4-Bit Micro-Controller

TM87 series

How to implement the UART transmission on TM8725, TM8726 and DEMO BOARD?

Application Note

Tenx reserves the right to change or discontinue this product without notice.

tenx technology inc.

CONTENTS

PRODUCT NAME

TM87 series

TITLE

How to implement the UART transmission on TM8725, TM8726 and DEMO BOARD?

APPLICATION NOTE

19200bps transmission rate is used as a demonstration in this example. There are two attached files in this example: UART.asm and UART.opt. These files can be used directly on TM87 ICE.
Because the accuracy of the transmission frequency is critical to the UART transmission, a 3.6864MHz Crystal oscillator is required to serve as the system frequency of TM87. Because all the TM87 instructions have to be executed within 4 clock cycles, it can be calculated that the amount of time needed to transmit one bit is 48 instruction cycles when the required baud rate is 19200：
3686400 / 19200 / 4 = 48

Notes：

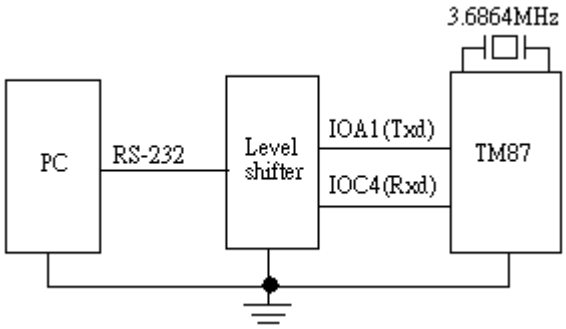
1. TM87 must work under a voltage source of more than 3V so that it can execute at a system frequency of 3.6864MHz.
2. TM87 must enter fast mode before the program to start transmission so that instructions will be executed at a system frequency of 3.6864MHz.

These results can be further used to control the DELAY that is needed for output or input, or the required Baud Rate.
Because the signal levels for RS-232 are: are 0: +3V ~ +15V and 1: -3V ~ -15V, there is a need to use a level shifter IC to convert the voltage level between TM87 and RS-232.

Description of the sample program functions：

This program can receive data values transmitted from a PC and resend back to the PC immediately. However, it can not receive data while transmitting. Therefore, the transmission interval will be at least the time spent for one transmission. Assuming the PC transmits 0123456789 continuously, the resending-back transmission can only be 02468.

Reference example：



BaudRate : 19200 bps
Parity : NO
Data Bits : 8 BIT
Stop Bit : 1 BIT

```
.chip tm8705
.code                   ;FAST CLOCK :3.6864MHz
      org 000h          ; Baud Rate: 19200
      jmp START         ; 0    X X X X X X X X   1
      ORG 014H          ; START Lo ~ Hi(8 Bits) STOP
      SIE* 00H          ;       73H ~ 74H
      JMP IOC           ;3686400 / 19200 / 4 =  48 Instruction cycles
                        ;48Instruction cycles = PH3 * 24  = PH3 * 22 + 4 Instruction cycles
                        ;                     = PH3 *14   + PH6 + 4 Instruction cycles

START:
      spa 0fh           ;IOA transmit DATA
      spc 10h           ;IOC receive DATA
      lds 70h,0fh       ;Initialize IOA
      opa 70h
      plc 10h
      tm2x 010000000b
      she 10h
      halt
      lds 74h,00h
      lds 73h,00h
      lds 70h,00h
      lds 71h,00h
next:   PLC 01H
      scc 14h           ;Cch = PH6
      sca 10h           ;Generate an interrupt as soon as IOC receives a START BIT
      SIE* 01H          ;
      plc 04h
      she 04h
      halt
      sie* 00h
      mrw 70h,73h
      mrw 71h,74h
SEND:
      lds 72h,00h       ;Send START BIT
      opa 72h           ;
      plc 10h           ;DELAY 1/19200 Sec
      tm2x 001010101b
      she 10h
      halt
      nop

      nop               ;Send BIT0
```

```
opa 70h             ;
plc 10h             ;DELAY 1/19200 Sec
tm2x 001010101b
she 10h
halt
nop

sr0 70h             ;Send BIT1
opa 70h             ;
plc 10h             ;DELAY 1/19200 Sec
tm2x 001010101b
she 10h
halt
nop

sr0 70h             ;Send BIT2
opa 70h             ;
plc 10h             ;DELAY 1/19200 Sec
tm2x 001010101b
she 10h
halt
nop

sr0 70h             ;Send BIT3
opa 70h             ;
plc 10h             ;DELAY 1/19200 Sec
tm2x 001010101b
she 10h
halt
nop

nop                 ;Send BIT4
opa 71h             ;
plc 10h             ;DELAY 1/19200 Sec
tm2x 001010101b
she 10h
halt
nop

sr0 71h             ;Send BIT5
opa 71h             ;
plc 10h             ;DELAY 1/19200 Sec
tm2x 001010101b
she 10h
halt
nop

sr0 71h             ;Send BIT6
```

```
        opa 71h             ;
        plc 10h             ;DELAY 1/19200 Sec
        tm2x 001010101b
        she 10h
        halt
        nop

        sr0 71h             ;Send BIT7
        opa 71h             ;
        plc 10h             ;DELAY 1/19200 Sec
        tm2x 001010101b
        she 10h
        halt

        nop                 ;Send STOP BIT
        lds 72h,0fh         ;
        opa 72h             ;DELAY 1/19200 Sec
        plc 10h
        tm2x 001010001b
        she 10h
        halt

        JMP next

IOC:
        sca 00h             ; Cch = PH6
        plc 10h             ; Interrupt vector, 2 instruction cycles (ORG 014H)
        tm2x 001001101b     ; PH3 * 14
        she 10h             ; TOTAL DELAY =  PH6 + PH3 *14 + 4 Instruction cycles
        halt                ;                = 1/19200 Sec (START BIT)

        ipc 00h              ;GET BIT0
        plc 10h              ;DELAY 1/19200 Sec
        tm2x 001010101b
        she 10h
        halt
        nop

        nop
        ipc 01h              ;GET BIT1
        plc 10h             ;DELAY 1/19200 Sec
        tm2x 001010101b
        she 10h
        halt
        nop

        nop
        ipc 02h             ;GET BIT2
```

```
plc 10h             ;DELAY 1/19200 Sec
tm2x 001010101b
she 10h
halt
nop

nop
ipc 03h             ;GET BIT3
plc 10h             ;DELAY 1/19200 Sec
tm2x 001010101b
she 10h
halt
nop

nop
ipc 04h             ;GET BIT4
plc 10h             ;DELAY 1/19200 Sec
tm2x 001010101b
she 10h
halt
nop

nop
ipc 05h             ;GET BIT5
plc 10h             ;DELAY 1/19200 Sec
tm2x 001010101b
she 10h
halt
nop

nop
ipc 06h             ;GET BIT6
plc 10h             ;DELAY 1/19200 Sec
tm2x 001010101b
she 10h
halt
nop

nop
ipc 07h             ;GET BIT7
plc 10h             ;DELAY 1/19200 Sec
tm2x 001010101b
she 10h
halt
nop

lds 08h,00h
ipc 08h             ;GET STOP BIT
```

```
        jz rio              ;Determine if STOP BIT is correct?
        sr0 00h             ;Use the time for STOP BIT to do data sorting
        sr0 00h
        sr0 00h
        sr0 01h
        sr0 01h
        sr0 02h
        sr0 04h
        sr0 04h
        sr0 04h
        sr0 05h
        sr0 05h
        sr0 06h
        lds 08h,00h
        or 00h
        or 01h
        or 02h
        or 03h
        or* 08h
        lds 09h,00h
        or 04h
        or 05h
        or 06h
        or 07h
        or* 09h
        lds 00h,00h
        lds 01h,00h
        lds 02h,00h
        lds 03h,00h
        lds 04h,00h
        lds 05h,00h
        lds 06h,00h
        lds 07h,00h          ;35
        mrw 73h,08h          ;LO
        mrw 74h,09h          ;HI
        plc 10h
        tm2x 001000001b
        she 10h
        halt
rio:    rts
.endc
```