NEC

Serial Interface Channel 2 in Asynchronous Serial Interface (UART) Mode

On-Chip Peripheral Program Example August 1999

Description

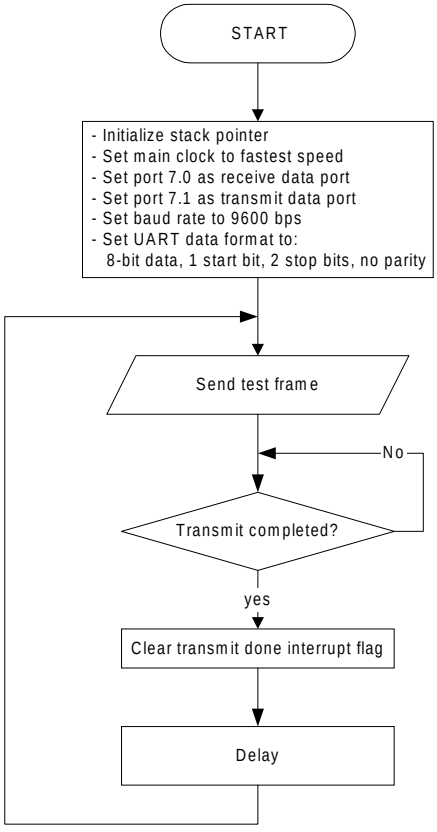
The serial channel 2 of the μPD7805x/78005x subseries can be used in 3-wire serial I/O mode or asynchronous serial interface (UART) mode.

This program demonstrates the asynchronous serial interface (UART) in transfer-only mode. A frame with one start bit, eight data bits, no parity bit and two stop bits is sent continuously to pin TxD/SO2/P71 at 9600 baud. The frame delay is 700 μs for an assembly language program and 2.5 ms for a C language program.

Program Specifications

- Baud rate: 9600
- Data bits: 8 (least significant bit first)
- Start bits: 1
- Stop bits: 2
- Parity: none
- Pin used in program: TxD/SO2/P71 UART data output

Flowchart



Assembly Language Program

```
;*********************************************************************
; Date:       08/16/1999
;
; Parameters: - fastest CPU clock
;               (fx = 5 MHz; 1 CPU clock cycle = 200 ns)
;             - Receive data pin is port 7.0
;             - Transmit data pin is port 7.1
;             - UART mode:
;               9600 baud, 8-bit data, LSB first,
;               1 start bit, 2 stop bits, no parity
;*********************************************************************
;=======================================
;               Constants/Variables       =
;=======================================
RXDirPort       equ     PM7.0     ;RX direction port
TXDirPort       equ     PM7.1     ;TX direction port
TXDataPort      equ     P7.1      ;TX Data port
;=====================================
;       Specify Interrupt vectors       =
;=====================================
Res_Vec CSEG AT 0000h             ; Set main program start vector
        DW      Start
;=====================================
;               Main Program            =
;=====================================
MAIN    CSEG
Start:  DI                        ; Disable interrupts
        MOVW    AX, #0FE20h       ; Load SP address
        MOVW    SP, AX            ; Set Stack Pointer
        MOV     OSMS,#01h         ; Don't use scaler
        MOV     PCC, #00h         ; Main system clock at fastest setting
        SET1    RXDirPort         ; RX set to input direction
        CLR1    TXDirPort         ; TX set to Output direction
        SET1    TXDataPort        ; Latch transmit output to high
        MOV     BRGC,#90h         ; 9600 bps (k = 0, n = 5)
        MOV     ASIM,#0CDh        ; Mode register settings:
                                  ;  - Baud rate generator output
                                  ;  - Two stop bits
                                  ;  - 8-bit data
                                  ;  - No parity
                                  ;  - RX and TX operation enable
        CLR1    STIF              ; Clear transmit done interrupt flag
Loop1:  MOV     A,#55h            ; Load 55h in A
        MOV     TXS,A             ; Transmit the pattern
Loop2:  NOP                       ; Needed NOP for other interrupt acknowledge
        BF      STIF,$Loop2       ; Wait till transmission is complete
        CLR1    STIF              ; Clear transmit done interrupt flag
        MOV     B,#0FFh           ; Loop3 is a delay between the outputs
Loop3:  NOP                       ; 2 clocks
        DBNZ    B,$Loop3          ; Decrement B and branch back, if B is not 0
        BR      Loop1             ; Repeat the output

        END
```

C Language Program

```
/*********************************************************************
; Date:       08/16/1999
;
; Parameters: - fastest CPU clock
;               (fx = 5 MHz; 1 CPU clock cycle = 200ns)
;             - Receive data pin is port 7.0
;             - Transmit data pin is port 7.1
;             - UART mode:
;               9600 baud, 8-bit data, LSB first,
;               1 start bit, 2 stop bits, no parity
;*********************************************************************/
/* extension functions in K0/K0S compiler */
#pragma sfr     /* key word to allow SFR names in C code */
#pragma asm     /* key word to allow ASM statements in C code */
#pragma NOP     /* key word for NOP instruction in C code */
#pragma DI      /* key word for DI instruction in C code */
#pragma EI      /* key word for EI instruction in C code */
/*=====================================
;       Constants/Variables             =
;=====================================*/
#define TRUE    1
#define FALSE   0
#define RXDirPort       PM7.0   // RX direction port
#define TXDirPort       PM7.1   // TX direction port
#define TXDataPort      P7.1    // TX Data port

unsigned int  j;
/*=====================================
;       Main Program                    =
;=====================================*/
void main(void)
{
        DI();                           /* Disable interrupts */
        OSMS = 0x01;                    /* Don't use scaler */
        PCC = 0x00;                     /* Main system clock at fastest setting */
        RXDirPort =1;                   /* RX set to input direction */
        TXDirPort =0;                   /* TX set to Output direction */
        TXDataPort=1;                   /* Latch transmit output to high */
        BRGC = 0x90;                    /* 9600 bps (k = 0, n = 5) */
        ASIM = 0xCD;                    /* Mode register settings:
                                                - Baud rate generator output
                                                - Two stop bits
                                                - 8-bit data
                                                - No parity
                                                - RX and TX operation enable
        STIF  = 0;                      /* Clear transmit done interrupt flag */
        while(TRUE)                     /* test loop */
        {
             TXS = 0x55;                /* Transmit the pattern */
             while( !STIF);             /* Wait till transmission is done */
             STIF = 0;                  /* Clear transmit done interrupt flag */
             for(j=0;j < 226 ; j++);    /* Delay 1 msec  */
        }                               /* End of WHILE loop */
}                                       /* end of MAIN() */
```

NEC

For literature, call **1-800-366-9782** 7 a.m. to 6 p.m. Pacific time
or FAX your request to **1-800-729-9288**
or visit our web site at **www.necel.com**

50881