嵌入式作业2

|  |
| --- |
| ;文件ASM\_Interrupt.s  ;(1)设置中断向量表  Mode\_USR EQU 0x50 ;IRQ中断开放，FIQ中断关闭  Mode\_FIQ EQU 0xD1 ;关闭IRQ、FIQ中断  Mode\_IRQ EQU 0xD2 ;关闭IRQ、FIQ中断  Mode\_SVC EQU 0xD3 ;关闭IRQ、FIQ中断  GET 2440Reg\_addr.inc  AREA MyCode**,** CODE**,READONLY**  IMPORT Uart\_Init  IMPORT INT\_UART0  ENTRY ;设置中断向量表  B ResetHandler ;Reset中断服务程序  B . ;handlerUndef  B . ;SWI interrupt handler  B . ;handlerPAbort  B . ;handlerDAbort  B . ;handlerReserved  B HandlerIRQ ;HandlerIRQ  B . ;HandlerFIQ  ;(2)复位的中断处理程序  ResetHandler  **BL** Clock\_Init ;初始化看门狗、时钟  **BL** MemSetup ;初始化SDRAM  LDR **SP,** **=**SvcStackSpace ;设置管理模式堆栈  MSR CPSR\_c**,** #Mode\_IRQ  LDR **SP,** **=**IrqStackSpace ;设置IRQ中断模式堆栈  MSR CPSR\_c**,** #Mode\_FIQ  LDR **SP,** **=**FiqStackSpace ;设置FIQ中断模式堆栈  MSR CPSR\_c**,** #Mode\_USR ;进入用户模式  LDR **SP,** **=**UsrStackSpace ;设置用户与系统模式堆栈  **BL** Init\_DATA ;初始化可读写数据  ;UART初始化  LDR R0**,** **=**pclk  LDR R1**,** **=**baud  **BL** Uart\_Init  ;把中断服务函数入口地址填入中断散转表  LDR R0**,** **=**pINT\_UART0  LDR R1**,** **=**INT\_UART0  **STR** R1**,** **[**R0**]**  MAIN\_LOOP  **NOP**  B MAIN\_LOOP ;死循环，被IRQ中断;初始化可读写数据区  Clock\_Init  GET Clock\_Init.s ;初始化看门狗、时钟  MemSetup  GET MemSetup.s ;初始化SDRAM  Init\_DATA  GET Init\_DATA.s ;初始化可读写数据区  ;(3)IRQ中断处理程序  HandlerIRQ  **SUB** LR**,**LR**,** #4 ;计算返回地址  STMFD **SP**!**,** {LR} ;保存断点到IRQ模式的堆栈  LDR LR**,** **=**Int\_Return ;修改LR，执行完INT\_UART0的处理程序后返回到Int\_Return处  LDR R0**,** **=**INTSUBMSK ;屏蔽发送和接收中断  LDR R1**,** **[**R0**]**  ORR R1**,** R1**,** #0x3  **STR** R1**,** **[**R0**]**  LDR R0**,** **=**INTOFFSET ;取得中断源的编号  LDR R1**,** **[**R0**]**  LDR R2**,** **=**Int\_EntryTable ;中断散转表起始地址  LDR PC**,** **[**R2**,**R1**,LSL**#2**]**  ;查中断散转表，取得INT\_UART0的处理程序入口地址  ;相当于无参的子程序调用，但没保存断点  Int\_Return ;执行完INT\_UART0的处理程序后须返回此处  LDR R0**,** **=**INTSUBMSK ;打开发送和接收中断  LDR R1**,** **[**R0**]**  MVN R2**,** #0x3  **and** R1**,** R1**,** R2  **STR** R1**,** **[**R0**]**  LDMFD **SP**!**,** {PC}**^** ;IRQ中断服务程序返回, ^表示将SPSR的值复制到CPSR  AREA MyRWData**,** DATA**,** READWRITE ;设置RW Base=0x33ffe700  Int\_EntryTable  GET Int\_EntryTable.s ;中断散转表，共32个入口  AREA MyZIData**,** DATA**,** READWRITE**,** NOINIT**,** ALIGN**=**8  ;下推式堆栈，根据对齐方式，段起始地址为0x33ffe800，各栈区空间均为1k  SPACE 0x100 **\*** 4 ;管理模式堆栈空间  SvcStackSpace SPACE 0x100 **\*** 4 ;中断模式堆栈空间  IrqStackSpace SPACE 0x100 **\*** 4 ;快速中断模式堆栈空间  FiqStackSpace SPACE 0x100 **\*** 4  UsrStackSpace  END |

|  |
| --- |
| ;文件Int\_EntryTable.s，定义中断散转表  ;共32路，对应仲裁器32路输入  pEINT0 DCD 0 ;存放EINT0-3中断服务程序入口地址  pEINT1 DCD 0  pEINT2 DCD 0  pEINT3 DCD 0  pEINT4\_7 DCD 0 ;存放EINT4\_7中断服务程序入口地址  pEINT8\_23 DCD 0 ;存放EINT8\_23中断服务程序入口地址——EINT8\_23  pINT\_CAM DCD 0 ;存放26个内部中断服务程序入口地址  pnBATT\_FLT DCD 0  pINT\_TICK DCD 0  pINT\_WDT\_AC97 DCD 0  ……  pINT\_UART0 DCD 0  pINT\_SPI1 DCD 0  pINT\_RTC DCD 0  pINT\_ADC DCD 0  END |

|  |
| --- |
| #include "2440Reg\_addr.h"  char**[**100**]** string**;**  void Uart\_Init**(**int pclk**,**int baud**)**  **{**  int i**;**  rGPHCON **|=** 0xa0**;** //GPH2,GPH3 as TXD0,RXD0  rGPHUP **=** 0x0**;** //GPH2,GPH3内部上拉  **if(**pclk **==** 0**)**  pclk **=** PCLK**;**  rUFCON0 **=** 0x0**;** //禁止3个通道的FIFO控制寄存器  rUFCON1 **=** 0x0**;**  rUFCON2 **=** 0x0**;**  rUMCON0 **=** 0x0**;** //初始化3个通道的MODEM控制寄存器，禁止AFC  rUMCON1 **=** 0x0**;**  rUMCON2 **=** 0x0**;**  //Line control register 0: Normal, No parity, 1 stop, 8 bits.  rULCON0 **=** 0x3**;**  // Control register 0: 发送和接收都采用中断方式, Loopback  rUCON0 **=** 0x325**;** //rUCN0[9:8]=11 rUCN0[5]=1 rUCN0[3:2]=01 rUCN0[1:0]=01  //Baud rate divisior register 0  rUBRDIV0**=(** **(**int**)(**pclk**/**16**/**baud**+**0.5**)** **-**1 **);**    rSRCPND **|=** 0x1**<<**28**;** //清除串口中断挂起SRCPND[28]      rSUBSRCPND **|=** 0x3**;** //清除收发中断SUBSRCPND[1:0]  rINTPND **|=** 0x1**<<**28**;** //清除串口中断请求  //打开总中断屏蔽和收发中断屏蔽  rINTMSK **&=** **~(**0x1**<<**28**);**  rINTSUBMSK **&=** **~(**0x3**);**  **}**  void INT\_UART0**()**  **{**  int sub\_int **=** rSUBSRCPND**;**  **switch** **(**sub\_int**)**  **{**  **case** 1**:** //INT\_RXD0  Uart\_GetString**(&**string**);**  **break;**  **case** 2**:** //INT\_TXD0  Uart\_SendString**(&**string**);**  **break;**  **default:**  **break;**  **}**  rSRCPND **|=** 0x1**<<**28**;** //清除串口中断挂起SRCPND[28]      rSUBSRCPND **|=** 0x3**;** //清除收发中断SUBSRCPND[1:0]  rINTPND **|=** 0x1**<<**28**;** //清除串口中断请求  **}**  //发送一字节数据  #define WrUTXH0(ch) ( \* (volatile unsigned char \* )0x50000020)=(unsigned char)(ch)  void Uart\_SendByte**(**unsigned char data**)**  **{**  WrUTXH0**(**data**);**  **}**  //发送字符串  void Uart\_SendString**(**unsigned char **\***pt**)**  **{**  **while(\***pt**)**  Uart\_SendByte**(\***pt**++);**  **}**  //接收一字节数据  unsigned char Uart\_Getch**(**void**)**  **{**  **return** RdURXH0**;**  **}**  //接收字符串  void Uart\_GetString**(**unsigned char **\***string**)**  **{**  unsigned char c**;**  **while((**c **=** Uart\_Getch**())** **!=** '\r'**)**  **\***string**++** **=** c**;**  **}** |