7544 Group

Clock Asynchronous Serial I/O (UART)

1. Abstract

The following article introduces and shows an application example of clock asynchronous (UART) of serial I/O.

2. Introduction

The explanation of this issue is applied to the following condition:
Applicable MCU: 7544 Group

3.0 Contents

For clock asynchronous serial I/O (UART), the baud rate and transfer formats used by a transmitter and receiver must be identical.
In the 7544 Group, eight serial data transfer formats can be selected.

3.1 Data Transfer Rate

The transfer bit rate is calculated by the following formula;

● When the internal clock is selected (when baud rate generator is used)

$$\text{Transfer bit rate [bps]} = \frac{f(X_{IN})}{\text{Division ratio}^{*1} \times (\text{BRG setting value}^{*2} + 1) \times 16}$$

Division ratio*1 : "1" or "4" is selected (set by bit 0 of serial I/O control register)
BRG setting value*2 : 0 to 255 (00_{16} to FF_{16}) is set

● When the external clock is selected

Transfer bit rate [bps] = Clock input to S_{CLK} pin/16

Table 1 shows the setting example of baud rate generator and transfer bit rate values.

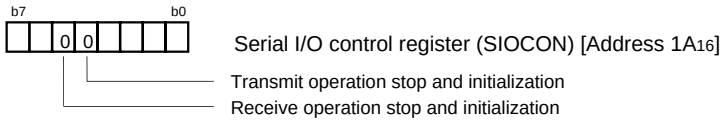
Table 1 Setting example of baud rate generator (BRG) and transfer bit rate values

BRG count source	BRG set value	Transfer bit rate (bps)	
		At $f(X_{IN})$ = 4.9152 MHz	At $f(X_{IN})$ = 8 MHz
$f(X_{IN})$ / 4	255 (FF_{16})	300	488.28125
$f(X_{IN})$ / 4	127 ($7F_{16}$)	600	976.5625
$f(X_{IN})$ / 4	63 ($3F_{16}$)	1200	1953.125
$f(X_{IN})$ / 4	31 ($1F_{16}$)	2400	3906.25
$f(X_{IN})$ / 4	15 ($0F_{16}$)	4800	7812.5
$f(X_{IN})$ / 4	7 (07_{16})	9600	15625
$f(X_{IN})$ / 4	3 (03_{16})	19200	31250
$f(X_{IN})$ / 4	1 (01_{16})	38400	62500
$f(X_{IN})$	3 (03_{16})	76800	125000
$f(X_{IN})$	1 (01_{16})	153600	250000
$f(X_{IN})$	0 (00_{16})	307200	500000

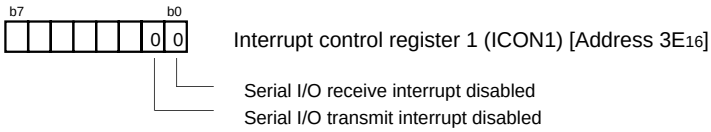
3.2 UART Setting Method

Figure 1 and Figure 2 show the setting method for UART of serial I/O.

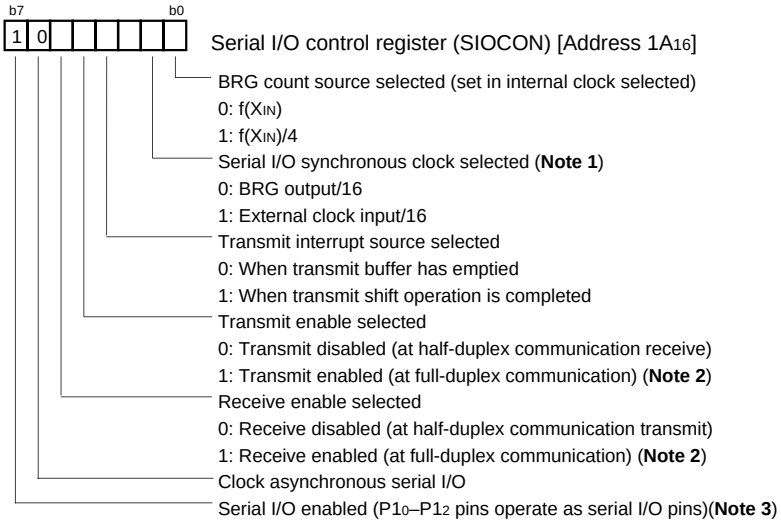
Process 1: Stop and initialize serial I/O.



Process 2: Disable serial I/O transmit/receive interrupt.



Process 3: Set serial I/O control register.



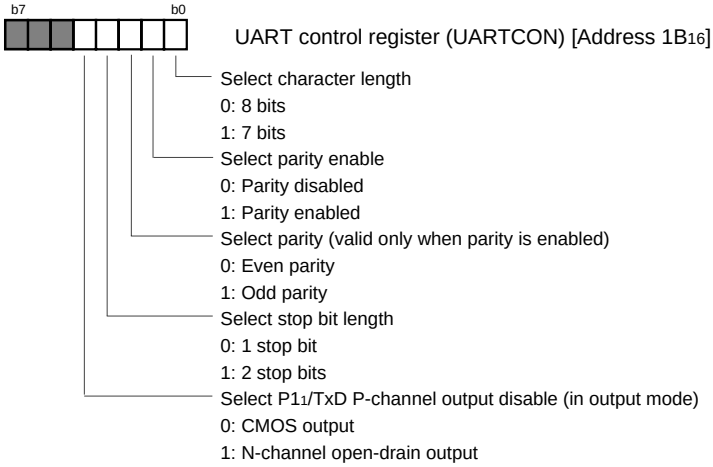
Note 1: Setting of serial I/O synchronous clock selection bit is as follows;
"0": P12 pin can be used as a normal I/O pin
"1": P12 pin is used as an input pin for an external clock.

2: When data transmission is executed at the state that an external clock input is selected as the synchronous clock, set "1" to the transmit enable bit while the SCLK is "H" state.

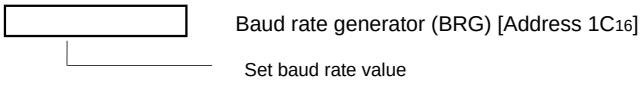
3: When clock asynchronous (UART) serial I/O is selected, P13 pin can be used as a normal I/O pin.

Figure 1 Setting method for UART of serial I/O (1)

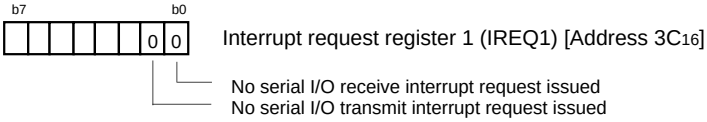
Process 4: Set UART control register.



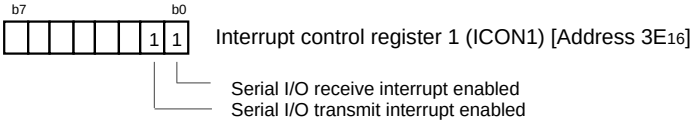
Process 5: When BRG output/16 is selected as synchronous clock, set value to baud rate generator.



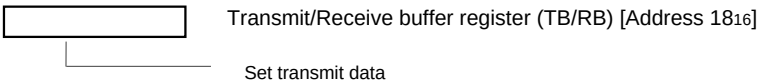
Process 6: In order not to execute the no requested interrupt processing, set “0” (no requested) to the serial I/O transmit/receive interrupt request bit.



Process 7: When the interrupt is used, set “1” (interrupt enabled) to the serial I/O transmit/receive interrupt enable bit.



Process 8: When transmitting, start serial data transmission (**Note**).



Note: When data transmission is executed at the state that an external clock input is selected as the synchronous clock, set the transmit data while the SCLK is “H” state.

Figure 2 Setting method for UART of serial I/O (2)

3.3 Communication Using UART of Serial I/O (Transmit/Receive)

Outline : 2-byte data is transmitted and received, using UART. Port $P0_0$ is used for communication control.

Specifications : •The Serial I/O (UART selected) is used.
•Transfer bit rate : 9600 bps ($f(X_{IN})$ = 4.9152 MHz divided by 512)
•Communication control using port $P0_0$ (output level of port $P0_0$ is controlled by software)
•2-byte data is transferred from the transmitter to the receiver at 10 ms intervals which the timer generates.

Figure 3 shows a connection diagram, Figure 4 shows a timing chart, Figure 5 shows the control procedure of transmitter, and Figure 6 shows an example of control procedure of receiver.

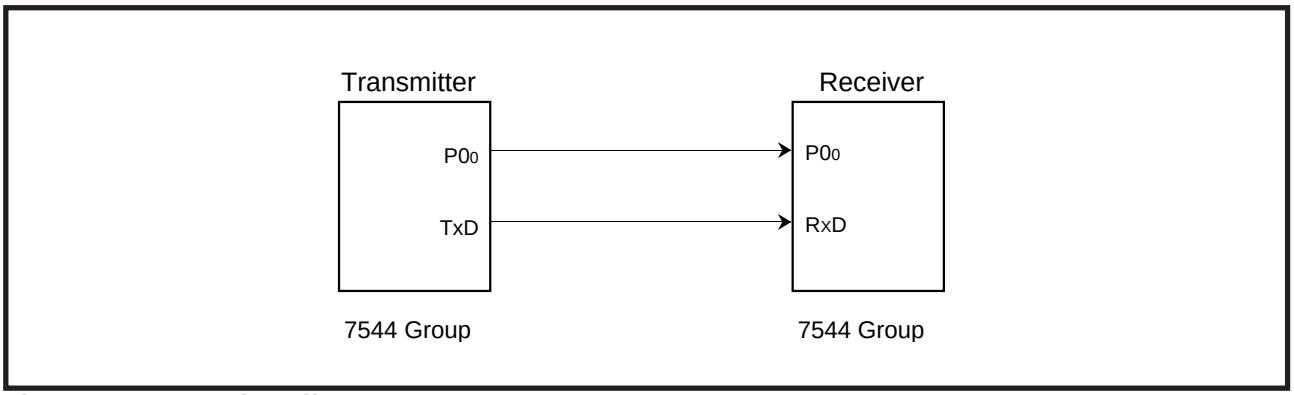


Figure 3 Connection diagram

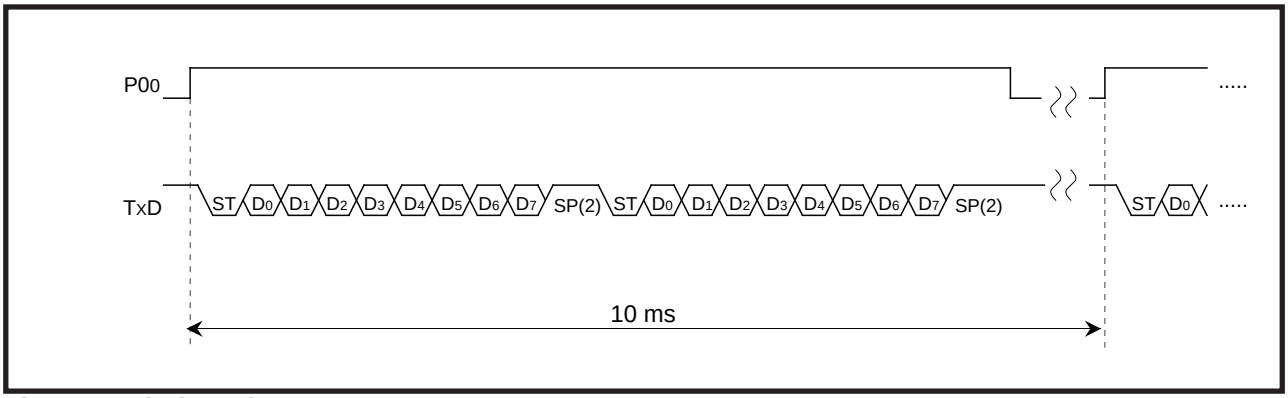


Figure 4 Timing chart

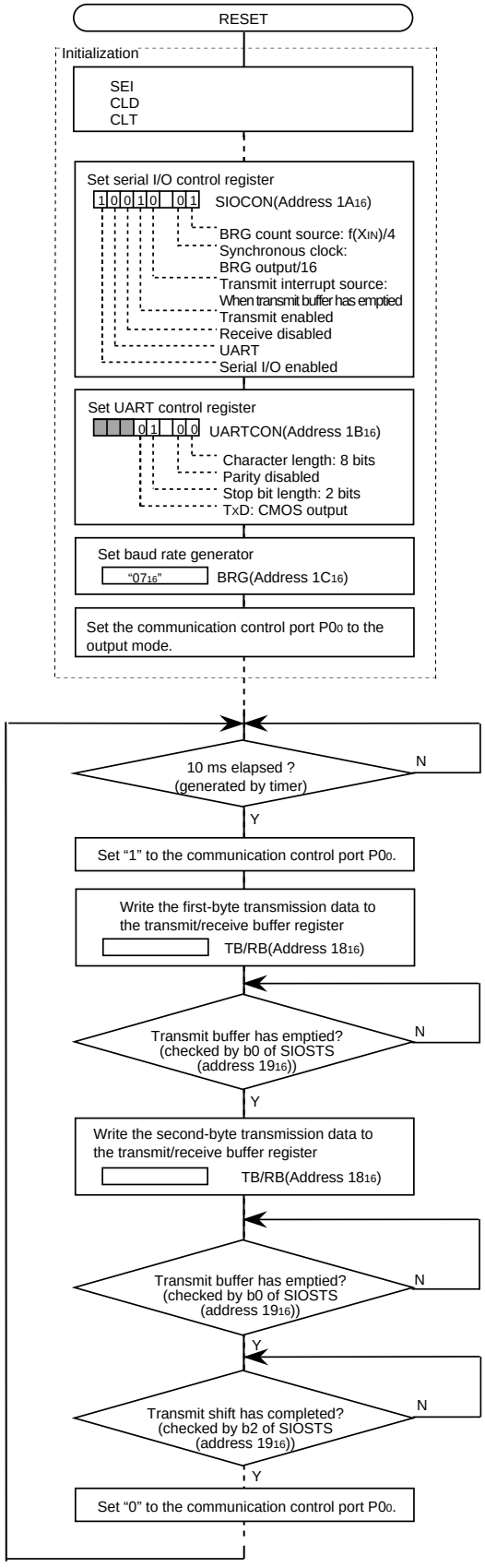


Figure 5 Control procedure of transmitter

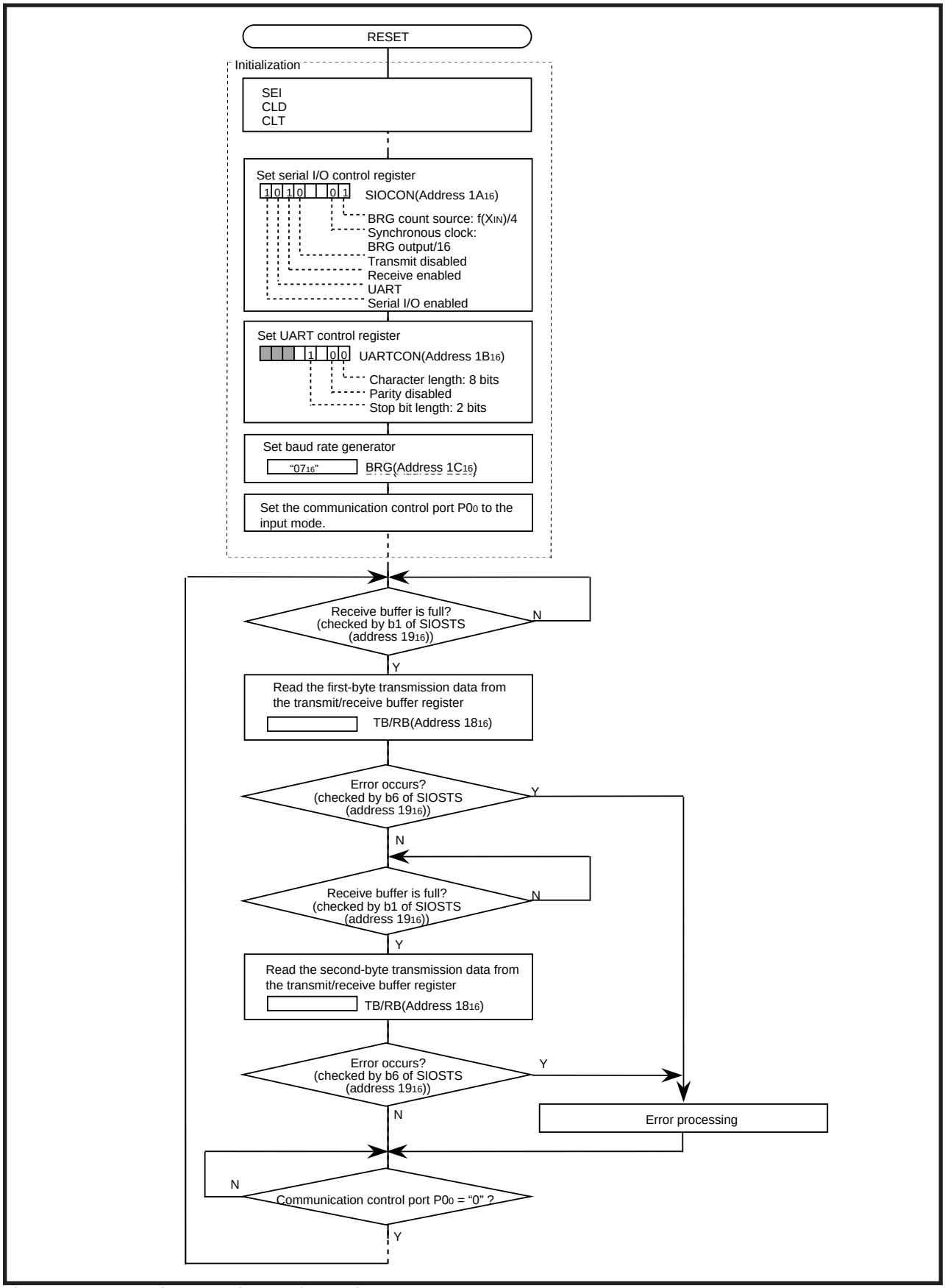


Figure 6 Control procedure of receiver

4. Sample Programming Code

[Reset Start ••• Main Routine Process]

```
RESET:
          SEI                       ; Interrupt disable
          CLD
          CLT
;
          LDX  #$FF                 ; Set stack bottom
          TXS
;
          LDM  #%10000000,CPUM      ; Set CPU mode register
;
; Wait f(XIN) oscillation stabilizing time
;
          LDM  #%00000000,CPUM      ; Set CPU mode register
;
          LDA  #0
          LDX  #>RAM_top
RAM_clear:STA  $00,X
          INX
          BNE  RAM_clear
;
Uart_initial:
          LDM  #%10010001,SIOCON    ; BRG count source : f(Xin)/4
                                    ; synchronous clock : divided 16
                                    ; interrupt request factor : transmit buffer is empty
                                    ; enable transmit
                                    ; disable receive
                                    ; serial I/O mode : asynchronous serial I/O mode
                                    ; enable serial I/O
          LDM  #%00001000,UARTCON   ; character length bit : 8 bits
                                    ; disable parity
                                    ; 2 stop bits
                                    ; TxD : CMOS output
          LDM  #$07,BRG             ; set baud rate

          LDM  #$055,SEND_DATA      ; set send data
          LDM  #$0AA,SEND_DATA+1

          LDM  #%00000000,P0
          LDM  #%00000001,P0D       ; Set Port P0 direction register

          LDM  #%00000001,TCSS2     ; select timer 1 count source : f(Xin)/2
          LDM  #$F9,PRE1            ; transmit cycle 10ms
          LDM  #$61,T1

     CLB 5,IREQ2                    ; clear timer 1 interrupt request
          NOP

;
;-------------------------------------------------------------------------------
__MAIN:
          BBC  5,IREQ2,__MAIN       ; 10ms?
          CLB  5,IREQ2              ; clear timer 1 interrupt request

          SEB  0,P0                 ; transmit start flag

          LDA  SEND_DATA
          STA  TBRB                 ; Send data write

__MAIN00:
          BBC  0,SIOSTS,__MAIN00    ; data send? -> no
;
          LDA  SEND_DATA+1
          STA  TBRB                 ; Next send data write

__MAIN01:
          BBC  0,SIOSTS,__MAIN01    ; data send? -> no
;
__MAIN02:
          BBC  2,SIOSTS,__MAIN02    ; Shift end check ? -> no
;
          CLB  0,P0

          BRA  __MAIN
;
```

Figure 7 Sample Programming Code (Transmit Side)

```
[Reset Start ••• Main Routine Process]

RESET:
          SEI                       ; Interrupt disable
          CLD
          CLT
;
          LDX  #$FF                 ; Set stack bottom
          TXS
;
          LDM  #%10000000,CPUM      ; Set CPU mode register
;
; Wait f(XIN) oscillation stabilizing time
;
          LDM  #%00000000,CPUM      ; Set CPU mode register
;
          LDA  #0
          LDX  #>RAM_top
RAM_clear:    STA  $00,X
          INX
          BNE  RAM_clear
;
Uart_initial:
          LDM  #%10100001,SIOCON    ; BRG count source : f(Xin)/4
                                    ; synchronous clock : divided 16
                                    ; disable transmit
                                    ; enable receive
                                    ; serial I/O mode : asynchronous serial I/O mode
                                    ; enable serial I/O
          LDM  #%00001000,UARTCON   ; character length bit : 8 bits
                                    ; disable parity
                                    ; 2 stop bits
          LDM  #$07,BRG             ; set baud rate

          LDM  #%00000000,P0D       ; Set Port P0 direction register
;
;-------------------------------------------------------------------------------
__MAIN:
          BBC  1,SIOSTS,__MAIN      ; data receive ? -> no
;
          LDA  TBRB
          STA  Receive_Data         ; store received data

          BBS  6,SIOSTS,__ERROR     ; error occur ? -> yes

__MAIN00:
          BBC  1,SIOSTS,__MAIN00    ; data receive ? -> no
;
          LDA  TBRB
          STA  Receive_Data+1       ; store received data

          BBC  6,SIOSTS,__MAIN_01   ; error occur ? -> no
;
__ERROR:
          LDM  #$00,SIOSTS          ; clear all error flag
;
; error processing
;

;
__MAIN_01:
          BBS 0,P0,__MAIN_01

          BRA  __MAIN
;
```

Figure 8 Sample Programming Code (Receive Side)

5. Reference

Data Sheet
7544 Group Data sheet
7544 Group Data sheet (QzROM Version)

Before using this manual, please visit our website to verify that this is the most updated document available.

Renesas Technology Corporation Semiconductor Home Page
http://www.renesas.com/en/740

E-mail Support
E-mail: support_apl@renesas.com

REVISION HISTORY	7544 Group Clock Asynchronous Serial I/O (UART)

Rev.	Date	Description	
		Page	Summary
1.00	Apr 01, 2003	-	First Edition issued
2.00	Nov 12, 2004	8-9	Sample Programming Code added.

Keep safety first in your circuit designs!

1. Renesas Technology Corporation puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage.
   Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of nonflammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

1. These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corporation product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corporation or a third party.
2. Renesas Technology Corporation assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corporation without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corporation or an authorized Renesas Technology Corporation product distributor for the latest product information before purchasing a product listed herein.
   The information described here may contain technical inaccuracies or typographical errors.
   Renesas Technology Corporation assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.
   Please also pay attention to information published by Renesas Technology Corporation by various means, including the Renesas Technology Corporation Semiconductor home page (http://www.renesas.com).
4. When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corporation assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
5. Renesas Technology Corporation semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corporation or an authorized Renesas Technology Corporation product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
6. The prior written approval of Renesas Technology Corporation is necessary to reprint or reproduce in whole or in part these materials.
7. If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.
   Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
8. Please contact Renesas Technology Corporation for further details on these materials or the products contained therein.