UART Auto Baud Rate Detection

Interfacing PC system using HMS99C58S

INTRODUCTION

The UART (Universal Asynchronous Receiver/Transmitter) interface features a two wire serial interface bus. This mode enables full-duplex operation wherein one byte of data after the start bit is transmitted or received. The on-chip baud rate generator dedicated to UART enables communications using a wide range of selectable baud rates. The simple interface make it the ideal device for interfacing with the PC system. This application note is intended for design engineers who want to implement serial interface with UART into their application by automatically detecting the baud rate of signal from host PC system and setting the HMS99C58S UART baud rate generator related register.

THE HARDWARD CONNECTION

A typical 2-wire hardware connection is illustrated in Figure 1. The HMS99C58S makes UART interface with host PC system through the MAX232 IC and RS-232 9-pin connector which is connected to serial port of the PC by serial cross-cable. In most systems, the UART is connected to circuitry like MAX232 IC that generates signals complying with the EIA RS232-C specification.

THE SOFTWARE STRUCTURE

An example interface algorithm is listed in Appendix A. A flow diagram is given in Figure 2. The function of example source of HMS99C58S is to initially receive data stream of character `U's from PC followed by automatically detecting baud rate and setting appropriate baud rate generator register and transmitting the received data to PC immediately. The initial detection character uses 'U' of which ascii value is 0x55 and which has signal level change bit by bit and is very convenient to detect consecutive 8 bit data.

Note: This auto baud rate detection algoritm uses 11.0592MHz oscillator and Timer 2 auto baud rate generator. If necessary, the user may use other oscillator, but the calculation algorithm should be changed for exact time capture.

The user should take the following steps when implementing UART serial communication.

A) Byte transmission time capture

1. Wait until the start bit of low level is detected. If detected, wait again until the first data bit of high level is detected.

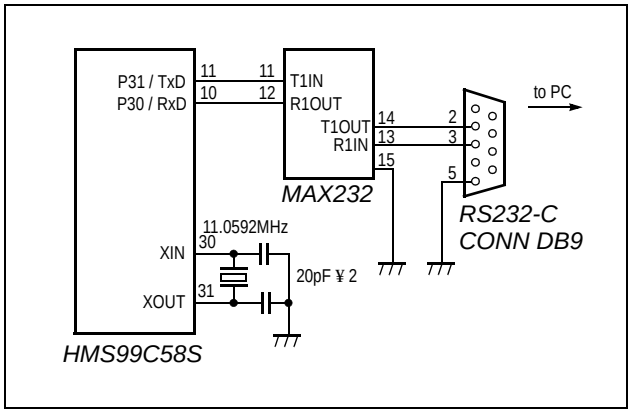

Figure 1. UART Interface Connection

2. Run the Timer 2 so that the time taken to receive one byte of data may be captured. Stop the Timer 2 after the eight bit of data is all detected and stop bit is detected. The exact time taken for eight bits is,

   Timer2 Counter x 12 / Fosc

   which is equal to 8 / baudrate. The baudrate formula is presented in manual like follows.

   Fosc / (32 x [65536 - RCAP2H, RCAP2L])

   As a result, the RCAP2 is,

   65536 - (Timer2 Counter x 3 / 64)

B) Baud rate generator register setting

1. Get the multiplication of Timer 2 capture register value by 3. This is calculated by combination of 16 bit left shift operation and adding original value.
2. Get the division of the above result by 64. This is calculated by 6 times of 16 bit right shift operation.
3. Get the complement value of the above result and then set the Timer 2 baud rate generator register RCAP2H and RCAP2L with the above caculated value.

C) Return message to host PC

1. Send the 'U's stream to host PC.
2. The host PC recognizes the connection operation is successful.

UART DATA FORMAT

Figure 3 shows the format of the transmit/receive data of UART. The one data frame begins with start bit of '0' level.

When 7-bit character mode is selected as the number of character bits, only the lower 7 bits (bits 0 to 6) are valid, so that during a transmission the highest bit (bit 7) is ignored and during reception the highest bit (bit 7) must be set to '0'.

The parity bit is controlled so that there should be an even/odd number of bits whose value is 1 according to the even/odd parity mode.

The one data frame ends with stop bit of 1 or 2 bits.

The transmit operation is started when transmit data is written to the transmit register. A start bit, parity bit, and stop bit(s) are automatically added to the data. Starting the transmit operation shifts out the data, thereby emptying transmit register, after which a transmit completion interrupt request is issued.

The receive operation is started when the UART RxD pin goes low. Once reception of one data frame is completed, the receive data in the shift register is transferred to the receive buffer register and a receive completion interrupt request occurs.

SUMMARY

The serial communication by UART is a simple method of interfacing with PC in the HMS90 Series MCU application. The generic code in Appendix A makes it easy to incorporate in any HMS90 series application. Any of the HMS90 Series MCU with UART can be applied, while at the same time using a designated I/O.

Code size: 111 bytes of program code

Author: *Vince Jeong*
MCU application team
e-mail: byoungsu.jeong@magnachip.com

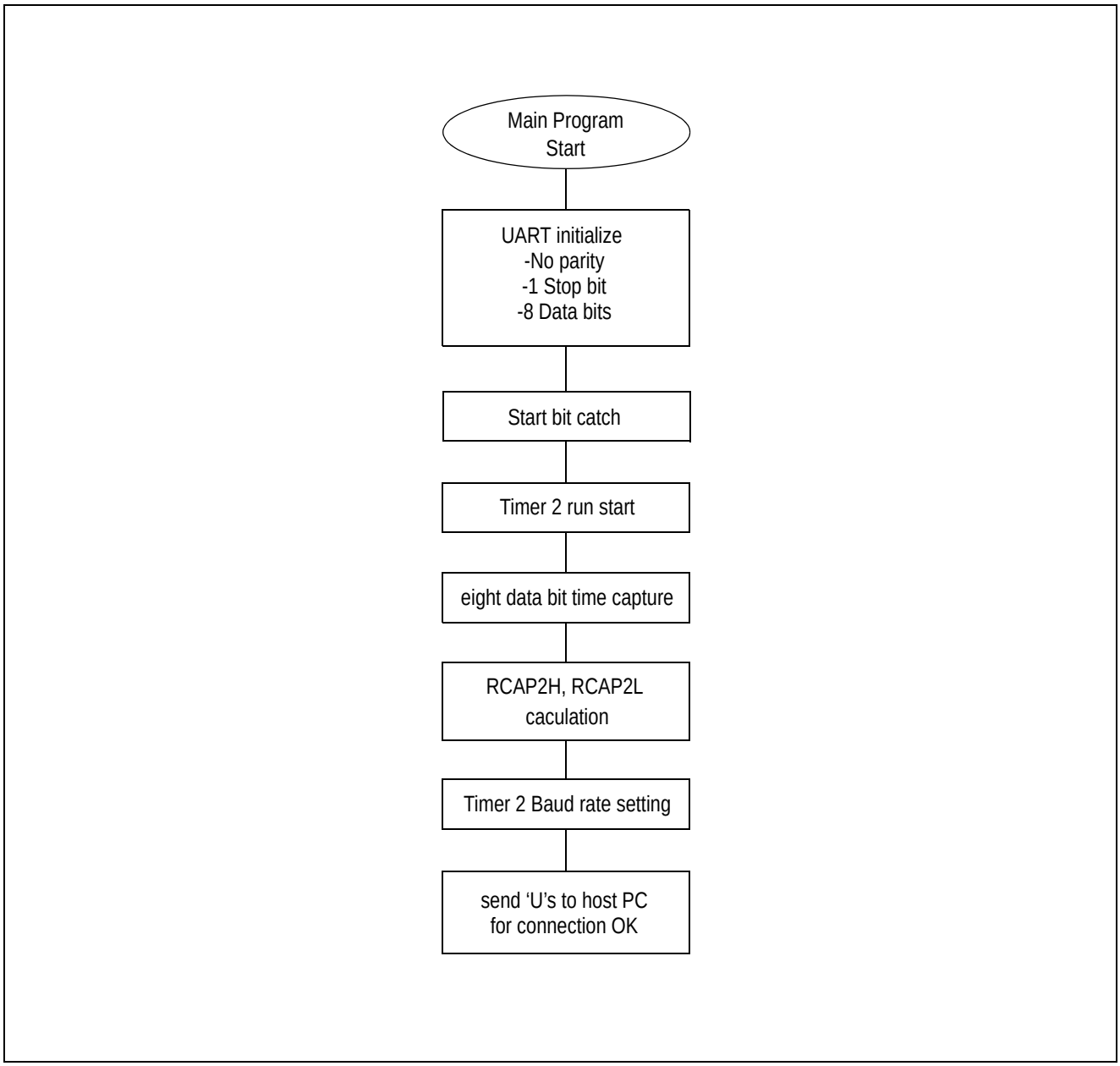

Figure 2. Flow Chart of Example Auto Baud Rate Detection Algorithm

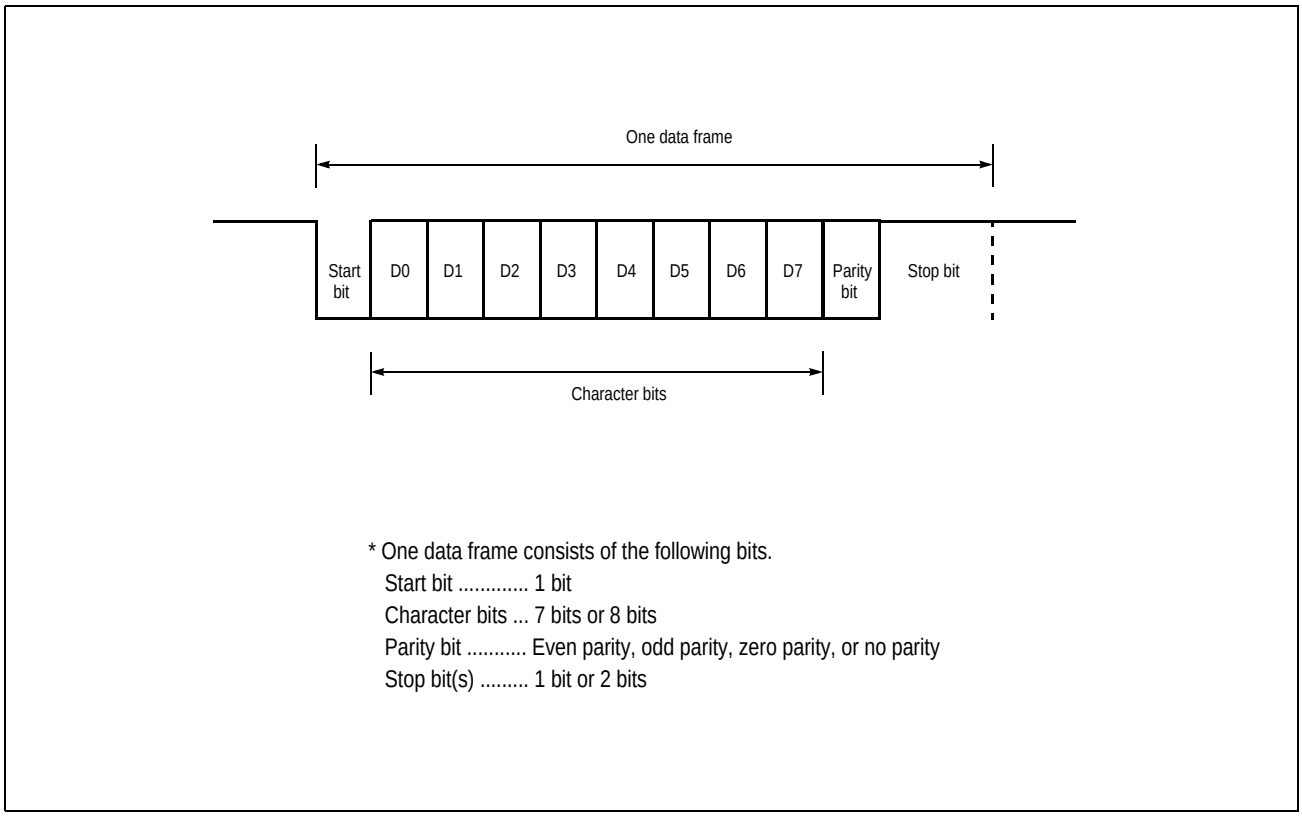

Figure 3. UART Data Format

APPENDIX A

```
$TITLE      (*** Auto baudrate detector ***)
AUTOBAUD_MODULE             SEGMENT     CODE
AUTOBAUD_DATA               SEGMENT     DATA

;_____ D E F I N I T I O N _____________________________________________
IDATA_SIZE          EQU 256                 ; IRAM size

T2CON   DATA    0C8h
RCAP2L  DATA    0CAh
RCAP2H  DATA    0CBh
TL2     DATA    0CCh
TH2     DATA    0CDh
TR2     BIT     0CAh

; *** DATA RAM INTERNAL ***
RSEG        AUTOBAUD_DATA
stack:      DS   40h; this size does not overwrite HEX_DATA segment

;_____ D E C L A R A T I O N _____________________________________________
RSEG        AUTOBAUD_MODULE

;*F*************************************************************************
; NAME:     Boot routine
;*************************************************************************
Boot:
    mov     R0,#IDATA_SIZE-1        ; initialise the IDATA space
idata_loop_:
    mov     @R0,#00h
    djnz    R0,idata_loop_
    mov     SP,#stack-1             ; initialize stack pointer

    acall   autobaud                ; wait 'U'
    setb    TI

return_U:
    mov     A,#'U'
    acall   send_char               ; send 'U'
    sjmp    return_U

;*F*************************************************************************
; NAME:     Send 1 character to host PC through RS232C
;*************************************************************************
send_char:
    jnb     TI,send_char    ; wait Tx is ready
    clr     TI
    mov     SBUF,A          ; send character
    ret

;*F*************************************************************************
; NAME:     Auto baudrate detect
;*************************************************************************
autobaud:                       ; autobaud with timer 2

    MOV     T2CON,#00h
                                ; for (i =0 ; i< 8 ; i++) : 8 data bits
    jb      P3.0,$              ; wait for start bit

    jnb     P3.0,$              ; wait for data bit 0 'H'
    mov     TL2, #00h
    mov     TH2, #00h
    SETB    TR2                 ; start timer
```

```
    MOV     R0,#03h
auto_bit:
    jb      P3.0,$              ; wait for data bit 1/3/5 'L'
    jnb     P3.0,$              ; wait for data bit 2/4/6 'H'
    djnz    R0,auto_bit
    jb      P3.0,$              ; wait for data bit 7 'L'
    jnb     P3.0,$              ; wait for stop bit 'H'
    clr     TR2                 ; stop timer
    dec     TL2                 ; remove one cycle of measurement
    mov     A,TL2
    cjne    A,#0FFh,TH2_00
    dec     TH2                 ; remove 1 to TH2 if initially TL2 was equal 00h
                                        ;(TH2,TL2:0200h--> 01FFh)

TH2_00:
    mov     A,TH2
    mov     R6,A
    mov     A,TL2
    clr     C                   ;first multiply by 2
    rlc     A
    xch     A,R6
    rlc     A
    xch     A,R6            ;T2 * 2
    add     A,TL2               ;add the original value,
    xch     A,R6
    addc    A,TH2           ;ACC=T2 * 3

; a,r6 = (a,r6)/64
    mov     R0,#6
div64:
    clr     C
    rrc     A
    xch     A,R6
    rrc     A
    xch     A,R6
    djnz    R0,div64
;ACC = T2 * 3 / 64
    cpl     A
    mov     RCAP2H,A
    mov     A,R6
    cpl     A
    mov     RCAP2L,A

    mov     SCON,#052h          ; serial port in 8 bits UART,smod=1,
    mov     T2CON,#034h         ; Timer2 in Baud rate Generator mode
    setb    TR2                 ; start Timer2
    ret

END
```