) は外部からのシリアルデータの受信に使用します。使用する水晶発振子の周波数は 9.8304MHz、ボーレートは 9600bps です。これらはユーザの必要条件を満たすように、C プログラム内で容易に変更できます。

送信には、P16 は MOS プルアップとし、出力に設定し送信に使われます。データは、送信サブルーチン呼び出すことで送信されます。

P14 は、MOS プルアップとし、入力に使われます。この I/O ピンは IRQ4 割り込みの入力にも使用します。

データの受信を開始する前に、P14 を外部割り込み IRQ4 受信用に設定します。IRQ4 割り込みは、High から Low へのエッジトリガに初期設定しています。したがって、スタートビット受信後、P14 が High から Low へ変化すると、割り込みが発生して、受信動作である IRQ4 割り込みサービスルーチンを開始します。

割り込みサービスルーチンでは、P14 を連続データ受信のための入力ポートとして設定します。IRQ4 割り込みが発生すると、ソフトウェアは $\frac{1}{2}$ ビット周期をウェイトし、スタートビットをサンプリングします。スタートビットを検出後、受信サブルーチンは1ビット周期をウェイトし、8ビットのデータをそれぞれサンプリングします。このプロセスを図2に示します。

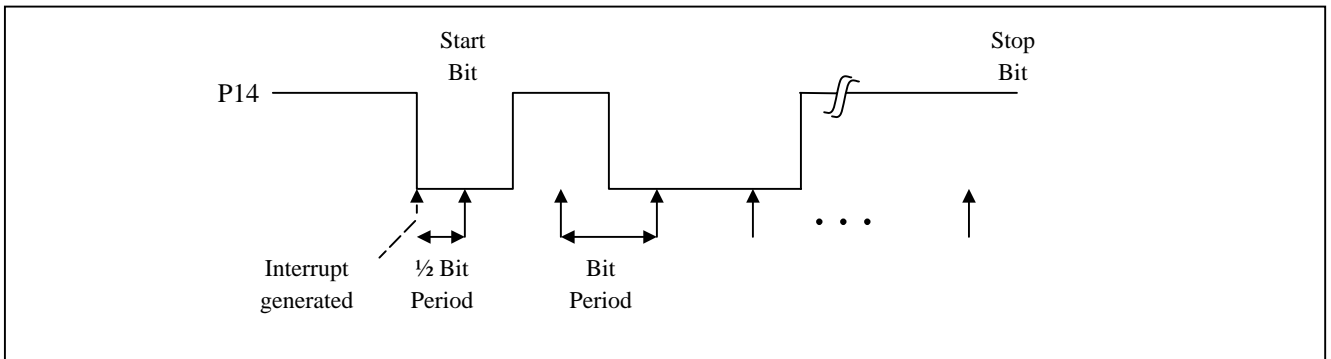


図2 サンプリング周期

タイマFは、ビット周期のウェイトに使用されます。タイマFは、16ビットタイマとして初期設定され、コンペアマッチが発生すると割り込みを発生します。アウトプットコンペアレジスタの値の計算は、次の1.3章で説明します。

### 1.3 アウトプットコンペアレジスタ値の計算

タイマFは、アウトプットコンペア機能内蔵のフリーランニングカウンタです。コンペアマッチが発生すると、割り込みが生成されるように初期設定されます。つまり、タイマコントロールレジスタF (TCRF) が値をインクリメントさせ、タイマコントロールレジスタFの値とアウトプットコンペアレジスタFH (OCR FH) の値が一致すると、割り込みが発生します。

タイマコントロールレジスタFをビット2~0に設定することで、表1の太字に示すように、内部クロックを $\phi/4$ に設定します。

表1 タイマFのクロック選択

ビット2 CKSL2	ビット1 CKSL1	ビット0 CKSL0	説明
0	0	0	外部イベント(TMIF) を立ち上がり/立ち下りエッジでカウント
0	0	1	
0	1	0	
0	1	1	使用禁止
1	0	0	内部クロック： $\phi/32$ でカウント
1	0	1	内部クロック： $\phi/16$ でカウント
<b>1</b>	<b>1</b>	<b>0</b>	<b>内部クロック：<math>\phi/4</math> でカウント</b>
1	1	1	内部クロック： $\phi_w/4$ でカウント

アウトプットコンペアレジスタ値は、以下のように計算されます。

$$\begin{aligned}
 & \text{ビット周期} = 1 / \text{ボーレート} \\
 & \emptyset = \text{水晶発振子の周波数} / 2 \\
 & \text{内部クロック} = \emptyset / 4 \\
 & d = \text{アウトプットコンペアレジスタ値} \\
 & d \times \text{内部クロック周期} = \text{必要なビット周期} \\
 & D \times \frac{1}{\text{水晶発振子の周波数}} = \frac{1}{\text{ビット周期}} \\
 & \qquad \qquad \qquad \frac{2 \times 4}{\text{水晶発振子の周波数}} \\
 & d = \frac{\text{水晶発振子の周波数}}{8 \times \text{ボーレート}}
 \end{aligned}$$

タイマ F は、初期化され 16 ビットタイマカウンタ TCF の値のインクリメントを開始させるまで、約 25 $\mu$ s の時間を要します。それがより長いビット周期の原因になり、値をオフセットにする必要があります。9.3204MHz の水晶発振子が使用される場合、49 のオフセットが必要です。

したがって、OCR FH にロードされる値は、 $d - 49$  です。

注意：初期設定時のタイマ F の遅延は、使用される水晶発振子の値に依存します。異なる水晶発振子を使用する場合、ユーザはオフセット値を変更する必要があります。

#### 1.4 その他の解決策

- スタートビットのポーリング**  
 スタートビットを検出するために割り込みを使用せずに、ユーザはポーリング方法を使用することができます。スタートビットを待ち受けするために、ユーザは継続して受信ピンである P14 を読み出します。コード例は以下のとおりです。

```

while(1)
{
    if (RX == 0)
        receive();
}

```

しかしこの方法は、ユーザは受信ピンをチェックし続ける必要があるため、メインプログラムが他に何も実行しない場合にだけ使用できます。

- for ループを使用した遅延**  
 ビット周期に必要な遅延を実行するために、タイマ F に代わり for ループが使用できます。ユーザは for ループに適切な遅延値を設定する必要があります。遅延サブルーチンの例は以下のとおりです。

```

void delay (unsigned short d)
{
    for (i = 0; i < d; i ++ )
    }
}

```

## 2. 動作説明

### 2.1 ハードウェアの設定

外部デバイスと通信するために RS-232C ドライバの接続を設定してください。マイコンと外部デバイス間の信号レベルを調整するために簡易シリアルドライバを設定する必要があります。I/O ピンは 10k の抵抗で High にプルアップしてください。

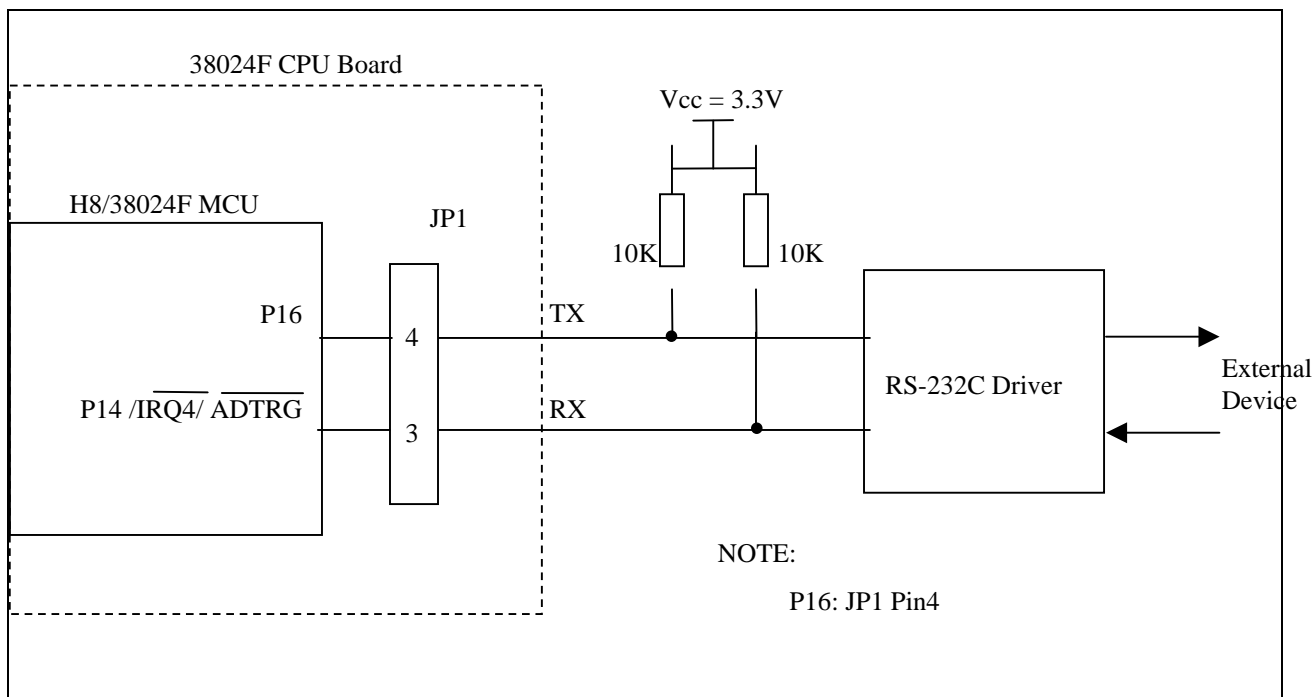


図3 RS - 232C シリアルドライバとマイコンの設定

RS - 232C ドライバに関する回路図を図 4 に示します。

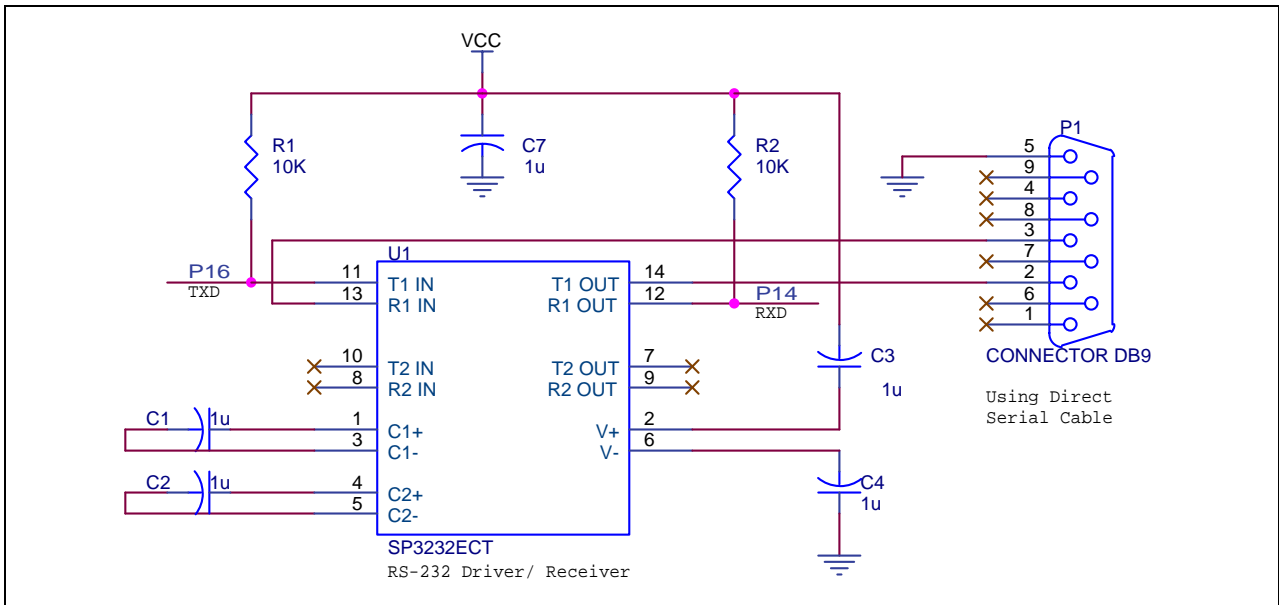


図 4 RS-232C シリアルドライバ

## 2.2 ハイパーターミナルの設定

ユーザがマイコンとハイパーターミナルを使用した PC との間に通信を設定するときは、UART プロトコルとプログラムに使用されるボーレートに従い、COM ポートを設定する必要があります。

9600bps のボーレートの場合、RS-232 ドライバに接続した COM ポートを以下のように設定する必要があります。ハイパーターミナルウィンドウの File メニューで Properties を選択し Configure... をクリックすると、ポート設定を変更します。



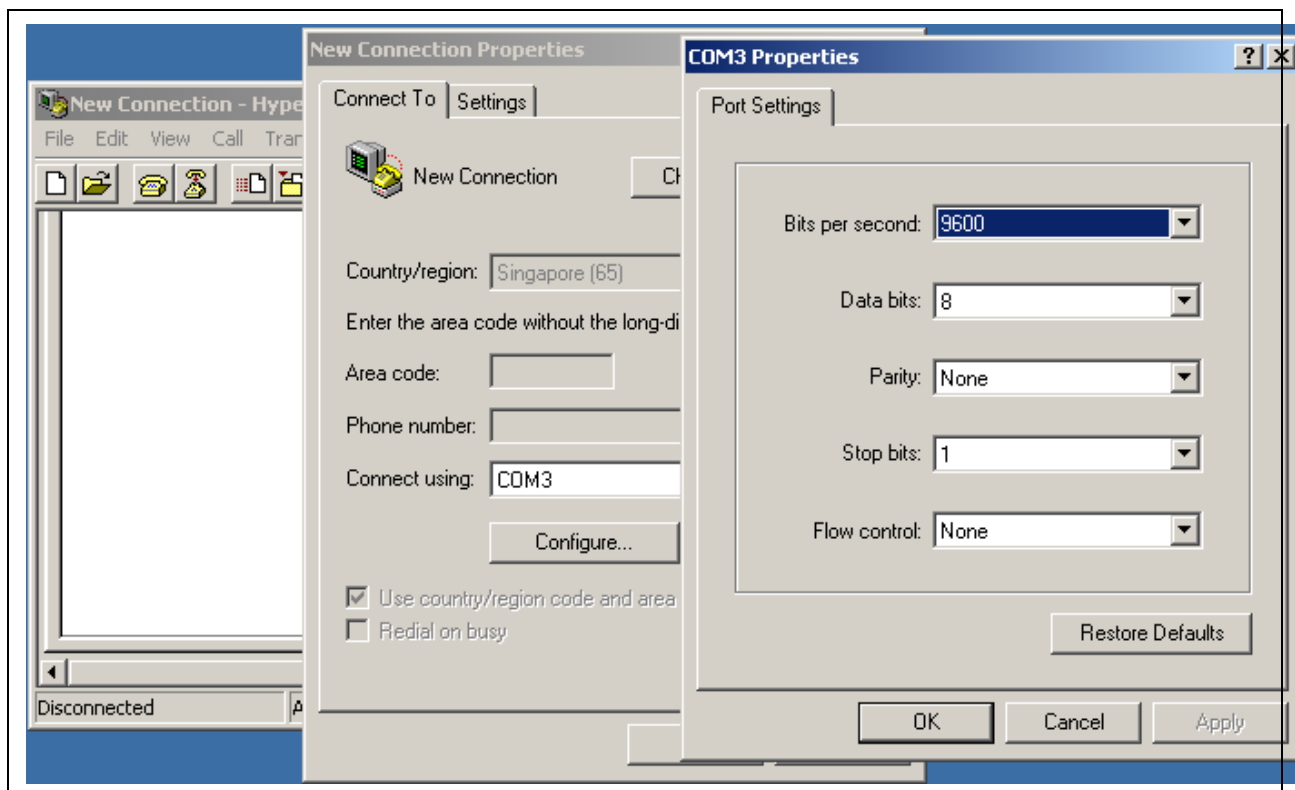


図 5 PC ハイパーターミナルの設定

次に、ユーザによって入力された文字がハイパーターミナルウインドウで表示されるように、ハイパーターミナルを設定します。設定タブにある *ASCII Setup...* をクリックし、図 6 に示すように、*Echo typed characters locally* のチェックボックスをクリックしてください。

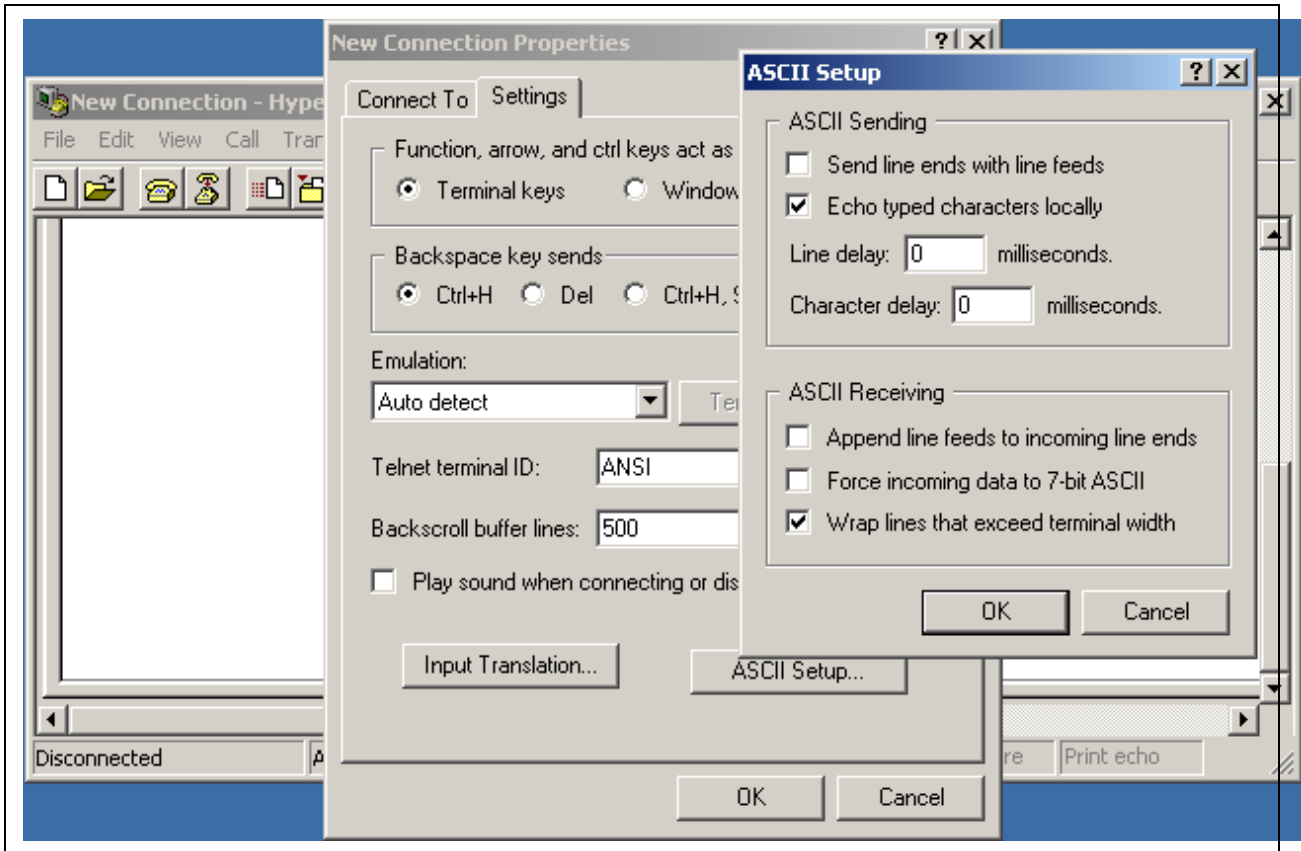


図 6 ASCII の設定

正しいボーレートを設定すると、ハイパーターミナルに文字 Test が表示されます。ユーザはハイパーターミナルにどのような文字も入力することができ、デコードされた文字は確認のためハイパーターミナルウインドウに再度送信されます。図 7 にハイパーターミナルウインドウの例を示します。文字が間違っデコードされた場合(ストップビットが検出されない場合)、Er が送信されハイパーターミナルウインドウに表示されます。

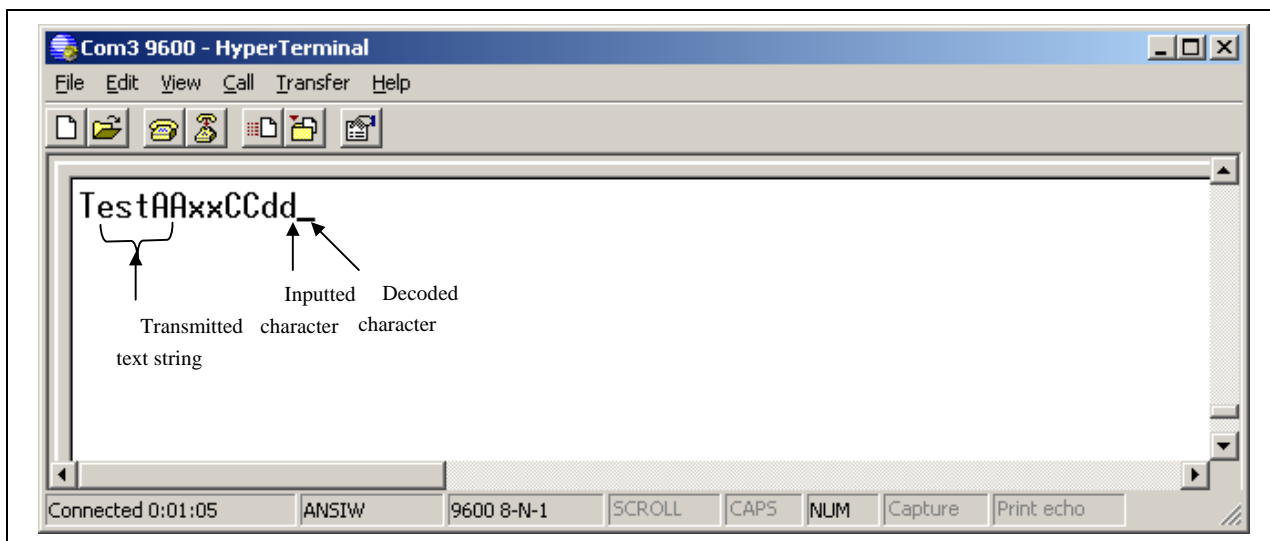


図 7 PC ハイパ ターミナルウィンドウの表示例

### 3. プログラムリスト

本アプリケーションノート以下のプログラムリストは、H8/38024 マイコンを対象とする HEW プロジェクトジェネレータを使用して作成しています。使用するツールチェーンは、SLP/TINY ツールチェーンです。

メイン機能を解説しユーザに更に理解してもらうために、フローチャートも作成しています。

注意：プログラムが正しく動作するように、最適化をオフにしてください。

```

/*****
/*
/* FILE      :Emulate_SCI.c
/* DATE      :Wed, Sep 10, 2003
/* DESCRIPTION :Main Program
/* CPU TYPE  :H8/38024F
/*
/*
/* This file is generated by Renesas Project Generator (Ver.2.1).
/*
/*
*****/

//include 38024F IO define header file
#include "iodefine.h"

#include <machine.h>
#include <_h_c_lib.h>

//include additional flag define header file
#include "flagdefine.h"

void initialize (void);
void transmit (unsigned char);
void receive (void);
void transmit_string (void);
void delay (unsigned short);

char *buff_ptr;
static const char TX_buffer[] = "Test"; // Transmit buffer
char RX_buffer; // Receive buffer
unsigned int i =0;

//----- Main Program -----//
void main(void)
{
    initialize();

    transmit_string();

    while(1);
}

//----- Initialization of Port 1, Timer F & IRQ4 -----//
void initialize (void)
{
// Initialize Port 1
    P_IO.PUCR1.BIT.PUCR16 = 1; // P16 MOS pull-up
    TX = 1;
    P_IO.PCR1.BIT.PCR16 = 1; // P16 as output (TX)

// Initialize IRQ4
    P_SYSCR.IEGR.BIT.IEG4 = 0; // Interrupt generated at falling edge of
//IRQ4
    P_SYSCR.IENR1.BIT.IEN4 = 1; // Enables IRQ4
    P_IO.PMR1.BIT.IRQ4 = 1; // P14 used as IRQ4

// Initialize Timer F
    P_TMRF.TCRF.BYTE = 0x8E; // Set TMOFH pin output level to HIGH and
//internal clock of o/4

```

```

P_TMRF.TCSRFBIT.CCLRH = 1;           //TCF cleared when TCF and OCRF match

}

//----- Transmit a character -----//

void transmit (unsigned char a)
{
    int i;
    MON_RAM.TX_CHAR.BYTE = a;

    // start bit
    TX = 0;
    delay(bit_period);

    // 8 data bits
    for (i=0; i<8; i++)
    {
        if (MON_RAM.TX_CHAR.BIT.bit0 == 0)
            TX = 0;

        else
            TX = 1;

        delay(bit_period);
        MON_RAM.TX_CHAR.BYTE = MON_RAM.TX_CHAR.BYTE >> 1;
    }

    // stop bit
    TX = 1;
    delay(bit_period);
}

//----- Transmit characters in Transmit Buffer -----//
void transmit_string (void)
{
    buff_ptr = (char *)&TX_buffer;

    while ( *buff_ptr != 0)
    {
        MON_RAM.TX_CHAR.BYTE = (*buff_ptr++);
        transmit(MON_RAM.TX_CHAR.BYTE);           // call transmit subroutine to
    }
}

//----- Store characters in RX_CHAR -----//
void receive (void)
{
    int j;
    RX_buffer = 0;

    // Receive data bits
    for (j=0; j<8; j++)
    {
        delay(bit_period);
        if (RX == 1)
            MON_RAM.RX_CHAR.BYTE = MON_RAM.RX_CHAR.BYTE | (0x01 << j);
    }
}

```

```

        else
            MON_RAM.RX_CHAR.BYTE = MON_RAM.RX_CHAR.BYTE & rotlc(j,0xFE);
        }

// Receive stop bit
delay(bit_period);
if (RX != 1)
{
    transmit('E'); // error if sampled stop
bit='0', transmit 'Er'
    transmit('r');
}
else
{
    RX_buffer = MON_RAM.RX_CHAR.BYTE ; // save character in receive //buffer
    transmit(RX_buffer); // transmit character in
receive //buffer
}

P_IO.PMR1.BIT.IRQ4 = 1; // P14 used as IRQ4
}

//----- Delay function using Timer F -----//
void delay (unsigned short d)
{
    d = d-49; // decrease d to offset for delay during
setup of timer

    P_TMRF.OCRFB.BYTE.H = d<<8; // save count (d) in output compare
register
    P_TMRF.OCRFB.BYTE.L = d;

    P_TMRF.TCSRFB.BIT.CMFH = 0; // Clear compare match flag
    P_TMRF.TCF.BYTE.H = 0; // Clear counter and start the
timer F
    P_TMRF.TCF.BYTE.L = 0;

    while (P_TMRF.TCSRFB.BIT.CMFH == 0); // Loop until a compare match occurs

    P_TMRF.TCSRFB.BIT.CMFH = 0; // Clear compare match flag
}

```

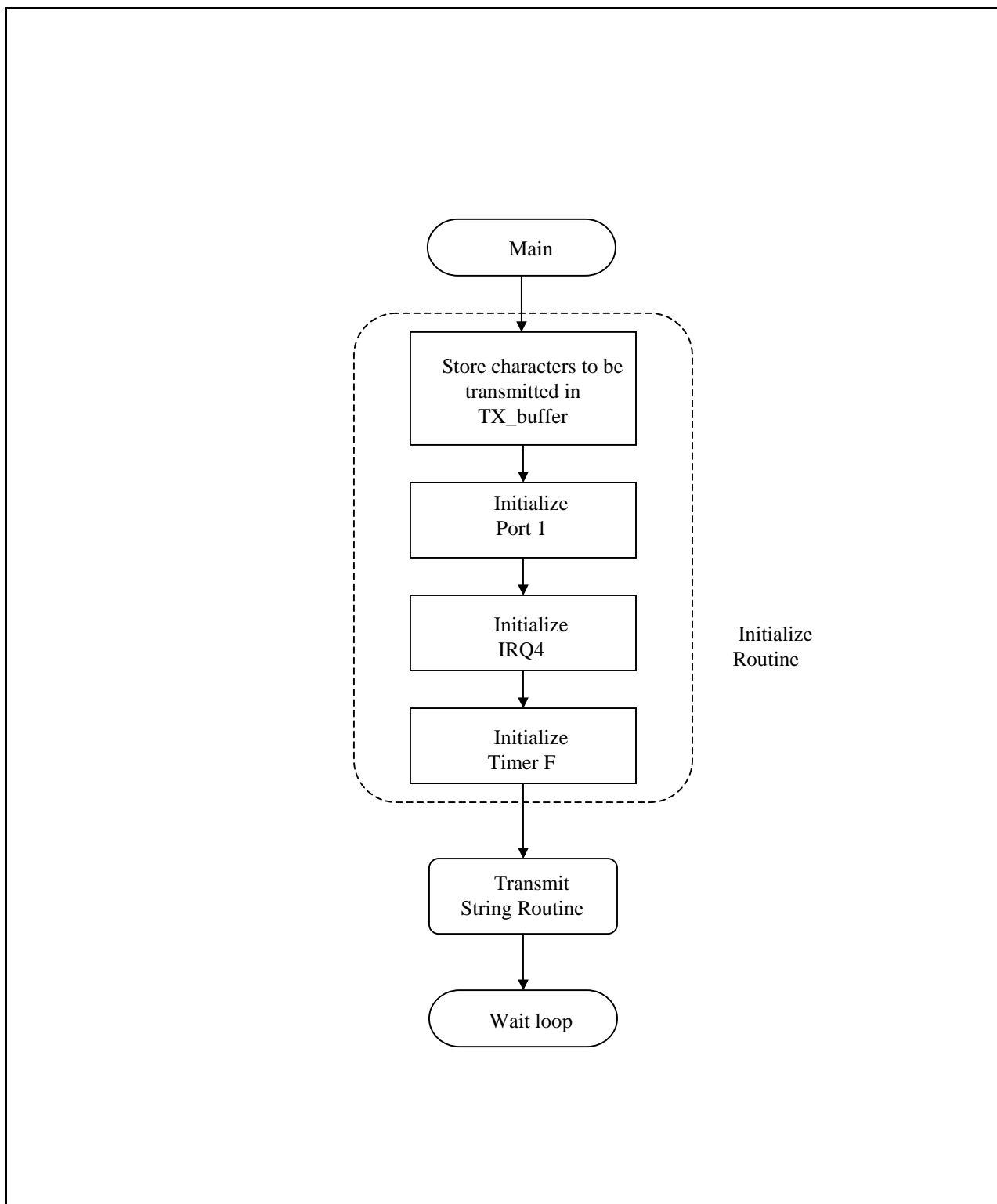


図 8 メインプログラム



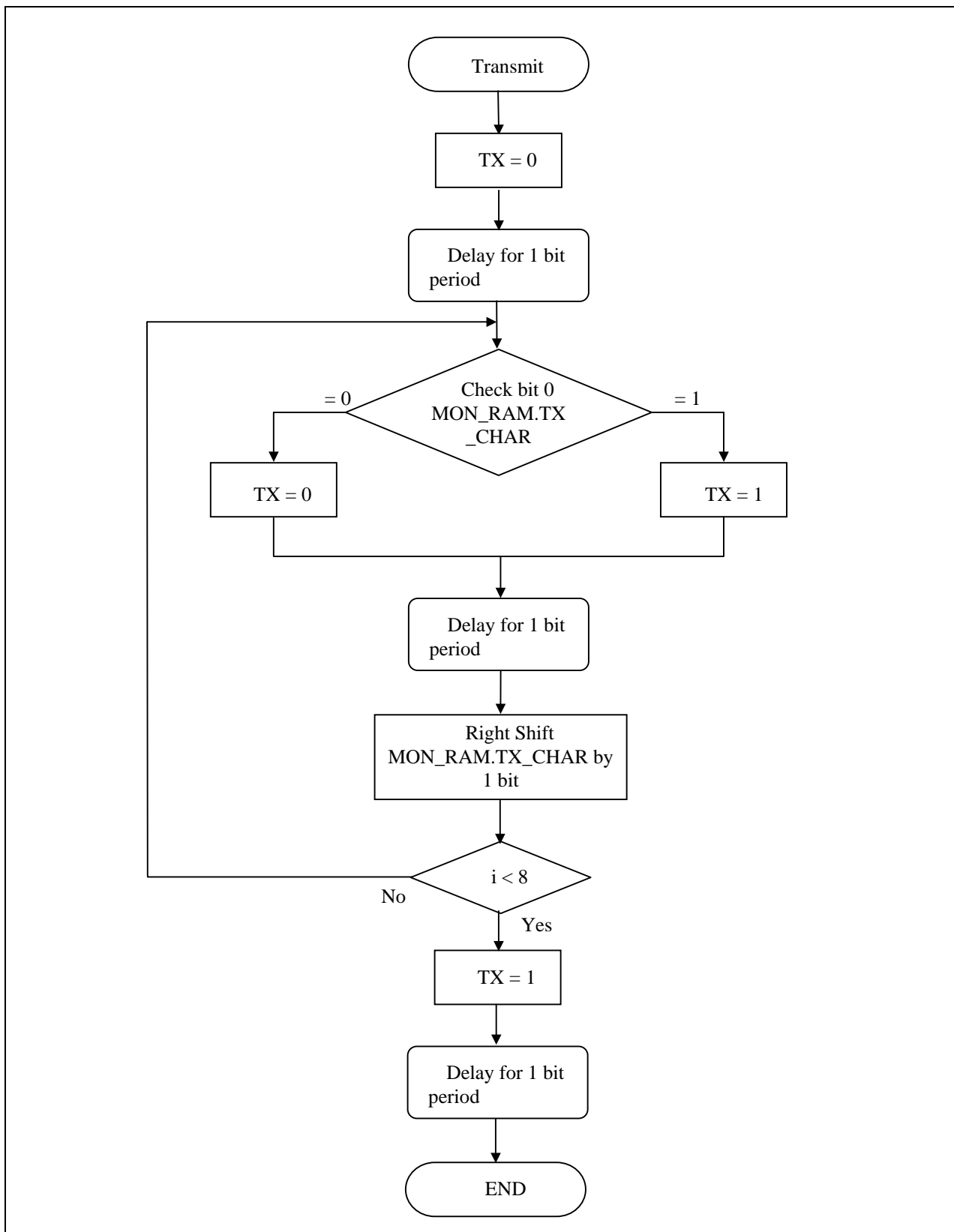


図9 送信ルーチン

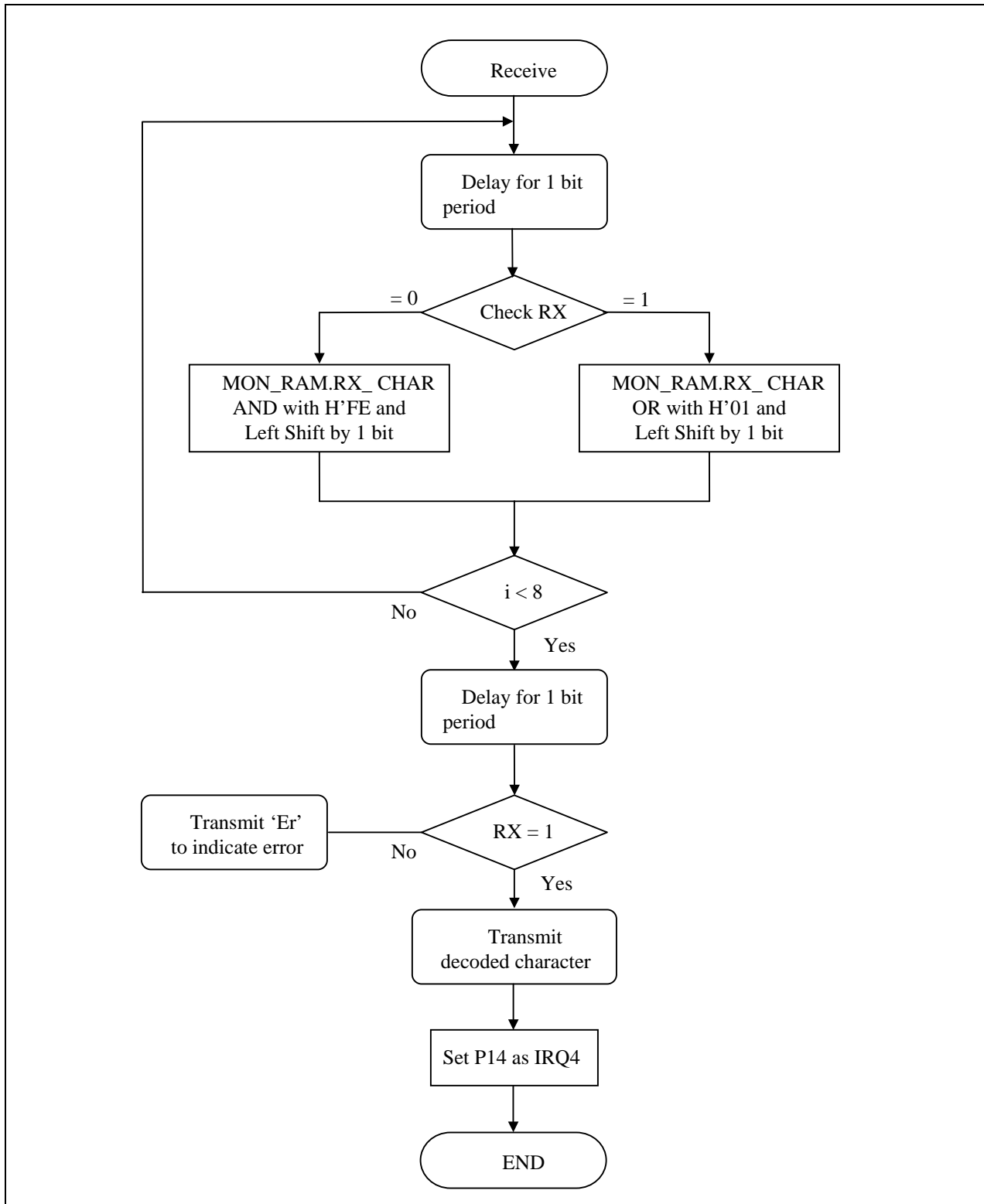


図 10 受信ルーチン

```

/*****/
/* FILE      :flagdefine.h                               */
/* DATE      :Tue, Aug 19, 2003                          */
/* DESCRIPTION:Definition of flag                          */
/* CPU TYPE  :H8/38024                                    */
/* Additional header file                                  */
/*                                                    */
/*****/

#define XTAL      9830400L          // for crystal frequency of 9.83204Mhz

#define BAUD      9600L             // for baud rate of 9600

#define bit_period (XTAL / (8*BAUD)) // for internal clk = 0 /4
// NOTE: 0 = XTAL/2

#define sample    ((bit_period) / 2L)

#define TX        P_IO.PDR1.BIT.P16 // Port 1 pin 6 as transmit pin
#define RX        P_IO.PDR1.BIT.P14 // Port 1 pin 4 as receive pin

/*****/
/*      H8/38024 Flag Definition File                      Ver 1.0      */
/*****/
struct MON                                           /*struct MON_RAM*/
{
    union {
        unsigned char BYTE;                          /* Byte Access */
        struct {
            unsigned char bit7:1;                     /* Bit Access */
            unsigned char bit6:1;
            unsigned char bit5:1;
            unsigned char bit4:1;
            unsigned char bit3:1;                       /*
            unsigned char bit2:1;                       */
            unsigned char bit1:1;                       */
            unsigned char bit0:1;                       */
        } BIT;
    } TX_CHAR;
    union {
        unsigned char BYTE;                          /* Byte Access */
        struct {
            unsigned char bit7:1;                     /* Bit Access */
            unsigned char bit6:1;
            unsigned char bit5:1;
            unsigned char bit4:1;
            unsigned char bit3:1;                       /*
            unsigned char bit2:1;                       */
            unsigned char bit1:1;                       */
            unsigned char bit0:1;                       */
        } BIT;
    } RX_CHAR;
};
#define MON_RAM    (*(volatile struct MON    *)0xFD20) /* MON_RAM Addr */

```

```

/*****
/*
/* FILE      :intprg.c
/* DATE      :Mon, Aug 25, 2003
/* DESCRIPTION :Interrupt Program
/* CPU TYPE  :H8/38024F
/*
/* This file is generated by Renesas Project Generator (Ver.2.1).
/*
/*****

// include 38024F IO define header file
#include "iodefine.h"

#include <machine.h>

// include additional flag define header file
#include "flagdefine.h"

extern void delay (unsigned short);
extern void receive (void);

```

NOTE: Add the following in the IRQ4 vector

```

void INT_IRQ4(void)
{
    set_imask_ccr(1);           // Disable interrupts

    P_IO.PMR1.BIT.IRQ4 = 0;     // P14 used as i/o pin

    P_IO.PCR1.BIT.PCR14 = 0;    // P14 as Input (RX)
    P_IO.PUCR1.BIT.PUCR14 = 1; // P14 MOS pull-up

    // start bit
    delay(sample);             // delay half a bit period to sample at the middle // of
                                // each bit

    if (RX == 0)               // Start receive sequence if sampled start bit equals '0'
        receive();

    P_SYSCR.IRR1.BIT.IRRI4 = 0; // clear interrupt request flag
}

```

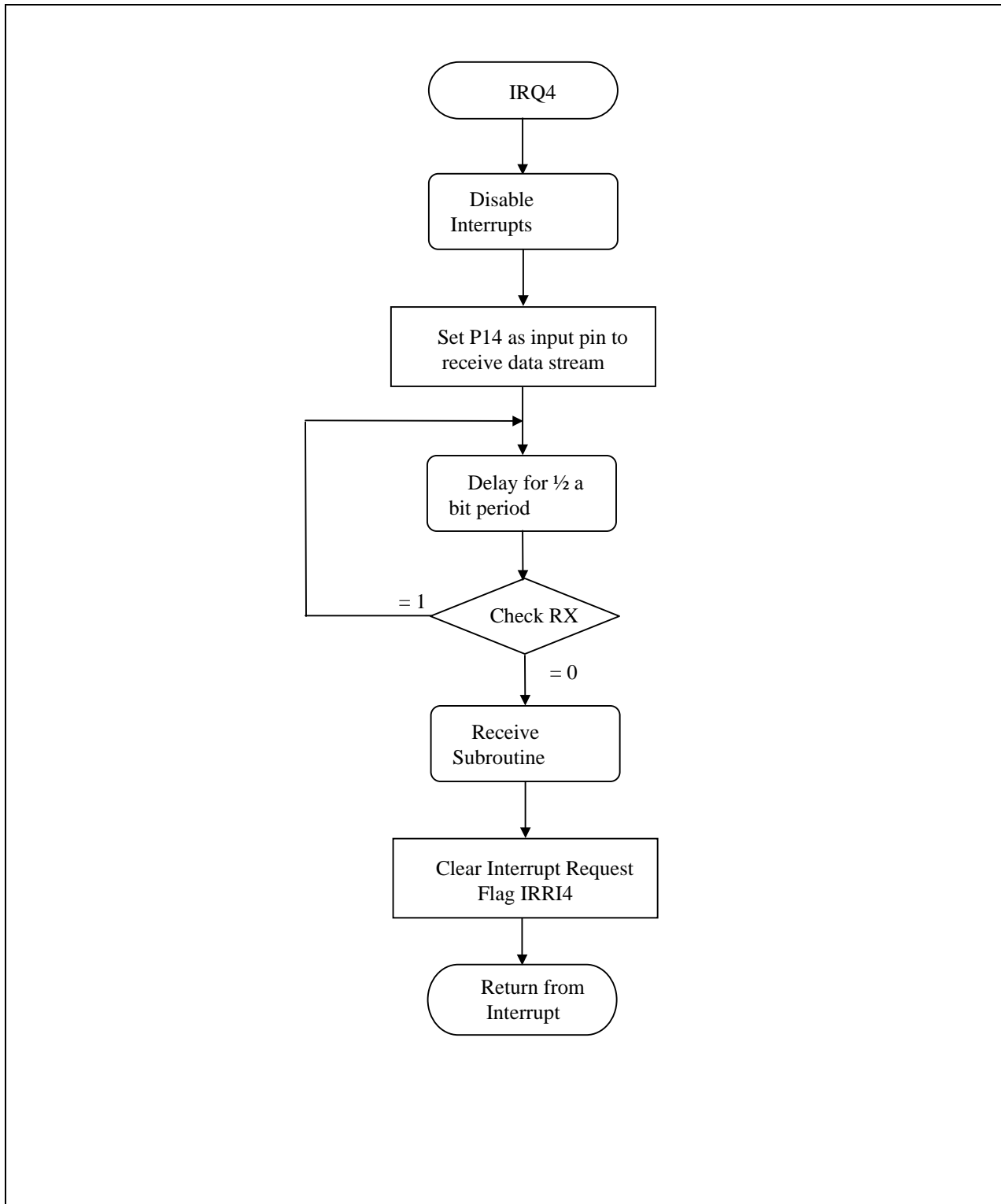


図 11 割り込みサービスルーチン 4(IRQ4)

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2004.08.06	—	初版発行

### 安全設計に関するお願い

1. 弊社は品質、信頼性の向上に努めておりますが、半導体製品は故障が発生したり、誤動作する場合があります。弊社の半導体製品の故障又は誤動作によって結果として、人身事故、火災事故、社会的損害などを生じさせないような安全性を考慮した冗長設計、延焼対策設計、誤動作防止設計などの安全設計に十分ご留意ください。

### 本資料ご利用に際しての留意事項

1. 本資料は、お客様が用途に応じた適切なルネサス テクノロジー製品をご購入いただくための参考資料であり、本資料中に記載の技術情報についてルネサス テクノロジーが所有する知的財産権その他の権利の実施、使用を許諾するものではありません。
2. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例の使用に起因する損害、第三者所有の権利に対する侵害に関し、ルネサス テクノロジーは責任を負いません。
3. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他全ての情報は本資料発行時点のものであり、ルネサス テクノロジーは、予告なしに、本資料に記載した製品または仕様を変更することがあります。ルネサス テクノロジー半導体製品のご購入に当たりましては、事前にルネサス テクノロジー、ルネサス販売または特約店へ最新の情報をご確認頂きますとともに、ルネサス テクノロジーホームページ(<http://www.renesas.com>)などを通じて公開される情報に常にご注意ください。
4. 本資料に記載した情報は、正確を期すため、慎重に制作したものです。万一本資料の記述誤りに起因する損害がお客様に生じた場合には、ルネサス テクノロジーはその責任を負いません。
5. 本資料に記載の製品データ、図、表に示す技術的な内容、プログラム及びアルゴリズムを流用する場合は、技術内容、プログラム、アルゴリズム単位で評価するだけでなく、システム全体で十分に評価し、お客様の責任において適用可否を判断してください。ルネサス テクノロジーは、適用可否に対する責任を負いません。
6. 本資料に記載された製品は、人命にかかわるような状況の下で使用される機器あるいはシステムに用いられることを目的として設計、製造されたものではありません。本資料に記載の製品を運輸、移動体用、医療用、航空宇宙用、原子力制御用、海底中継用機器あるいはシステムなど、特殊用途へのご利用をご検討の際には、ルネサス テクノロジー、ルネサス販売または特約店へご照会ください。
7. 本資料の転載、複製については、文書によるルネサス テクノロジーの事前の承諾が必要です。
8. 本資料に関し詳細についてのお問い合わせ、その他お気付きの点がございましたらルネサス テクノロジー、ルネサス販売または特約店までご照会ください。