

HT48 & HT46 MCU UART 的软件实现方法

文件编码：HA0004s

介绍

具有低功耗、高性能的 HOLTEK 的 8 位单片机，十分适用于各种控制系统产品，如办公自动化和一些消费性产品中。

有时在一些应用系统中需要附加与其它单片机进行异步串行通信，在这种情况下，使用软件方法实现比选择专用的 IC 硬件方法要经济得多。本文就是介绍简单的 UART 软件实现方法。该方法适用于带一位停止位并且不带奇偶校验位的通用 8 位数据传输，其传输波特率可调整，调整范围由使用的单片机的系统频率决定。

功能实现

在本文中，测试时将使用到一个包含文件 dfscript.inc 和一个源文件 dfscript.asm。

测试时请按下述步骤做：

1. 根据实际电路，在 dfscript.inc 文件中定义 TXPIN 和 RXPIN 引脚，这两个引脚分别作为发送和接收引脚。
2. 按照下公式定义传输波特率和使用的系统频率以及计算机的波特率常数：

$$\text{波特率常数} = (\text{系统频率} / (\text{波特率} \times 12)) - 3$$
 或者，也可以根据下表查到波特率常数：

系统频率 \ 波特率	4MHz	2MHz	1MHz
9600	31	14	X
7200	43	20	8
4800	66	31	14
3600	89	43	20
2400	135	66	31
2000	163	80	38
1800	182	89	43
1200	X	135	66

X：不适用

确定波特率常数后，按其修改在 dfscript.inc 里定义的值。

3. 将包含文件 dfscript.inc, dfscript.asm 加到实现发送和接收功能的源文件中。
4. 两个功能的实现需要以下支持：
 - a. 发送
要发送的 8 位数据必须在调用之前放在仿真器中。
 - b. 接收
接收的 8 位数据必须放在仿真器中。

例子

假设需要下列条件：

1. 系统频率： 4MHz
2. 波特率： 4800
3. 使用 PA3 口作为发送
4. 使用 PA2 口作为接收

```

;-----
; 文件名: DFSCRIPT.INC
; 作者: Jason Tseng
;-----

BaudRateConst EQU 66           ; 可从表格中查到
TXPIN EQU PA.3
RXPIN EQU PA.2

ifndef DFSCRIPT_ASM
EXTERN TRANSMIT:NEAR
EXTERN RECEIVE:NEAR
endif

;-----
; End of DFSCRIPT.INC
;-----

;-----
; 文件名: TEST.ASM
; 作者: Jason Tseng
;-----

include dfscript.asm
MyData .SECTION 'DATA'
:
MyCode .SECTION 'CODE'
ORG 0H
JMP START
START:
:
SET PAC.2           ;设置接收引脚为输入口
CLR PAC.3           ;设置发送引脚为输出口
:
MOV A , TRANSMIT_DATA ;将要发送的数据赋给累加器ACC
CALL TRANSMIT       ;调用发送子程序
:
:
CALL RECEIVE        ;调用接收子程序
MOV RECEIVE_DATA , A ;从累加器ACC接收数据

```

```

:
END
;-----
; End of TEST.ASM
;-----

```

占用资源

```

RAM 空间   :   4
ROM 空间   :  49
I/O  引脚   :   2

```

实现

在本文中提供两个文件：dfscript.inc（详见附件 1）和 dfscript.asm（详见附件 2）。在这个应用实现中，我们假定为传输 8 位数据、一个停止位、不带奇偶校验的通用 UART。基本地，我们是利用消耗指令周期来获得期望的波特率，于是，在仿真中会存在一点误差。总体上来说，系统频率和波特率越高，误差则越小。根据我们经验，波特率常数最佳值应设在 7~256 之间。

附件 1：

```

;-----
; 文件名: DFSERIPT.INC
; 作者: Jason Tseng
; 作用: 设置适当的波特率常数, TXPIN和RXPIN
include ht48r30a-1.inc
BaudRateConst EQU 66
TXPIN EQU PA.3
RXPIN EQU PA.2
ifndef DFSERIPT_ASM
EXTERN TRANSMIT:NEAR
EXTERN RECEIVE:NEAR
endif
;-----
; End of DFSERIPT.INC
;-----

```

附件 2：

```

;-----
; 文件名: DFSERIPT.ASM
; 作者: Jason Tseng
;-----
#define DFSERIPT_ASM
#include DFSERIPT.INC
PUBLIC TRANSMIT
PUBLIC RECEIVE

```

```

BAUDRATE EQU BAUDRATECONST           ; 波特率为波特率常数(在dfscript.inc中由用户定义)
TX EQU TXPIN                          ; 用户定义TXPIN引脚指定为TX
RX EQU RXPIN                          ; 用户定义RXPIN引脚指定为RX

SDATA .SECTION 'DATA'

    COUNT DB ?                        ; 串行位计数器
    TXREG DB ?                        ; 发送数据寄存器
    RCREG DB ?                        ; 接收数据寄存器
    DELAY DB ?                        ; 延时计数器

SERIAL .SECTION 'CODE'

TRANSMIT PROC                          ; TRANSMIT子程序
MOV TXREG,A                            ; 将发送的数据赋给TXREG
MOV A,BAUDRATE                         ; 设置波特率延时
MOV DELAY,A                            ;
CLR TX                                 ; 发送起始标志位“0”
MOV A,9                                ; 设置发送数据的位数
MOV COUNT,A                            ;
TXDELAY:                               ;
SDZ DELAY                              ; 波特率延时循环
JMP TXDELAY                             ;
MOV A,BAUDRATE                         ; 重新载入波特率延时常数
MOV DELAY,A                            ;
SDZ COUNT                              ; 检查数据发送是否完毕
JMP SENDBIT                            ; 如果未发送完毕，跳到SENBIT发送下一位
JMP ENDTX                              ; 数据发送完毕，跳到ENDTX

SENBIT:
RRC TXREG                              ; 带借位C循环右移txreg以获得发送的数据
SNZ C                                  ; 检查借位C
JMP LOBIT                              ; C=0跳到LOBIT发送“0”
SET TX                                  ; C=1发送“1”
JMP TXDELAY                             ;
LOBIT:
CLR TX                                  ; 发送“0”
JMP TXDELAY                             ;
ENDTX:
NOP                                     ; 延时
NOP                                     ;
SET TX                                  ; 发送停止位
T1:
SDZ DELAY                              ; 在发送每一位之间延时
JMP T1
MOV A,BAUDRATE                         ;
MOV DELAY,A                            ;
T2:

```

```

SDZ DELAY ;
JMP T2 ;
RET ;
TRANSMIT ENDP ;
RECEIVE PROC ;接收子程序
SZ RX ;检查起始位"0"
JMP RECEIVE ;若无起始位，则跳到receive
MOV A,9 ;接收到起始位则对接收数据进行初始化
MOV COUNT,A ;设置接收位数
MOV A,BAUDRATE+1 ;设置波特率延时
MOV DELAY,A ;
RXDELAY: ;
SDZ DELAY ;波特率延时
JMP RXDELAY ;
MOV A,BAUDRATE+1 ;重新载入波特率常数
MOV DELAY,A ;
SDZ COUNT ;检查数据接收是否完毕
JMP RXBIT ;如果接收未完毕则跳到RXBIT接收下一位数据
MOV A,RCREG ;如果接收完毕，则将准备接收数据
RET ;
RXBIT: ;
SET C ;先设置接收位为"1"
SNZ RX ;检查所接收位是否为"1"
CLR C ;若接收标志为"0"，则清除接收位 "0"
RRC RCREG ;带借位循环右移rcreg
JMP RXDELAY ;
RECEIVE ENDP ;
END
;-----
; End of DFSCRIPT.ASM
;-----

```

校对日期：2003-11-28

在 HT48r30A-1 上在两台仿真器之间运行通过

注意：为了保证接收正确，仿真器要共地。