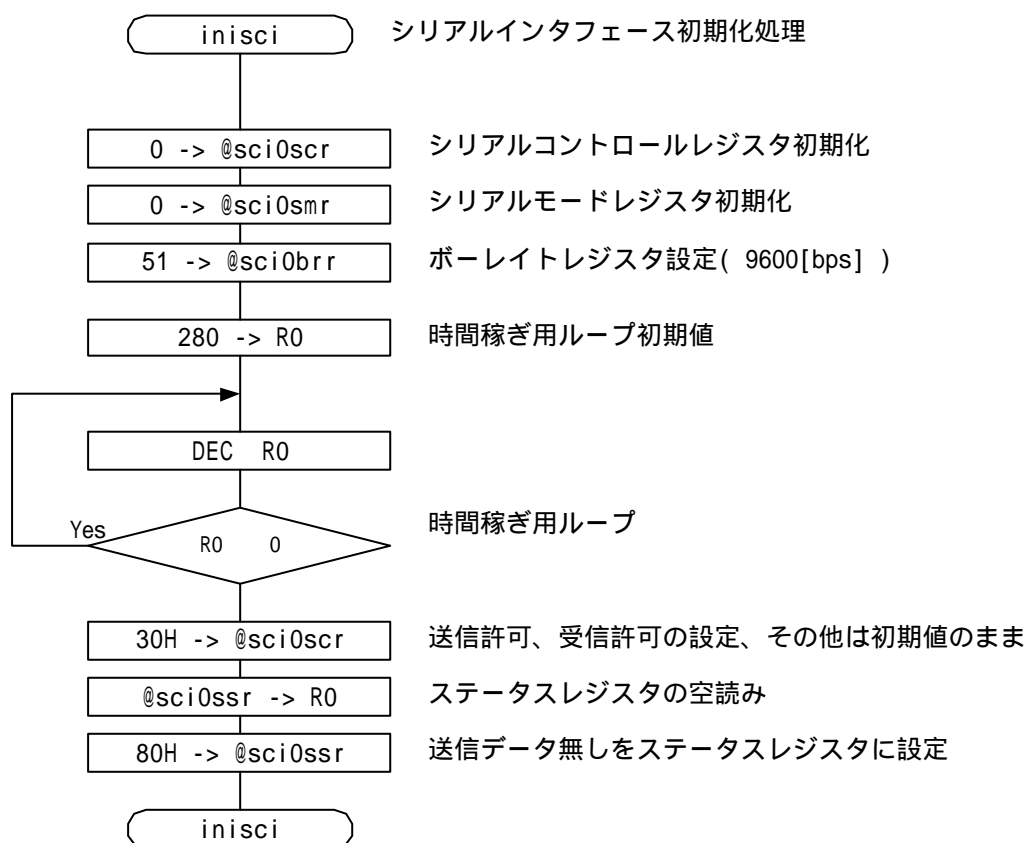
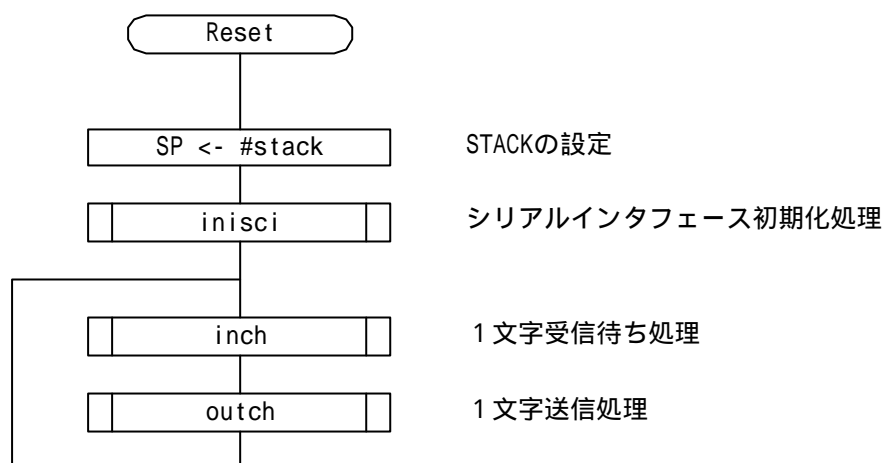
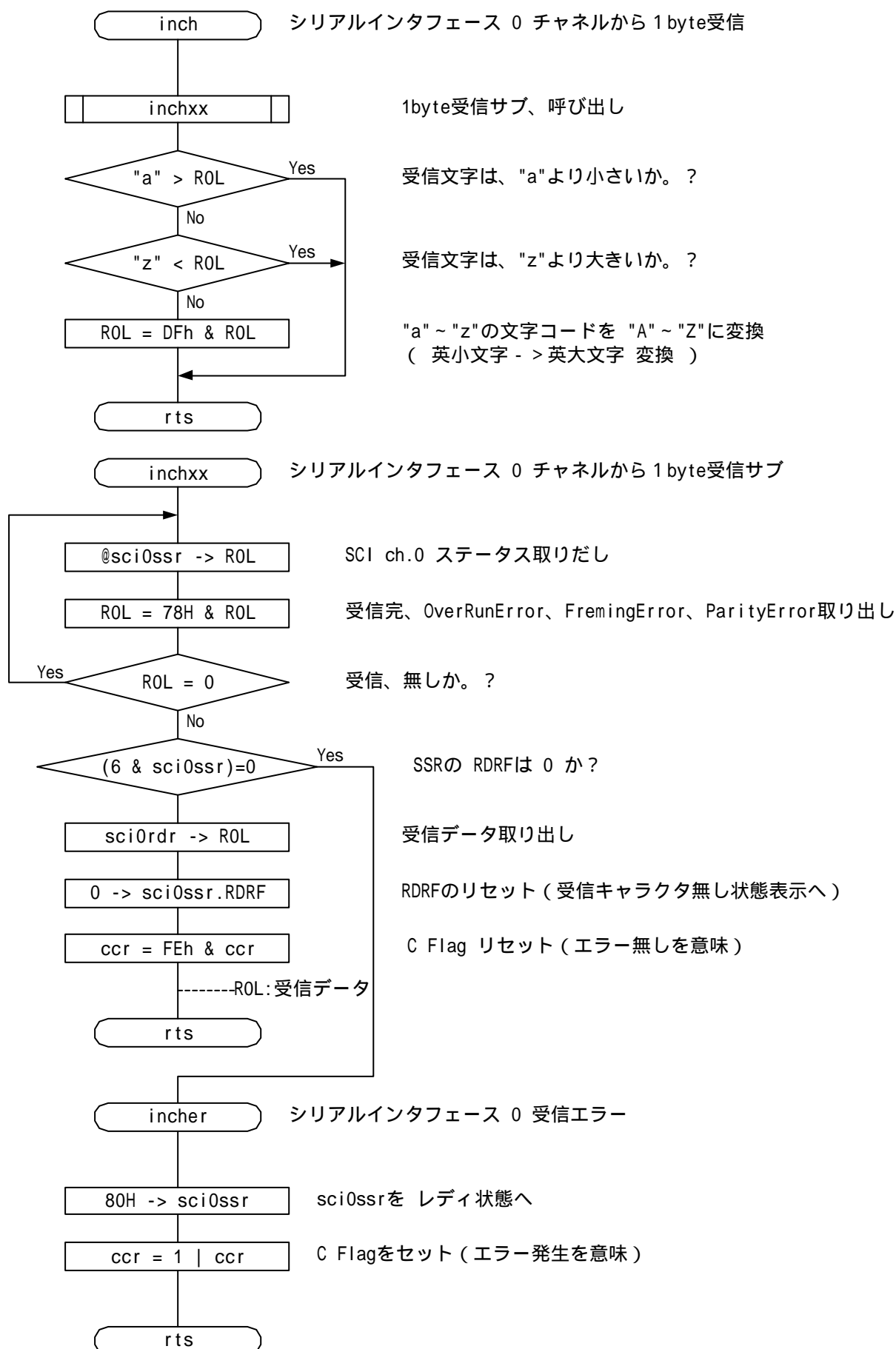


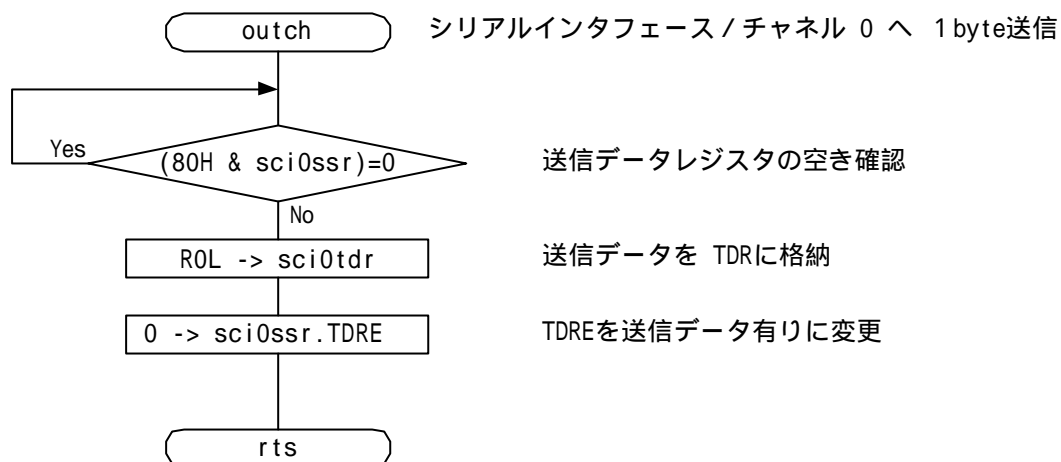
シリアル通信 簡易エコーバックプログラム(1/3)



シリアル通信 簡易エコーバックプログラム(2/3)



シリアル通信 簡易エコーバックプログラム(3/3)



SCI (シリアルインタフェースポート) のアドレス

FFFB0	sci0smr	; モードレジスタ
FFFB1	sci0brr	; ボーレートレジスタ
FFFB2	sci0scr	; コントロールレジスタ
FFFB3	sci0tdr	; 送信データレジスタ
FFFB4	sci0ssr	; ステータスレジスタ
FFFB5	sci0rdr	; 受信データレジスタ

シリアル通信 簡易エコーバックプログラムリスト

```

*****
**                                     **
** AKI-H8 Progrming Test           **
** No.1   ( RS-232C )             **
** -----                       **
** 2002-11-19 By - Take           **
*****

.heading "Test for H8/3048 2002-11-19"
.print  src, nocref, nosct, list

*****
** Simple In/Out for H8/3048 **
** H8/3048 SingleChip Advanced Mode **
** ( 20bit Address )           **
*****

.cpu    300HA:20

*****
** Reset Vector Table           **
*****

.section rom, data, locate=h'0

.data.l reset

; -----
reset:
    mov.l    #stack, sp
    bsr      inisci

loop:
    bsr      inch                ; 1文字入力
    bsr      outch               ; 1文字折り返し送信
    cmp.b    #13, r0l            ; 送信したコードは ( Cr )か。?
    bne      loop                ; 違うなら頭へ戻る
    mov.b    #10, r0l            ; Lf code を レジスタ0 のLowへセット
    bsr      outch               ; 1文字送信
    bra      loop

*****
** SCI 初期化 **
*****
inisci:
    mov.b    #0, r0l
    mov.b    r0l, @sci0scr        ; clear all flags
    mov.b    #0, r0l
    mov.b    r0l, @sci0smr        ; Async, 8bit, NoParity, (Even), stop1, 1/1
    mov.b    #51, r0l
    mov.b    r0l, @sci0brr        ; 9600[bps] ( CPU=16MHz )
    mov.w    #280, r0             ; wait 1 bit time ( 1/9600 sec )

```

```

inisc1:
    dec.w    #1, r0
    bne      inisc1
    mov.b    #h'30, r0l
    mov.b    r0l, @sci0scr          ; scr=0011 0000 ( TE=1, RE=1 )
    mov.b    @sci0ssr, r0l         ; Dummy Read
    mov.b    #h'80, r0l
    mov.b    r0l, @sci0ssr         ; Clear Error Flag (TDRE=1)
    rts

; *****
; **   inch to r0l   **
; **   if err then carry = 1   **
; *****
inch:
    bsr      inchxx                ; to upper
    cmp.b    #"a", r0l
    blo      inch2
    cmp.b    #"z", r0l
    bhi      inch2
    and.b    #h'df, r0l            ; 'a' ~ 'z'を 'A' ~ 'Z'に変換
inch2: rts

; *****
; **   受信文字 取りだし?   **
; *****
inchxx:
    mov.b    @sci0ssr, r0l
    and.b    #h'78, r0l
    beq      inchxx                ; RDRF = ORER = FER = PER = 0
    btst.b   #6, @sci0ssr:8
    beq      incher
    mov.b    @sci0rdr, r0l
    bclr.b   #6, @sci0ssr
    andc.b   #h'fe, ccr
    rts

; *****
; **   受信エラー   **
; *****
incher:
    mov.b    #h'80, r0l
    mov.b    r0l, @sci0ssr
    orc.b    #1, ccr
    rts

; *****
; **   1文字送信   **
; *****
outch:
    btst.b   #7, @sci0ssr
    beq      outch
    mov.b    r0l, @sci0tdr
    bclr.b   #7, @sci0ssr
    rts

```

```
.sectionstk, stack, locate=h'ff400
stack .equ    $

ramend .equ    h'fff0f

; -----
;      initial I/O defsh'fff1c - h'fffff
; -- sci defs

      .sectionio, data, locate=h'fffb0

sci0smr .res.b 1          ; h'fffb0      serial mode register    (h'fffb0)
sci0brr .res.b 1          ; h'fffb1      serial bit rate register
sci0scr .res.b 1          ; h'fffb2      serial control register
sci0tdr .res.b 1          ; h'fffb3      serial TX data
sci0ssr .res.b 1          ; h'fffb4      serial status register
sci0rdr .res.b 1          ; h'fffb5      serial RX data

      .end
```