;File: putchar.a51

;Author: Eckart Hartmann Date:15/10/2003

;Development progress: Urt834.df

;

;\_putchar=========Output a character to UART.

;C Function prototype: char \_putchar(char cTx);

;Description of Function: Sends a character via UART.

;User interface: UART must be initialised with \_UrtCfg() before calling

; \_putchar. Put character to send in parameter1.

; Call \_putchar.

; \_putchar waits if UART busy then adds the required parity

; and sends the new character via the UART.

; Robustness: CD bits bits must be correct when communicating - variable

; cUrtVar (bits 4 to 6) contains the required CD value.

; Can be controlled by XON/XOFF protocol.

;Side effects: Overwrites a, cy, p, r1 RI and TI.

; Uses two stack levels.

;

NAME PUTCHAR

$NOMOD51

$IC(..kei841.inc) ; Parameter passing registers for Keil .

$IC(..kei841.dat) ; SFR definition for Keil .

;

extrn DATA (cUrtVar)

public \_putchar

?PR?\_putchar?PUTCHAR SEGMENT CODE

RSEG ?PR?\_putchar?PUTCHAR

USING 0

\_putchar:

mov a,cP1l ;if(ch=='\n')

xrl a,#0AH

jnz pncr ; {

mov ip1h,#0dh ; r6 = '\r';

lcall xoap ; xoap(); }

pncr: mov a,cP1l ;r6 = ch;

mov ip1h,a

lcall xoap ;xoap();

ret

;

xoap: jnb RI,putcTi ; if(RI) {

mov a,SBUF ; if(SBUF==XOFF)

cjne a,#013h,putcTi ; {

putcX: ; do {

clr RI ; RI = 0;

putcRi: jnb RI,putcRi ; while(!RI);

; }

mov a,SBUF ; while(SBUF!=XON)

cjne a,#011h,putcX

clr RI ; RI = 0;

; }}

putcTi: jnb TI,putcTi ; while(!TI);

clr TI ; TI = 0;

setb TB8 ; // As default.

mov a,iP1h ; c = parity(iP1h);

mov c,p

mov r1,#cUrtVar ;if(noparity);

mov a,@r1

jnb ACC.3,putcR ;else

anl a,#0ch ; {

mov TB8,c ; TB8 = c;

cjne a,#8,putcR ; if(parity=='o')

cpl TB8 ; TB8 = !TB8; }

putcR: mov SBUF,iP1h ;SBUF = iP1h;

ret

;

;Function End==========================================================Function End

end